МИР
цифровой обработки

РАФАЭЛ С. ГОНСАЛЕС,
РИЧАРД Е. ВУДС

Цифровая обработка изображений

Третье издание,
исправленное и дополненное

Перевод с английского
Л.И. Рубанова, П.А. Чочиа

Научный редактор перевода
П.А. Чочиа

ТЕХНОСФЕРА
Москва
2012

Издано при финансовой поддержке Федерального агентства по печати и массовым коммуникациям в рамках Федеральной целевой программы «Культура России (2012-2018 годы)»

УДК 004.932
ББК 32.81
Г65
Г65 Гонсалес Р., Вудс Р.
Цифровая обработка изображений
Издание 3-е, исправленное и дополненное

Москва: Техносфера, 2012. – 1104 с. , ISBN 978-5-94836-331-8

Настоящее издание является результатом значительной переработки книги «Цифровая обработка изображений» (Гонсалес и Уинтц, 1977 г. и 1978 г.; Гонсалес и Вудс, 1992 г. и 2002 г.). Одна из важнейших причин популярности книги, которая уже более 30 лет является мировым лидером в своей области – высокая степень внимания авторов к изменению образовательных потребностей читателя. Нынешнее издание базируется на самом обширном из когда-либо проводившихся исследований читательского мнения.

Как и прежде, основные цели книги — служить введением в основные понятия и методы цифровой обработки изображений, а также создать основу для последующего изучения и проведения самостоятельных исследований в этой области. Все разделы сопровождаются большим количеством примеров и иллюстраций.

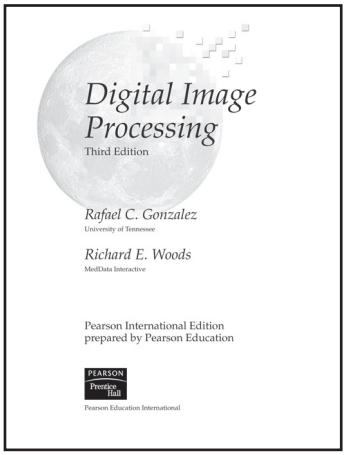
Книга рассчитана на научных работников, профессиональных программистов, специалистов по компьютерному дизайну, студентов и преподавателей. Книга постоянно занимает первое место в рейтинге продаж Amazon.com и широко используется разработчиками и дизайнерами.

УДК 004.932
ББК 32.81

Authorized translation from the English language edition, entitled DIGITAL IMAGE PROCESSING: International Version 3rd Edition; ISBN 0132345633, by GONZALEZ, RAFAEL C; WOODS, RICHARD E., published by Pearson Education, Inc, publishing as Prentice Hall. Copyright ©2008 by Pearson Education, Inc.
All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc. RUSSIAN language edition published by TECHNOSPHERA publishers.
Copyright ©2012

Авторизованный перевод издания на английском языке, под названием DIGITAL IMAGE PROCESSING (ЦИФРОВАЯ ОБРАБОТКА ИЗОБРАЖЕНИЙ): Международная версия, 3-е Издание; ISBN 0132345633, GONZALEZ, RAFAEL C; WOODS, RICHARD E., изданного компанией Pearson Education, Inc, опубликованного Prentice Hall, Copyright ©2008, Pearson Education, Inc. Все права защищены. Никакая часть этой книги не может быть воспроизведена или передана в любой форме или любыми средствами, электронными или механическими, включая фотокопирование, запись или извлечение любой информации из поисковой системы без получения разрешения от компании Pearson Education, Inc.

© 2012, ЗАО «РИЦ «Техносфера», перевод на русский язык, оригинал-макет, оформление



ISBN 978-5-94836-331-8
ISBN 978-0-13-234563-7 (англ.)

Саманте и Дженис,
Дэвиду и Джонатану

Содержание

Предисловие к английскому изданию 11
Благодарности 15
Сайт книги в сети Интернет 16
Об авторах 17
Рафаэл С. Гонсалес 17
Ричард Е. Вудс 18
Предисловие научного редактора перевода 19

Глава 1. Введение 22
1.1. Что такое цифровая обработка изображений? 22
1.2. Истоки цифровой обработки изображений 24
1.3. Примеры областей применения цифровой обработки изображений 29
1.3.1. Формирование изображений с помощью гамма-лучей 30
1.3.2. Рентгеновские изображения 32
1.3.3. Изображения в ультрафиолетовом диапазоне 34
1.3.4. Изображения в видимом и инфракрасном диапазонах 36
1.3.5. Изображения в микроволновом диапазоне 43
1.3.6. Изображения в диапазоне радиоволн 44
1.3.7. Примеры, иллюстрирующие другие способы формирования изображений 46
1.4. Основные стадии цифровой обработки изображений 51
1.5. Компоненты системы обработки изображений 54
Заключение 58
Ссылки и литература для дальнейшего изучения 58
Литература, добавленная при переводе 61

Глава 2. Основы цифрового представления изображений 65
Введение 65
2.1. Элементы зрительного восприятия 65
2.1.1. Строение человеческого глаза 66
2.1.2. Формирование изображения в глазу 69
2.1.3. Яркостная адаптация и контрастная чувствительность 70
2.2. Свет и электромагнитный спектр 74
2.3. Считывание и регистрация изображения 78
2.3.1. Регистрация изображения с помощью одиночного сенсора 79
2.3.2. Регистрация изображения с помощью линейки сенсоров 80
2.3.3. Регистрация изображения с помощью матрицы сенсоров 82
2.3.4. Простая модель формирования изображения 83
2.4. Дискретизация и квантование изображения 85
2.4.1. Основные понятия, используемые при дискретизации и квантовании 85
2.4.2. Представление цифрового изображения 87
2.4.3. Пространственное и яркостное разрешения 92
2.4.4. Интерполяция цифрового изображения 99

2.5. Некоторые фундаментальные отношения между пикселями 102
2.5.1. Соседи отдельного элемента 102
2.5.2. Смежность, связность, области и границы 102
2.5.3. Меры расстояния 105
2.6. Введение в математический аппарат, применяемый в цифровой обработке изображений 107
2.6.1. Поэлементные и матричные операции 107
2.6.2. Линейные и нелинейные преобразования 108
2.6.3. Арифметические операции 109
2.6.4. Теоретико-множественные и логические операции 115
2.6.5. Пространственные операции 121
2.6.6. Векторные и матричные операции 129
2.6.7. Преобразования изображений 130
2.6.8. Вероятностные методы 133
Заключение 135
Ссылки и литература для дальнейшего изучения 136
Задачи 138

Глава 3. Яркостные преобразования и пространственная фильтрация 143
Введение 143
3.1. Предпосылки 143
3.1.1. Основы яркостных преобразований и пространственной фильтрации 143
3.1.2. О примерах, приводимых в данной главе 146
3.2. Некоторые основные градационные преобразования 147
3.2.1. Преобразование изображения в негатив 147
3.2.2. Логарифмическое преобразование 148
3.2.3. Степенные преобразования (гамма-коррекция) 150
3.2.4. Кусочно-линейные функции преобразований 155
3.3. Видоизменение гистограммы 160
3.3.1. Эквализация гистограммы 162
3.3.2. Приведение гистограммы (задание гистограммы) 170
3.3.3. Локальная гистограммная обработка 179
3.3.4. Использование гистограммных статистик для улучшения изображения 181
3.4. Основы пространственной фильтрации 186
3.4.1. Механизмы пространственной фильтрации 186
3.4.2. Пространственная корреляция и свертка 188
3.4.3. Векторное представление линейной фильтрации 193
3.4.4. Формирование масок пространственных фильтров 193
3.5. Сглаживающие пространственные фильтры 194
3.5.1. Линейные сглаживающие фильтры 195
3.5.2. Фильтры, основанные на порядковых статистиках (нелинейные фильтры) 199
3.6. Пространственные фильтры повышения резкости 200
3.6.1. Основы 201
3.6.2. Повышение резкости изображений с использованием вторых производных: лапласиан 203
3.6.3. Нерезкое маскирование и фильтрация с подъемом высоких частот 207
3.6.4. Использование производных первого порядка для (нелинейного) повышения резкости изображений: градиент 209
3.7. Комбинирование методов пространственного улучшения 212

3.8. Применение нечетких методов для яркостных преобразований и пространственной фильтрации 217
3.8.1. Введение 217
3.8.2. Начала теории нечетких множеств 218
3.8.3. Использование нечетких множеств 222
3.8.4. Использование нечетких множеств для яркостных преобразований 232
3.8.5. Использование нечетких множеств для пространственной фильтрации 235
Заключение 238
Ссылки и литература для дальнейшего изучения 238
Задачи 239

Глава 4. Фильтрация в частотной области 246
Введение 246
4.1. Основы 247
4.1.1. Краткая история ряда и преобразования Фурье 247
4.1.2. О примерах, приводимых в данной главе 248
4.2. Предварительные понятия 249
4.2.1. Комплексные числа 249
4.2.2. Ряды Фурье 250
4.2.3. Импульсы и их свойство отсеивания 250
4.2.4. Преобразование Фурье функции одной непрерывной переменной 252
4.2.5. Свертка 256
4.3. Дискретизация и преобразование Фурье дискретных функций 257
4.3.1. Дискретизация 258
4.3.2. Преобразование Фурье дискретизованных функций 258
4.3.3. Теорема отсчетов 261
4.3.4. Наложение спектров 264
4.3.5. Реконструкция (восстановление) функции из отсчетов 267
4.4. Дискретное преобразование Фурье (ДПФ) одной переменной 268
4.4.1. Получение ДПФ из непрерывного преобразования дискретизованных функций 268
4.4.2. Взаимосвязь между шагом дискретизации и частотными интервалами 271
4.5. Расширение на функции двух переменных 272
4.5.1. Двумерный импульс и его свойство отсеивания 272
4.5.2. Пара двумерных непрерывных преобразований Фурье 273
4.5.3. Двумерная дискретизация и двумерная теорема отсчетов 274
4.5.4. Наложение спектров при преобразовании изображений 276
4.5.5. Двумерное дискретное преобразование Фурье и его обращение 284
4.6. Некоторые свойства двумерного дискретного преобразования Фурье 284
4.6.1. Взаимосвязи пространственных и частотных интервалов 284
4.6.2. Сдвиг и поворот 285
4.6.3. Периодичность 285
4.6.4. Свойства симметрии 287
4.6.5. Фурье-спектр и фаза 293
4.6.6. Двумерная теорема о свертке 298
4.6.7. Краткое изложение свойств двумерного дискретного преобразования Фурье 304
4.7. Основы фильтрации в частотной области 304
4.7.1. Дополнительные характеристики частотной области 304

4.7.2. Основы частотной фильтрации 306
4.7.3. Последовательность шагов частотной фильтрации 313
4.7.4. Соответствие между пространственными и частотными фильтрами 314
4.8. Частотные фильтры сглаживания изображения 320
4.8.1. Идеальные фильтры низких частот 320
4.8.2. Фильтры низких частот Баттерворта 324
4.8.3. Гауссовы фильтры низких частот 327
4.8.4. Дополнительные примеры низкочастотной фильтрации 330
4.9. Повышения резкости изображений частотными фильтрами 332
4.9.1. Идеальные фильтры высоких частот 334
4.9.2. Фильтры высоких частот Баттерворта 336
4.9.3. Гауссовы фильтры высоких частот 336
4.9.4. Лапласиан в частотной области 338
4.9.5. Нерезкое маскирование, высокочастотная фильтрация с подъемом частотной характеристики, фильтрация с усилением высоких частот 339
4.9.6. Гомоморфная фильтрация 342
4.10. Избирательная фильтрация 346
4.10.1. Режекторные и полосовые пропускающие фильтры 346
4.10.2. Узкополосные фильтры 347
4.11. Вопросы реализации 351
4.11.1. Разделимость двумерного ДПФ 351
4.11.2. Вычисление обратного ДПФ при помощи алгоритма прямого ДПФ 352
4.11.3. Быстрое преобразование Фурье 353
4.11.4. Некоторые замечания по поводу построения фильтров 356
Заключение 357
Ссылки и литература для дальнейшего изучения 357
Задачи 359

Глава 5. Восстановление и реконструкция изображений 366
Введение 366
5.1. Модель процесса искажения/восстановления изображения 367
5.2. Модели шума 368
5.2.1. Пространственные и частотные свойства шума 369
5.2.2. Функции плотности распределения вероятностей для некоторых важных типов шума 369
5.2.3. Периодический шум 374
5.2.4. Построение оценок для параметров шума 375
5.3. Подавление шумов — пространственная фильтрация 378
5.3.1. Усредняющие фильтры 378
5.3.2. Фильтры, основанные на порядковых статистиках 382
5.3.3. Адаптивные фильтры 386
5.4. Подавление периодического шума — частотная фильтрация 393
5.4.1. Режекторные фильтры 393
5.4.2. Полосовые фильтры 395
5.4.3. Узкополосные фильтры 396
5.4.4. Оптимальная узкополосная фильтрация 398
5.5. Линейные трансляционно-инвариантные искажения 403
5.6. Оценка искажающей функции 406
5.6.1. Оценка на основе визуального анализа изображения 407
5.6.2. Оценка на основе эксперимента 407
5.6.3. Оценка на основе моделирования 408
5.7. Инверсная фильтрация 411

5.8. Фильтрация методом минимизации среднего квадрата отклонения (винеровская фильтрация) 414
5.9. Фильтрация методом минимизации сглаживающего функционала со связью 418
5.10. Среднегеометрический фильтр 424
5.11. Реконструкция изображения по проекциям 424
5.11.1. Введение 425
5.11.2. Принципы компьютерной томографии (КТ) 428
5.11.3. Проекции и преобразование Радона 432
5.11.4. Теорема о центральном сечении 438
5.11.5. Реконструкция по проекциям в параллельных пучках методом фильтрации и обратного проецирования 439
5.11.6. Реконструкция на основе фильтрованных обратных проекций с веерным пучком 445
Заключение 452
Ссылки и литература для дальнейшего изучения 453
Задачи 454

Глава 6. Обработка цветных изображений 460
Введение 460
6.1. Основы теории цвета 461
6.2. Цветовые модели 468
6.2.1. Цветовая модель RGB 469
6.2.2. Цветовые модели CMY и CMYK 474
6.2.3. Цветовая модель HSI 475
6.3. Обработка изображений в псевдоцветах 483
6.3.1. Квантование по яркости 484
6.3.2. Преобразование яркости в цвет 489
6.4. Основы обработки цветных изображений 495
6.5. Цветовые преобразования 496
6.5.1. Постановка задачи 496
6.5.2. Цветовое дополнение 501
6.5.3. Вырезание цветового диапазона 502
6.5.4. Яркостная и цветовая коррекция 504
6.5.5. Обработка гистограмм 509
6.6. Сглаживание и повышение резкости 511
6.6.1. Сглаживание цветных изображений 511
6.6.2. Повышение резкости цветных изображений 514
6.7. Сегментация изображения, основанная на цвете 515
6.7.1. Сегментация в цветовом пространстве HSI 515
6.7.2. Сегментация в цветовом пространстве RGB 517
6.7.3. Обнаружение контуров на цветных изображениях 520
6.8. Шум на цветных изображениях 524
6.9. Сжатие цветных изображений 527
Заключение 529
Ссылки и литература для дальнейшего изучения 529
Задачи 530

Глава 7. Вейвлеты и кратномасштабная обработка 536
Введение 536
7.1. Предпосылки 537

7.1.1. Пирамиды изображений 538
7.1.2. Субполосное кодирование 543
7.1.3. Преобразование Хаара 550
7.2. Кратномасштабное разложение 554
7.2.1. Разложения в ряды 554
7.2.2. Масштабирующие функции 557
7.2.3. Вейвлет-функции 562
7.3. Одномерные вейвлет-преобразования 566
7.3.1. Разложение в вейвлет-ряды 566
7.3.2. Дискретное вейвлет-преобразование 569
7.3.3. Интегральное вейвлет-преобразование 571
7.4. Быстрое вейвлет-преобразование 574
7.5. Двумерные вейвлет-преобразования 583
7.6. Вейвлет-пакеты 593
Заключение 604
Ссылки и литература для дальнейшего изучения 604
Задачи 606

Глава 8. Сжатие изображений 611
Введение 611
8.1. Основы 612
8.1.1. Кодовая избыточность 614
8.1.2. Пространственная и временна́я избыточность 616
8.1.3. Лишняя информация 617
8.1.4. Измерение содержащейся в изображении информации 618
8.1.5. Критерии верности воспроизведения 621
8.1.6. Модели сжатия изображений 624
8.1.7. Форматы изображений, контейнеры и стандарты сжатия 626
8.2. Некоторые основные методы сжатия 630
8.2.1. Кодирование Хаффмана 630
8.2.2. Кодирование Голомба 633
8.2.3. Арифметическое кодирование 637
8.2.4. LZW-кодирование 640
8.2.5. Кодирование длин серий 644
8.2.6. Кодирование на базе шаблонов 651
8.2.7. Кодирование битовых плоскостей 655
8.2.8. Блочное трансформационное кодирование 659
8.2.9. Кодирование с предсказанием 679
8.2.10. Вейвлет-кодирование 700
8.3. Нанесение цифровых водяных знаков на изображение 713
Заключение 722
Ссылки и литература для дальнейшего изучения 722
Задачи 724

Глава 9. Морфологическая обработка изображений 728
Введение 728
9.1. Начальные сведения 728
9.2. Эрозия и дилатация 732
9.2.1. Эрозия 732
9.2.2. Дилатация 734
9.2.3. Двойственность 737

9.3. Размыкание и замыкание 737
9.4. Преобразование «попадание/пропуск» 742
9.5. Некоторые основные морфологические алгоритмы 745
9.5.1. Выделение границ 745
9.5.2. Заполнение дырок 746
9.5.3. Выделение связных компонент 748
9.5.4. Выпуклая оболочка 751
9.5.5. Утончение 753
9.5.6. Утолщение 754
9.5.7. Построение остова 755
9.5.8. Усечение 758
9.5.9. Морфологическая реконструкция 761
9.5.10. Сводная таблица морфологических операций 768
9.6. Морфология полутоновых изображений 771
9.6.1. Эрозия и дилатация 772
9.6.2. Размыкание и замыкание 774
9.6.3. Некоторые основные алгоритмы полутоновой морфологии 777
9.6.4. Полутоновая морфологическая реконструкция 784
Заключение 787
Ссылки и литература для дальнейшего изучения 788
Задачи 788

Глава 10. Сегментация изображений 797
Введение 797
10.1. Основы 798
10.2. Обнаружение точек, линий и перепадов 800
10.2.1. Основы 801
10.2.2. Обнаружение изолированных точек 805
10.2.3. Обнаружение линий 806
10.2.4. Модели перепадов 811
10.2.5. Простые методы обнаружения контурных перепадов 817
10.2.6. Более совершенные методы обнаружения контуров 826
10.2.7. Связывание контуров и нахождение границ 840
10.3. Пороговая обработка 854
10.3.1. Обоснование 854
10.3.2. Обработка с глобальным порогом 858
10.3.3. Метод Оцу оптимального глобального порогового преобразования 860
10.3.4. Применение сглаживания изображения для улучшения обработки с глобальным порогом 865
10.3.5. Использование контуров для улучшения обработки с глобальным порогом 867
10.3.6. Обработка с несколькими порогами 870
10.3.7. Обработка с переменным порогом 874
10.3.8. Пороги, основанные на нескольких переменных 880
10.4. Сегментация на отдельные области 881
10.4.1. Выращивание областей 882
10.4.2. Разделение и слияние областей 886
10.5. Сегментация по морфологическим водоразделам 889
10.5.1. Исходные предпосылки 889
10.5.2. Построение перегородок 892
10.5.3. Алгоритм сегментации по водоразделам 894

10.5.4. Использование маркеров 896
10.6. Использование движения при сегментации 899
10.6.1. Пространственные методы 899
10.6.2. Частотные методы 903
Заключение 907
Ссылки и литература для дальнейшего изучения 907
Задачи 909

Глава 11.Представление и описание 919
Введение 919
11.1. Представление 919
11.1.1. Прослеживание границы 920
11.1.2. Цепные коды 922
11.1.3. Аппроксимация ломаной линией минимальной длины 925
11.1.4. Другие методы аппроксимации ломаной линией 932
11.1.5. Сигнатуры 934
11.1.6. Сегменты границы 936
11.1.7. Остовы областей 938
11.2. Дескрипторы границ 941
11.2.1. Некоторые простые дескрипторы 941
11.2.2. Нумерация фигур 942
11.2.3. Фурье-дескрипторы 944
11.2.4. Статистические характеристики 947
11.3. Дескрипторы областей 948
11.3.1. Некоторые простые дескрипторы 949
11.3.2. Топологические дескрипторы 950
11.3.3. Текстурные дескрипторы 954
11.3.4. Инварианты моментов двумерных функций 966
11.4. Использование главных компонент для описания 969
11.5. Реляционные дескрипторы 980
Заключение 983
Ссылки и литература для дальнейшего изучения 984
Задачи 985

Глава 12.Распознавание объектов 990
Введение 990
12.1. Образы и классы образов 990
12.2. Распознавание на основе методов теории принятия решений 995
12.2.1. Сопоставление 996
12.2.2. Статистически оптимальные классификаторы 1003
12.2.3. Нейронные сети 1013
12.3. Структурные методы распознавания 1036
12.3.1. Сопоставление номеров фигур 1036
12.3.2. Сопоставление строк символов 1037
Заключение 1039
Ссылки и литература для дальнейшего изучения 1040
Задачи 1041
Приложения 1045
Кодовые таблицы для сжатия изображений 1045
Литература 1050
Предметный указатель 1081

Предисловие к английскому изданию

Если что-то удается читать без усилий,
значит больших усилий стоило это написать.
Энрике Хардиел Понсела

Настоящее издание является результатом значительной переработки книги «Цифровая обработка изображений». При подготовке этого издания, как и при подготовке предыдущих четырех редакций (Гонсалес и Уинтц, 1977 г. и 1978 г.; Гонсалес и Вудс, 1992 г. и 2002 г.), мы ориентировались прежде всего на студентов и преподавателей. Как и прежде, основные цели этой книги — служить введением в основные понятия и методы цифровой обработки изображений, а также создать основу для последующего изучения и проведения самостоятельных исследований в этой области. С учетом этих целей мы вновь сконцентрировались на материале, который считаем фундаментальным и применимым не только для решения узкоспециальных задач. Уровень математической сложности книги остался в рамках программы колледжа высшей ступени или первого курса университета и предполагает начальную подготовку в области математического анализа, линейной алгебры и линейных систем, теории вероятности и математической статистики, а также компьютерного программирования. Дополняющий книгу Интернет-сайт содержит учебные материалы, помогающие читателям при необходимости освежить в памяти эти темы.

Одна из важнейших причин, почему настоящая книга уже более 30 лет является мировым лидером в своей области, — это высокая степень внимания к изменению образовательных потребностей наших читателей. Нынешнее издание базируется на самом обширном из когда-либо нами проводившихся исследований читательского мнения. К исследованию были привлечены профессорско-преподавательский состав, студенты и независимые читатели из 134 организаций в 32 странах. Исследование показало, что необходимо:

- расширить введение в начале книги касательно математических инструментов, используемых в обработке изображений;
- развернуто объяснить технику обработки гистограмм;
- излагать сложные алгоритмы в форме пошагового краткого описания;
- подробнее объяснить пространственную корреляцию и свертку;
- добавить введение в теорию нечетких множеств и применение ее в цифровой обработке изображений;
- пересмотреть материал, относящийся к частотной области, начиная с базовых принципов и демонстрации того, как дискретное преобразование Фурье следует из дискретизации данных;
- охватить тематику, связанную с компьютерной томографией;
- разъяснить основные понятия в главе о вейвлет-анализе ;
- переработать главу о сжатии данных, чтобы шире охватить методы сжатия видеоинформации, обновленные стандарты и технологию цифровых водяных знаков;
- расширить главу о морфологии с целью охвата морфологического восстановления изображений и переработать раздел о полутоновой морфологии;

- расширить тематику сегментации изображений за счет более совершенных методов обнаружения контуров, таких как алгоритм Кэнни, и более широкой трактовки пороговой обработки изображений;
- обновить содержание главы о представлении и описании изображений;
- упростить материал, относящийся к структурным методам распознавания образов.

Переработка и дополнение книги, предпринятые в этом издании, отражают наше стремление соблюсти разумный баланс между строгостью и ясностью изложения и конъюнктурными соображениями, оставаясь при этом в рамках приемлемого объема книги. Наиболее важные изменения, сделанные в этом издании книги, перечислены ниже.

Глава 1: Внесены изменения в несколько иллюстраций, а текст частично переработан в соответствии с изменениями в последующих главах.

Глава 2: Приблизительно половина главы переработана, чтобы включить новые изображения и более понятные объяснения. В число крупных изменений входят новый раздел, посвященный интерполяции изображений, и обширный новый раздел, обобщающий главные математические инструменты, которые используются в книге. Однако вместо сухого перечисления математических концепций мы использовали эту возможность, чтобы во второй главе ввести в оборот ряд прикладных задач обработки изображений, которые прежде были разбросаны по всей книге. Например, в эту главу мы перенесли усреднение и вычитание изображений в качестве иллюстрации арифметических операций. Здесь мы следуем тенденции, начатой во втором издании книги: как можно раньше вводить в обсуждение приложения, не только для иллюстрации, но и для мотивации студентов. Закончив изучение главы 2 в ее обновленном виде, читатель получит базовое понимание того, как осуществляется манипулирование цифровыми изображениями и их обработка. Это та твердая основа, на которой строится остальная часть книги.

Глава 3: Основная переработка этой главы связана с подробным обсуждением пространственной корреляции и свертки и их применением для фильтрации изображений с помощью пространственных масок. Исследование рынка также выявило потребность в численных примерах, иллюстрирующих технику эквализации и задания гистограммы, поэтому мы добавили несколько примеров для демонстрации действия этих инструментов обработки. При исследовании часто высказывалось пожелание рассмотреть нечеткие множества и их приложения к обработке изображений. Мы включили в эту главу новый раздел, посвященный основам теории нечетких множеств и ее применениям для яркостных преобразований и пространственной фильтрации — двум главным областям использования этой теории в обработке изображений.

Глава 4: За последние четыре года больше всего замечаний и предложений мы слышали в связи с изменениями, сделанными в главе 4 при переходе от первого ко второму изданию. Целью этих изменений было упростить изложение преобразования Фурье и частотной области. Очевидно, мы зашли слишком далеко, и многие пользователи книги жаловались на излишнюю поверхностность материала. В настоящем издании мы исправили этот дефект. Теперь изложение материала начинается с преобразования Фурье одной непрерывной переменной и далее переходит к выводу дискретного фурье-преобразования,

исходя из базовых понятий дискретизации и свертки. Побочным результатом такой формы подачи материала стал наглядный вывод теоремы отсчетов и ее следствий. После этого изложение для одномерной постановки распространяется на двумерный случай, где мы приводим ряд примеров, иллюстрирующих влияние дискретизации на цифровые изображения, в том числе эффекты, вызванные наложением спектров и муаром. Затем мы иллюстрируем двумерное дискретное преобразование Фурье, выводя и суммируя ряд его важных свойств. Эти понятия далее используются в качестве основы для фильтрации в частотной области. В заключение обсуждаются вопросы реализации, такие как разложение преобразований и вывод алгоритма быстрого преобразования Фурье. К концу этой главы читатель перейдет от дискретизации одномерных функций через понятный вывод основ дискретного фурье-преобразования к некоторым важнейшим его применениям для цифровой обработки изображений.

Глава 5: Основным изменением в этой главе стало добавление раздела, касающегося реконструкции изображения по проекциям, с упором на компьютерную томографию (КТ). Изложение КТ начинается с наглядного примера, демонстрирующего основополагающие принципы реконструкции изображения по проекциям, и описания различных применяемых на практике методов формирования изображения. Затем мы выводим преобразование Радона и теорему о центральном сечении, используя их как основу для формулировки концепции фильтрации и обратного проецирования. Обсуждается реконструкция изображений с помощью как параллельных, так и веерных пучков; обсуждение иллюстрируется примерами. Включение этого уже давно ожидавшегося материала представляет собой важное добавление к книге.

Глава 6: Изменения в этой главе ограничивались только уточнениями и незначительной коррекцией обозначений без добавления каких-либо новых понятий.

Глава 7: Многочисленные полученные замечания относились к тому, что переход от предшествующих глав к вейвлетам оказался трудным для начинающих. Чтобы сделать материал более понятным, мы заново переписали несколько подготовительных разделов.

Глава 8: Эта глава полностью переработана с учётом достигнутого современного уровня. Основные изменения касаются новых методов кодирования, более широкого обсуждения видеоданных, полностью переработанного раздела о стандартах сжатия и введения в технологию цифровых водяных знаков. Новая организация материала упрощает его освоение начинающими студентами.

Глава 9: Основные изменения в этой главе состоят в добавлении нового раздела о морфологической реконструкции и полной переработке раздела, посвященного полутоновой морфологии. Благодаря рассмотрению морфологической реконструкции как для двоичных, так и для полутоновых изображений, появилась возможность изложить более сложные и полезные морфологические алгоритмы, чем прежде.

Глава 10: Эта глава также подверглась значительной переработке. Сохранив прежнюю организацию, мы добавили новый материал, лучше подчеркивающий основные принципы, а также обсуждение более совершенных методов сегментации. Модели перепадов теперь рассматриваются и иллюстрируются более подробно, как и свойства градиента. Детекторы контуров Марра-Хилдрета

и Кэнни добавлены в качестве иллюстрации более совершенных методов обнаружения контуров. Также переработан раздел о пороговой обработке, в который добавлен метод Оцу нахождения оптимального порога, популярность которого значительно возросла за последние несколько лет. Мы включили этот подход вместо выбора оптимального порога на основе байесовского правила классификации не только потому, что его легче понять и реализовать, но и поскольку он значительно чаще используется на практике. Байесовский подход перенесен в главу 12, где решающее правило Байеса обсуждается более подробно. Мы также добавили обсуждение того, как использовать информацию о контурах для улучшения пороговой обработки, и несколько новых примеров обработки с адаптивным порогом. В разделы, посвященные морфологическим водоразделам и использованию движения для сегментации, внесены только незначительные уточнения.

Глава 11: Главные изменения в этой главе связаны с добавлением алгоритма прослеживания границы, более подробным изложением алгоритма аппроксимации дискретной границы с помощью ломаной минимальной длины и новым разделом об использовании матриц сочетаемости яркостей для описания текстуры. Новыми являются многие примеры в разделах 11.2 и 11.3, а также все примеры в разделе 11.4.

Глава 12: Изменения в этой главе включают новый раздел о корреляционном сопоставлении и новый пример, демонстрирующий использование байесовского классификатора для распознавания интересуюших областей в мультиспектральных изображениях. Раздел о структурной классификации теперь ограничен обсуждением только сопоставления строк символов.

Все вышеуказанные изменения выразились в появлении в книге более 400 новых изображений, свыше 200 новых рисунков и таблиц, и более 80 новых задач и упражнений для самостоятельной работы. Сложные процедуры обработки по мере необходимости сведены в сжатую форму пошаговых алгоритмов. Также обновились ссылки на литературу в конце каждой главы.

Интернет-сайт книги, основанный при выпуске второго издания, успешно действует, привлекая более 20 тысяч посетителей ежемесячно. Этот сайт был переделан и пополнен новой информацией в соответствии с выпуском настоящего издания книги. Более подробно возможности сайта и его наполнение представлены в разделе «Сайт книги в сети Интернет», сразу после раздела «Благодарности».

Это издание книги «Цифровая обработка изображений» отражает то, как изменились с 2002 г. образовательные потребности наших читателей. Как обычно для подобных проектов, прогресс продолжается и после окончания работы над рукописью. Одна из причин, по которым эта книга была так хорошо принята, начиная с ее появления в 1977 г., — это ее постоянный акцент на фундаментальных концепциях. Такой подход является одной из попыток указать устойчивые ориентиры в стремительно развивающейся совокупности знаний. Мы старались соблюсти тот же принцип при подготовке этого издания книги.

Р. С. Г.
Р. Е. В.

Благодарности

Мы в неоплатном долгу перед множеством лиц, как из академических кругов, так и из промышленности и правительства, которые внесли большой вклад в это издание книги. Их помощь была столь важной и разносторонней, что нам трудно выделить их как-то иначе, чем просто перечислить в алфавитном порядке. В частности, мы глубоко признательны за те многочисленные советы по улучшению формы изложения и содержания книги, которые предложили наши коллеги Монги А. Абиди, Стивен Л. Эддинс, Йонг Мин Ким, Брайан Морс, Эндрю Олдройд, Али М. Реза, Эдгардо Фелипе Риверон, Хосе Руиз Шульклопер и Кэмерон Х. Г. Райт.

Многие лица и организации оказали нам ценную помощь во время написания настоящего издания. Мы перечисляем их опять-таки по алфавиту. Мы особо признательны Кертни Эспозито и Наоми Фернандес из компании The MathWorks, предоставившим нам пакет программ MATLAB и оказавшим помощь в его использовании, что позволило построить и уточнить многие примеры и экспериментальные результаты, включенные в настоящее издание. Значительная доля использованных в нем новых изображений (а в некоторых случаях – их история и объяснение) поступили к нам благодаря усилиям лиц, чей вклад воистину неоценим, среди которых, в частности: Серж Бехер, Мелисса Д. Байнд, Джеймс Бланкеншип, Уве Боос, Эрнесто Брибиеска, Майкл Е. Кейси, Майкл У. Дэвидсон, Сьюзен Л. Форсбург, Томас Р. Гест, Лалит Гупта, Дэниел А. Хаммер, Жонг Хе, Роджер Хиди, Хуан А. Херрера, Джон М. Хьюдек, Майкл Гурвиц, Крис Й. Йохансен, Ронда Найтон, Эшли Мохамед, А. Моррис, Кертис С. Обер, Джозеф Е. Пассенте, Дэвид Р. Пикенс, Майкл Робинсон, Барретт А. Шефер, Майкл Шаффер, Пит Сайтс, Салли Стоу, Крейг Уотсон, Дэвид К. Уи и Роберт А. Вест. Мы также признательны другим лицам и организациям, упоминающимся в подписях ко многим фотографиям, за любезное разрешение использовать в книге предоставленные ими материалы.

Сотрудники издательства Прентис-Холл Винс О'Брайен, Роуз Кернан, Скотт Дисанно, Майкл Мак-Дональд, Джо Руддик, Хитер Скотт и Элис Дворкин заслуживают особой благодарности. Их творческие способности, помощь и терпение при выпуске книги были действительно неоценимыми.

Р. С. Г.
Р. Е. В.

Сайт книги в сети Интернет

www.prenhall.com/gonzalezwoods
или зеркальный сайт
www.imageprocessingplace.com

Хотя книга «Цифровая обработка изображений» содержит в себе весь необходимый материал, дополняющий ее сайт в Интернет предлагает помощь в ряде важных вопросов.

Для студентов и самостоятельных читателей на сайте имеются :

- краткие обзоры по таким областям, как теория вероятностей, математическая статистика, линейная алгебра;
- подробные решения к некоторым из предлагаемых в книге упражнений;
- учебные проекты для самостоятельной компьютерной реализации;
- раздел «Учебные пособия» с десятками обучающих материалов по большинству тем, рассматриваемых в книге;
- база данных, содержащая все изображения из книги.

Для преподавателей сайт предлагает:

- «Пособие для преподавателя» с полными решениями всех упражнений из книги, а также примерами учебных планов и методическими указаниями к проведению лабораторных работ. Это руководство бесплатно предоставляется преподавателям, которые принимают данную книгу в качестве основы для учебного курса;
- материалы для учебных презентаций в формате PowerPoint;
- изъятые из книги части прежних ее изданий в формате PDF, которые преподаватель может при желании получить по сети и использовать в учебном процессе;
- многочисленные ссылки на другие образовательные ресурсы.

Для практиков на сайте имеются дополнительные специализированные разделы, например:

- ссылки на коммерческие сайты Интернет по тематике;
- отдельные новые литературные ссылки;
- ссылки на коммерческие базы данных изображений.

Данный Интернет-сайт является идеальным инструментом для актуализации книги в промежутках между изданиями, позволяя добавлять новые темы, цифровые изображения и прочий материал, появившийся после выхода книги из печати. Хотя выпуск книги осуществлялся с большим вниманием и заботой, сайт также позволяет удобно размещать информацию об обнаруженных ошибках. Ссылки на материалы сайта отмечаются на полях следующим значком:

Об авторах

Рафаэл С. Гонсалес

Р. С. Гонсалес получил степень бакалавра в университете шт. Майами в 1965 г., а затем степени магистра и доктора философии по электротехнике в университете шт. Флорида (г. Гейнсвилл) в 1967 и 1970 гг. соответственно. В 1970 г. он поступил на работу на факультет электронной и компьютерной техники университета шт. Теннеси (UTK) в г. Ноксвилл, где последовательно получил звания адъюнкт-профессора (1973 г.), профессора (1978 г.) и профессора с особыми заслугами (1984 г.). С 1994 по 1997 гг. он был президентом факультета, а в настоящее время является почетным профессором UTK в отставке.

Гонсалес был основателем двух лабораторий UTK: Лаборатории анализа изображений и образов и Лаборатории робототехники и машинного зрения. Он также основал в 1982 г. компанию Perceptics Corporation и вплоть до 1992 г. был ее президентом. В течение трех последних лет этого периода он являлся сотрудником Westinghouse Corporation, которая приобрела компанию Perceptics в 1989 г.

Под руководством Гонсалеса компания Perceptics достигла больших успехов в обработке изображений, машинном зрении и технологии запоминающих устройств на основе лазерных дисков. За первое десятилетие своей работы компания Perceptics внедрила ряд новаторских изделий, в том числе: первую в мире коммерческую систему машинного зрения для автоматического чтения номерных знаков транспортных средств; ряд крупномасштабных систем обработки и хранения изображений, которые применяются на шести различных заводах ВМФ США при контроле реактивных двигателей ракет «Трайдент 2» для подводных лодок; семейство плат обработки изображений для модернизации компьютеров «Макинтош», занимающее передовые позиции на рынке; а также линейку запоминающих устройств на базе лазерных дисков терабайтной емкости.

Он часто привлекается промышленными предприятиями и правительством в качестве консультанта в области распознавания образов, обработки изображений и обучающихся машин. Его академические заслуги в этих областях отмечены в 1977 г. премией технического факультета UTK за высокие достижения; в 1978 г. премией ректора UTK для ученых-исследователей; в 1980 г. – премией им. М. Е. Брукса как выдающийся профессор и почетным званием профессора Magnavox Engineering. В 1981 г. он становится профессором IBM в университете шт. Теннеси, а в 1984 г. получил там звание профессора с особыми заслугами. Он отмечен званием выдающегося выпускника университета шт. Майами (1985 г.), премией научного общества «фи-каппа-фи» (1986 г.) и премией им. Натана В. Догерти университета шт. Теннеси за высокие достижения в области техники (1992 г.).

Среди наград за промышленные достижения – премия выдающемуся инженеру IEEE 1987 г. за коммерческие разработки в Теннеси; Национальная премия им. Альберта Роуза 1988 г. за успехи в создании коммерческих систем обработки

изображений; премия Б. Отто Уили за достижения в переносе технологий; звание «предприниматель года» агентства Куперс и Либранд в 1989 г.; премия выдающемуся инженеру IEEE 1992 г.; и Национальная премия 1993 г. за развитие технологии Ассоциации по автоматизированной обработке изображений.

Гонсалес является автором и соавтором свыше 100 технических статей, двух монографий и четырех учебников в области распознавания образов, обработки изображений и робототехники. Его книгами пользуются в более чем 1000 университетах и исследовательских институтах по всему миру. Он включен в престижные справочники «Кто есть кто в Америке», «Кто есть кто в технике», «Кто есть кто в мире», а также еще в 10 национальных и международных биографических перечней. Он совладелец двух патентов США и член редколлегий научных журналов «Труды по теории систем» (IEEE Transactions on Systems), «Человек и кибернетика» (Man and Cybernetics), Международного журнала по вычислительной технике и информатике (International Journal of Computer and Information Sciences). Он является членом IEEE и состоит в многочисленных профессиональных и почетных обществах (в том числе «тау-бета-пи», «фи-каппа-фи», «эта-каппа-ню» и «сигма-кси»).

Ричард Е. Вудс

Ричард Е. Вудс получил степени бакалавра, магистра и доктора философии по электротехнике в университете шт. Теннеси (г. Ноксвилл). Его профессиональный опыт охватывает широкий диапазон от предпринимательской деятельности до исполнения более традиционных обязанностей в академической, консультативной, правительственной и промышленной сферах. Недавно он основал работающую в области высоких технологий компанию MedData Interactive, которая специализируется на разработке портативных компьютерных систем медицинского назначения. Он также является соучредителем и вице-президентом компании Perceptics Corporation, отвечая в ней за разработку многих видов продукции для количественного анализа изображений и автономного принятия решений.

До работы в компаниях Perceptics и MedData д р Вудс был доцентом по электротехнике и вычислительной технике в университете Теннеси, а еще раньше — инженером по компьютеризации в компании Union Carbide Corporation. В качестве консультанта он привлекался к разработке ряда процессоров специального назначения для многочисленных космических и военных ведомств, включая агентство NASA, Центр управления баллистическими ракетами и Национальную лабораторию Оук Ридж (Oak Ridge).

Д-р Вудс опубликовал множество статей по цифровой обработке сигналов и состоит в ряде профессиональных обществ, включая «тау-бета-пи», «фи-каппа-фи» и IEEE. В 1986 г. он был признан выдающимся выпускником университета шт. Теннеси.

Предисловие научного редактора перевода

Тот факт, что не менее 90 % информации человек получает при помощи зрения, давно превратился в банальное утверждение, однако не потерял своей значимости. Более того, объемы зрительной информации, которую приходится получать, воспринимать и анализировать, постоянно возрастают. Все бóльшая роль при этом отводится техническим системам обработки и анализа видеоинформации — как с целью преобразования и видоизменения поступающих данных для последующего предоставления человеку, так и с целью автоматического их анализа и извлечения информации, в том числе для выбора последующей реакции.

Число областей, в которых исходные данные поступают в виде изображений, огромно. Здесь и различные системы наблюдения, технического зрения, мониторинга, видеотелефонии, которые регистрируют и передают огромные объемы видеоданных, и различные автономные системы (роботы), принимающие решение на основе анализа видеоинформации, и персонализированное телевизионное вещание, и полиграфия со значительно возросшим объемом иллюстраций в печатной продукции, и медицина, и Интернет, и многие другие направления. При этом наряду со значительным повышением уровня развития техники, самую существенную роль играют методы обработки видеоинформации. Они обеспечивают улучшение изображений для их наилучшего восприятия, сжатие видеоданных для хранения и передачи по каналам связи, а также анализ, распознавание и интерпретацию зрительных образов для принятия решений и управления поведением автономных технических систем.

Широкое распространение как профессиональных, так и бытовых средств фото- и видеорегистрации, а также достаточно мощных вычислительных машин, привело к появлению разнообразных программных средств обработки изображений. Они предоставляют богатые возможности преобразования и улучшения данных, но, как правило (кроме простейших случаев), не дают понимания того, какие именно операции над изображениями производятся. В такой ситуации у пользователя нет возможности ни объяснить выполняемые действия, ни быть уверенным, что при повторном их выполнении над другими данными будет получен соответствующий результат, ни составить правильный план последовательности необходимых действий для достижения какой-то поставленной цели. Решение этих задач возможно только на основе научного подхода.

Монография известных американских специалистов в области цифровой обработки изображений Р. Гонсалеса и Р. Вудса является введением в данную область знаний и создает основы для дальнейшего изучения этой многогранной и быстро развивающейся науки. В книге излагаются основы теории и методов преобразования и анализа видеоинформации, и она является одним из наиболее известных и распространенных за рубежом учебников в данной области науки. Многие из алгоритмов, которые в ней приведены, реализованы в различных широко используемых пакетах компьютерного редактирования изображений. В США данная книга вышла уже третьим изданием (первое появилось в 1992 г.).

Перед этим было два издания книги с тем же названием, написанной Гонсалесом и Винцем (в 1977 и в 1987 гг.).

Круг рассматриваемых в книге вопросов достаточно широк. Изучаются все основные направления обработки и анализа изображений, включая основы теории восприятия и регистрации видеоинформации, основные методы фильтрации и вейвлет-преобразования, методы улучшения, восстановления и сжатия черно-белых и цветных изображений, восстановления изображения по проекциям; также обсуждаются вопросы сегментации, распознавания образов, описания и представления деталей, морфологического анализа изображения. При этом необходимо отметить, что некоторые из вопросов, например, вопросы вейвлет-преобразования изображений или сопоставление разнообразных подходов к кодированию изображений и видеопоследовательностей до настоящего момента были опубликованы на русском языке лишь в виде отдельных разрозненных статей. Без сомнения, проведенные в книге рассмотрения не являются исчерпывающими в каждом из отдельных вопросов, однако они дают достаточно хорошее представление о сущности проблем и о наиболее распространенных способах их решения.

Все разделы книги содержат достаточно подробные обсуждения, приводится большое количество примеров и обработанных изображений, иллюстрирующих рассматриваемые методы. Оригинал книги сопровождается Интернет-сайтом (англоязычным), расположенным по адресу *http://www.imageprocessingplace.com* или *http://www.prenhall.com/gonzalezwoods* (адрес зеркального сайта). На нем представлены вспомогательные и дополнительные материалы, решения предложенных авторами задач, методические материалы по составлению учебных планов, а также база данных, содержащая многие из приведенных в книге иллюстраций.

Материал книги излагается на вполне доступном уровне. Предполагается, что читатель имеет математическую подготовку в объеме первых курсов технического вуза и знаком с основами математического анализа, линейной алгебры, теории вероятностей и математической статистики, а также владеет минимальными навыками программирования.

Данная монография создавалась авторами как введение в основные понятия и методы цифровой обработки изображений, дающее основы для последующего углубленного изучения проблематики обработки изображений, а также проведения самостоятельных исследований в этой области. В качестве стиля изложения авторы выбрали постепенный переход от простого к сложному и снабдили материал большим количеством задач, приведенных в конце каждой главы. Эти качества, вместе с полнотой охвата и достаточной подробностью рассмотрения изучаемых вопросов, позволяют рекомендовать эту книгу в качестве основы для построения учебного курса по теории и методам цифровой обработки изображений для студентов соответствующих специальностей. Данная книга будет полезна широкому кругу читателей: профессионалам в области анализа и переработки видеоинформации, студентам и аспирантам, специализирующимся в области информатики, а также всем интересующимся вопросами компьютерной обработки и анализа неподвижных изображений и видеопоследовательностей.

Следуя стилю авторов, мы сочли уместным привести список наиболее важных книг, изданных на русском языке, связанных с проблематикой обработки изображений. Данный список помещен в конце главы 1 вслед за аналогичным списком монографий, рекомендованных авторами. К сожалению, в нашей стране в последние годы заметно ощущается недостаток учебной литературы, отражающей как фундаментальные основы, так и изменения, происходящие в области цифровой обработки видеоинформации. Мы надеемся, что книга Р. Гонсалеса и Р. Вудса сможет восполнить возникший в отечественной учебной литературе дефицит книг из области теории и методов обработки изображений.

При переводе книги пришлось столкнуться с рядом трудностей, прежде всего терминологического характера; встречались также и не отмеченные авторами опечатки и неточности в формулировках. В русскоязычном издании устранены многие из них: как те, которые были отмечены авторами, так и те, которые встретились уже при переводе. Переводчиками была проделана большая работа по проверке значительной части формулировок, алгоритмов и выкладок. Насколько было возможно, переводчики старались сохранить стиль авторов; однако книга написана не строго формально, в результате чего в некоторых местах для достижения большей точности и ясности изложения приходилось несколько отклоняться от текста оригинала. В тех случаях, когда по мнению переводчиков полезным являлось дать некоторые разъяснения, были написаны дополнительные примечания. Нумерация всех примечаний сделана сплошная по каждой из глав: т. е. по умолчанию ссылка указывает на примечание внутри той же самой главы.

Предыдущее издание настоящей книги (в том числе и его перевод на русский язык) вызвало большой интерес читателей. Хочется надеяться, что перевод третьего издания также будет востребован.

Перевод книги выполнен канд. техн. наук Л. И. Рубановым (вводная часть, гл. 1, 2, 9—12) и канд. техн. наук П. А. Чочиа (гл. 3—8). Данный перевод в значительной степени основан на переводе второго издания той же книги. В нем, кроме двух указанных переводчиков, принимал участие канд. физ.-мат. наук Д. В. Сушко (гл. 4—7), который кроме того оказал существенную помощь при переводе гл. 5 настоящего издания.

П. А. Чочиа

ГЛАВА 1

ВВЕДЕНИЕ

Лучше один раз увидеть, чем сто раз услышать.
Пословица

Интерес к методам цифровой обработки изображений происходит из двух основных областей ее применения, которыми являются повышение качества изображений для улучшения его визуального восприятия человеком и обработка изображений для их хранения, передачи и представления в автономных системах машинного зрения. Эта глава преследует несколько целей: (1) определить границы области, называемой «обработка изображений»; (2) нарисовать историческую перспективу развития этой области; (3) дать представление о современном состоянии предмета, рассмотрев несколько важнейших областей, где применяется обработка изображений; (4) кратко обсудить основные подходы, используемые в цифровой обработке изображений; (5) дать общее представление о компонентах типичной многоцелевой системы обработки изображений и (6) указать книги и периодические издания, в которых публикуются работы по тематике обработки изображений.

1.1. Что такое цифровая обработка изображений?

Изображение можно определить как двумерную функцию $f(x, y)$, где x и y — координаты в *пространстве* (конкретно на плоскости) и значение f которой в любой точке, задаваемой парой координат (x, y), называется *интенсивностью* или *уровнем серого*[1] изображения в этой точке. Если величины x, y и f принимают конечное число дискретных значений, то говорят о *цифровом изображении*. *Цифровой обработкой изображений* называется обработка цифровых изображений с помощью цифровых вычислительных машин (компьютеров). Заметим, что цифровое изображение состоит из конечного числа элементов, каждый из которых расположен в конкретном месте и принимает определенное значение. Эти элементы называются *элементами изображения* или *пикселями*[2]. Чаще всего для элементов цифрового изображения используется термин «пиксель». Более строгие формальные определения будут даны в главе 2.

---

[1] В отечественной литературе для этого понятия широко распространен термин *яркость*, который мы и будем использовать во всех случаях, когда это не приводит к недоразумениям. — *Прим. перев.*

[2] Слово *pixel* образовано от английского словосочетания picture element — «элемент изображения». В англоязычной литературе также широко употребляется сокращенное написание *pel*. — *Прим. перев.*

Зрение является наиболее совершенным из наших органов чувств, поэтому неудивительно, что зрительные образы играют важнейшую роль в человеческом восприятии. Однако, в отличие от людей, способных воспринимать электромагнитное излучение лишь в видимом диапазоне, машинная обработка изображений охватывает практически весь электромагнитный спектр от гамма-излучения до радиоволн. Обрабатываемые изображения могут порождаться такими источниками, которые для человека непривычно связывать с наблюдаемыми изображениями. Таковы, например, ультразвуковые изображения, изображения, получаемые в электронной микроскопии или генерируемые компьютером. Таким образом, цифровая обработка изображений охватывает широкие и разнообразные области применения.

Не существует общепринятой точки зрения, где заканчивается обработка изображений и начинаются другие смежные области, например анализ изображений и машинное зрение. Иногда разграничение делается здесь по тому принципу, что обработка изображений определяется как дисциплина, в которой на входе и на выходе процесса присутствуют изображения. Мы убеждены, что такое вычленение является ограниченным и отчасти искусственным. Например, в рамках этого определения даже такая тривиальная задача, как определение средней интенсивности по полю изображения (при решении которой ищется единственное число), не может рассматриваться как операция обработки изображения. С другой стороны, существуют такие области, как машинное зрение, где конечной целью является компьютерная имитация человеческого зрения, включая обучение, способность к умозаключениям и действиям на основе наблюдаемой информации. Эта область сама по себе образует лишь одно из направлений искусственного интеллекта, целью которого является имитация интеллектуальной деятельности человека. Искусственный интеллект находится еще на начальной стадии своего развития, причем следует отметить, что прогресс идет значительно медленнее, чем первоначально ожидалось. Область, связанная с анализом изображений (иначе называемая «понимание» или «интерпретация» изображений), занимает промежуточное положение между обработкой изображений и машинным зрением.

Во всем диапазоне от обработки изображений до машинного зрения нет четких границ, тем не менее можно различать в нем компьютеризованные процессы низкого, среднего и высокого уровня. Процессы низкого уровня касаются только примитивных операций типа предобработки с целью уменьшения шума, повышения контраста или улучшения резкости изображений. Для низкоуровневых процессов характерен тот факт, что на входе и на выходе присутствуют изображения. Обработка изображений на среднем уровне охватывает такие задачи, как сегментация (разделение изображения на области или выделение на нем объектов), описание объектов и сжатие их в удобную для компьютерной обработки форму, а также классификация (распознавание) отдельных объектов. Для процессов среднего уровня характерно наличие изображений только на входе, на выход же поступают признаки и атрибуты, извлекаемые из этих изображений (например границы областей, линии контуров, отличительные признаки конкретных объектов). Наконец, высокоуровневая обработка включает в себя «осмысление» набора распознанных объектов, как это делается в анализе

изображений, и в пределе осуществление познавательных функций, которые принято связывать со зрением.

Исходя из вышеприведенного замечания мы видим, что естественным этапом перехода от обработки изображений к их анализу выступает распознавание отдельных областей или объектов на изображении. Таким образом, то, что в этой книге называется «цифровой обработкой изображений», включает процессы с изображениями на входе и на выходе, а также процессы извлечения признаков из изображений, вплоть до (или включая) распознавания индивидуальных объектов. В качестве иллюстрации, разъясняющей указанные понятия, рассмотрим область автоматизированного анализа печатного или рукописного текста. В сферу цифровой обработки изображений, которая рассматривается в данной книге, входят процессы получения изображения области, содержащей этот текст, предварительной обработки полученного изображения, выделения (сегментации) отдельных символов текста, описания символов в подходящей для компьютерной обработки форме и, наконец, распознавания этих символов. Что касается осмысления содержимого страницы, то оно может быть отнесено уже к сфере анализа изображений или даже машинного зрения, в зависимости от уровня сложности, который подразумевается за словом «осмысление». Как мы вскоре увидим, определенная таким образом цифровая обработка изображений успешно применяется в широком круге областей, важных с социально-экономической точки зрения. Концепции, развиваемые в последующих главах, служат основой для методов, которые используются в этих прикладных областях.

1.2. Истоки цифровой обработки изображений

Одно из первых применений цифровых изображений было опробовано в газетном деле для передачи иллюстраций по трансокеанскому подводному кабелю между Лондоном и Нью-Йорком. В начале 1920-х годов была внедрена система «Бартлейн» для передачи изображений по кабелю, что позволило уменьшить время доставки иллюстраций через Атлантику с обычной недельной задержки до менее чем трех часов. С помощью специального печатающего оборудования осуществлялось кодирование исходного изображения для передачи по кабелю и последующее восстановление этого изображения на приемной стороне. На рис. 1.1 показано изображение, переданное таким образом и распечатанное затем на телеграфном буквопечатающем аппарате со специальным шрифтом, имитирующим различные уровни почернения.

Ряд начальных проблем, связанных с улучшением визуального качества этих первых цифровых изображений, относился к выбору процедуры печати и распределению уровней интенсивности. Способ печати, использованный для получения рис. 1.1, в конце 1921 г. был отвергнут в пользу технологии фотографической репродукции с использованием перфоленты, формируемой телеграфным аппаратом на приемной стороне линии. На рис. 1.2 показано изображение, полученное таким способом. Улучшения по сравнению с рис. 1.1 очевидны как в отношении качества передачи полутонов, так и в пространственном разрешении.

Рис. 1.1. Цифровое изображение, полученное в 1921 г. с кодовой ленты на телеграфном аппарате с особым шрифтом [McFarlane[3]]

Ранние системы «Бартлейн» были способны кодировать изображения с помощью пяти градаций яркости. В 1929 г. эти возможности были увеличены до 15 градаций. На рис. 1.3 приведено типичное изображение, которое могло быть получено с помощью оборудования, использующего 15 градаций яркости. За этот период были внедрены системы, в которых фотопластинка экспонировалась с помощью световых лучей, модулируемых закодированной на перфоленте информацией, что позволило значительно улучшить процесс репродукции.

Хотя в вышеприведенных примерах фигурируют цифровые изображения, их нельзя рассматривать как результат цифровой обработки изображений в контексте нашего определения, коль скоро при их получении не использовались компьютеры. Таким образом, история цифровой обработки изображений тесно связана с развитием цифровой вычислительной техники. В самом деле, для цифровых изображений требуется такая большая память и вычислительная мощность, что прогресс в области цифровой обработки изображений в значительной степени определяется развитием компьютеров и вспомогательных технологий для хранения, отображения и передачи данных.

Идея компьютера восходит к абаку, изобретенному на полуострове Малая Азия более 5000 лет назад. Ближе к нашим дням, в последние два столетия делались продвижения, заложившие основу для появления компьютеров. Однако принципы того, что мы называем *современным* цифровым компьютером, были

Рис. 1.2. Цифровое изображение, полученное в 1922 г. с использованием перфоленты после прохождения сигнала через Атлантику дважды [McFarlane]

[3] Список литературы в конце книги расположен в алфавитном порядке по фамилии первого автора.

Рис. 1.3. Неретушированная фотография генералов Першинга и Фоша, переданная в 1929 г. по кабелю из Лондона в Нью-Йорк с помощью 15-градационного оборудования [McFarlane]

заложены всего лишь в 1940-х годах, когда Джон фон Нейман ввел в рассмотрение два ключевых понятия: (1) равноправное хранение в памяти и данных, и программ и (2) условный переход в программе. Эти две идеи заложены в фундамент центрального процессора, который является сердцем современных компьютеров. Вслед за фон Нейманом был целый ряд ключевых продвижений, которые привели к появлению компьютеров, достаточно мощных для использования при цифровой обработке изображений. Кратко перечислим эти достижения: (1) изобретение транзистора в компании Bell Laboratories в 1948 г.; (2) изобретение в 1950-х и 1960-х гг. языков программирования высокого уровня КОБОЛ (COBOL, Common Business-Oriented Language — общий язык программирования, ориентированный на бизнес) и ФОРТРАН (FORTRAN, Formula Translator — «транслятор формул»); (3) изобретение интегральной микросхемы компанией Texas Instruments в 1958 г.; (4) разработка операционных систем в начале 1960-х гг.; (5) выход на рынок персонального компьютера IBM в 1981 г. и (7) последовательная миниатюризация электронных компонентов, начиная с появления больших интегральных схем (БИС) в конце 1970-х гг., затем сверхбольших интегральных схем (СБИС) в 1980-х гг., вплоть до сегодняшних ультрабольших интегральных схем (УБИС). Одновременно с перечисленными продвижениями шло развитие в области запоминающих устройств и систем отображения, наличие которых необходимо для цифровой обработки изображений.

Первые компьютеры с мощностью, достаточной для выполнения осмысленных задач цифровой обработки изображений, появились в начале 1960-х гг. Рождение того, что мы сегодня называем цифровой обработкой изображений, прослеживается с момента возникновения таких машин и появления программ изучения космоса. Параллельный прогресс в этих двух областях привел в действие мощный потенциал идей цифровой обработки изображений. Работы по использованию вычислительной техники для улучшения визуального качества изображений, получаемых с помощью беспилотных космических аппаратов, были развернуты в Лаборатории реактивного движения в Пасадене, шт. Калифорния, в 1964 г., когда переданные космическим аппаратом «Рейн-

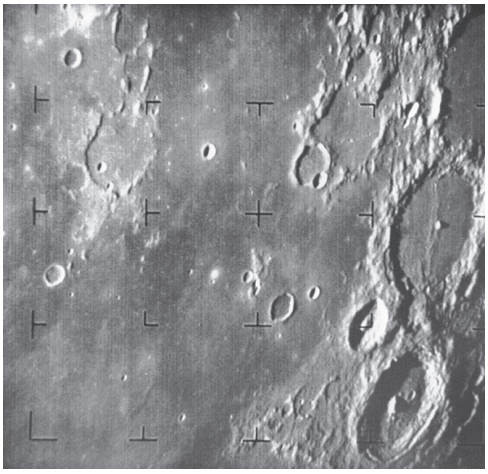


Рис. 1.4. Изображение лунной поверхности, переданное космическим аппаратом «Рейнджер-7» 31 июля 1964 г. (Снимок предоставлен агентством NASA)

жер-7» изображения лунной поверхности были подвергнуты компьютерной обработке для исправления различных искажений, обусловленных конструкцией бортовой телевизионной камеры. На рис. 1.4 приведено изображение Луны, полученное «Рейнджером-7» 31 июля 1964 г. в 9 час. 9 мин. восточно-американского времени (EDT), приблизительно за 17 мин. до его удара о лунную поверхность (на фотографии видны метки так называемой *ризо-маркировки*, применяемой для коррекции геометрических искажений; эта техника обсуждается подробнее в главе 2). Это было первое изображение Луны, полученное американским космическим аппаратом[4]. Опыт, накопленный при обработке первых космических изображений, послужил основой для разработки усовершенствованных методов восстановления и улучшения изображений. Эти методы позже применялись при обработке изображений, полученных в ходе полетов к Луне космических аппаратов «Сервейер», полетов аппаратов «Маринер» по пролетной траектории вблизи Марса, пилотируемых полетов космических кораблей «Аполлон» на Луну и т. д.

Параллельно с космическими исследованиями в конце 1960-х — начале 1970-х гг. методы цифровой обработки изображений начали применяться в медицине, дистанционном исследовании земных ресурсов, астрономии. В начале 1970-х гг. была изобретена рентгеновская вычислительная томография, кратко называемая также компьютерной томографией (КТ), что стало важнейшим событием в области применения обработки изображений для медицинской диагностики. При компьютерной томографии набор детекторов излучения и рентгеновский источник располагаются на кольце, внутрь которого помещается исследуемый объект (т. е. пациент), и кольцо вращается вокруг объекта. Проходящее через объект рентгеновское излучение улавливается детекторами,

[4] Напомним, что первые космические снимки лунной поверхности (в том числе обратной стороны Луны) были сделаны советской автоматической межпланетной станцией «Луна 3» в 1959 г. — *Прим. перев.*

находящимися на противоположной стороне кольца; этот процесс повторяется по мере вращения источника. Томография состоит из алгоритмов, которые на основе использования данных от детекторов строят изображения «сечения» объекта в плоскости кольца. При движении объекта вдоль оси кольца создается набор таких сечений, которые в совокупности образуют трехмерное представление внутреннего строения объекта. Томографию независимо друг от друга предложили сэр Годфри Н. Хаунсфилд и проф. Ален М. Кормак, которые в 1979 г. были удостоены за это изобретение Нобелевской премии по медицине. Интересно отметить, что рентгеновские лучи были открыты в 1895 г. Вильгельмом Конрадом Рентгеном, получившим за это Нобелевскую премию по физике в 1901 г. Эти два открытия, которые разделяет почти 100 лет, привели к одному из наиболее важных сегодняшних применений цифровой обработки изображений.

С 1960-х гг. до настоящего времени область применения обработки изображений значительно расширилась. Помимо медицинских и космических приложений, методы цифровой обработки изображений сегодня используются в широком круге областей. Компьютеризированные процедуры применяются для облегчения восприятия рентгеновских и иных изображений в промышленности, медицине и биологии путем повышения контраста или цветового кодирования различных уровней интенсивности (представления изображений в псевдоцветах). Аналогичные методы применяются в географии для изучения картины загрязнений окружающей среды по данным аэрофотосъемки и космическим снимкам. Методы улучшения и восстановления изображений применяются при обработке некачественных изображений утраченных объектов или трудновоспроизводимых экспериментальных результатов. Например, в археологии с помощью методов цифровой обработки изображений удалось по имеющимся нечетким фотографиям успешно восстановить первоначальный вид раритетов, которые со времени съемки были утрачены или повреждены. В физике и смежных областях компьютерная обработка является обычным способом улучшения качества изображений, получаемых в ходе экспериментов, как, например, в электронной микроскопии или физике высокотемпературной плазмы. Аналогичные примеры успешного применения технологий обработки изображений можно найти в астрономии, биологии, медицинской радиологии, промышленности, в оборонной и правоохранительной сфере.

Приведенные примеры относились к случаям, когда результаты обработки предназначены для восприятия человеком. Другая крупная область применения методов обработки изображений, упоминавшаяся в начале этой главы, — это решение задач, связанных с машинным восприятием изображений. В этом случае интерес вызывают процедуры, извлекающие из изображения некоторую информацию и представляющие ее в форме, подходящей для компьютерной обработки. Часто эта информация весьма мало похожа на визуальные признаки, используемые людьми при интерпретации содержимого изображения. Примерами информации такого рода, часто применяемой при машинном восприятии изображений, могут быть статистические моменты, коэффициенты преобразования Фурье, значения многомерного расстояния и т.д. Типичными задачами машинного восприятия, в которых интенсивно используются методы обработки изображений, являются автоматическое распознавание символов, системы

машинного зрения для автоматизации сборки и контроля продукции, задачи опознавания «свой—чужой» для военных объектов, автоматическая обработка отпечатков пальцев, проверка анализов крови и результатов рентгеновских исследований, компьютерная обработка аэрофотоснимков и спутниковых изображений с целью прогнозирования погоды и экологического мониторинга. Продолжающееся уменьшение соотношения цена/производительность современных компьютеров, рост пропускной способности сетей телекоммуникаций и развитие Интернета создали беспрецедентные возможности для дальнейшего расширения сферы цифровой обработки изображений. Некоторые из прикладных областей рассматриваются в следующем разделе.

1.3. Примеры областей применения цифровой обработки изображений

Сегодня в технике нет почти ни одной области, которую в той или иной мере не затрагивала бы цифровая обработка изображений. Наше обсуждение коснется лишь сравнительно небольшого числа таких прикладных областей, однако, несмотря на вынужденную ограниченность материала, у читателя не должно остаться сомнений в широте и важности применения обработки изображений. В этом разделе демонстрируются многочисленные прикладные области, в каждой из которых повседневно используются методы цифровой обработки изображений, описываемые в последующих главах. Многие из приведенных здесь изображений впоследствии упоминаются во встречающихся в книге примерах. Все эти изображения являются цифровыми.

Области применения цифровой обработки изображений столь разнообразны, что попытка охватить их во всей широте требует какой-то системы изложения. Один из простейших способов — классификация прикладных областей в соответствии с видами источников, формирующих соответствующие изображения (например оптические, рентгеновские и т. д.). Главным источником энергии для формирования применяемых сегодня изображений является электромагнитное излучение. Среди других важных энергетических источников, которые могут создавать изображения, упомянем акустические и ультразвуковые (механические) колебания, а также электронные пучки, применяемые в электронной микроскопии. Кроме того, имеется целый класс синтетических (искусственных) изображений, которые синтезируются компьютерными программами и используются для моделирования и визуализации. В этом разделе мы кратко обсудим, как регистрируются изображения этих многочисленных категорий и каковы области их применения. Методы преобразования изображений в цифровую форму будут рассмотрены в следующей главе.

Наиболее привычны изображения, создаваемые электромагнитным излучением, особенно в видимом спектре или в рентгеновском диапазоне. Электромагнитные волны можно трактовать как распространяющиеся синусоидальные колебания определенной частоты, а можно — как поток частиц, движущихся со скоростью света. Каждая такая частица обладает определенной энергией, но нулевой массой и называется *квантом излучения* (*фотоном*). Если располо-

Рис. 1.5. Спектр электромагнитного излучения в порядке убывания энергии фотона

жить диапазоны излучения в порядке убывания энергии фотона, то получим изображенный на рис. 1.5 спектр, простирающийся от гамма-лучей (обладающих максимальной энергией) до радиоволн. Плавное изменение окраски интервалов диапазона на этом рисунке призвано подчеркнуть тот факт, что спектр электромагнитного излучения не разграничен строго, а имеет смысл говорить скорее о плавном переходе одного участка диапазона в другой.

1.3.1. Формирование изображений с помощью гамма-лучей

Изображения, полученные с помощью гамма-излучения, используются главным образом в медицинской радиологии и астрономических наблюдениях. В медицинской радиологии применяется подход, при котором пациенту вводится радиоактивный изотоп, распад которого сопровождается гамма-излучением. Это излучение регистрируется детекторами гамма-излучения, сигналы которых и используются для формирования изображения. На рис. 1.6(*а*) приведен полный снимок скелета, полученный с помощью гамма-лучей описанным образом. Изображения такого вида используются для обнаружения участков различных патологий костей, в частности при инфекционных или онкологических заболеваниях. Рис. 1.6(*б*) демонстрирует другой важный вид медицинских радиологических изображений, получаемых методом позитронной эмиссионной томографии (ПЭТ). Используется тот же принцип, что и при рентгеновской томографии, кратко описанный в разделе 1.2, однако вместо использования внешнего источника рентгеновского излучения пациент принимает радиоактивный изотоп, распад которого сопровождается позитронным излучением. При встрече позитрона с электроном они аннигилируют с выделением двух гамма-квантов. Это гамма-излучение регистрируется, и формируется томографическое изображение в соответствии с основными принципами томографии. Приведенное на рис. 1.6(*б*) изображение представляет собой один кадр из последовательности изображений, которые в совокупности дают трехмерное представление тела пациента. На этом кадре хорошо видны небольшие белые скопления — опухоли в мозге и в легком пациента.

Около 15 тыс. лет назад в созвездии Лебедя произошел взрыв сверхновой, что привело к образованию расширяющегося облака газа сверхвысокой температуры, которое получило название Петли Лебедя. Столкновение этого облака с окружающими газовыми облаками порождает излучение широкого спектра, которое в видимом диапазоне дает эффектную цветовую картину. На рис. 1.6(*в*) приведено изображение Петли Лебедя в диапазоне гамма-излучения. В отличие от примеров, показанных на рис. 1.6(*а*, *б*), это

а *б*
в *г*

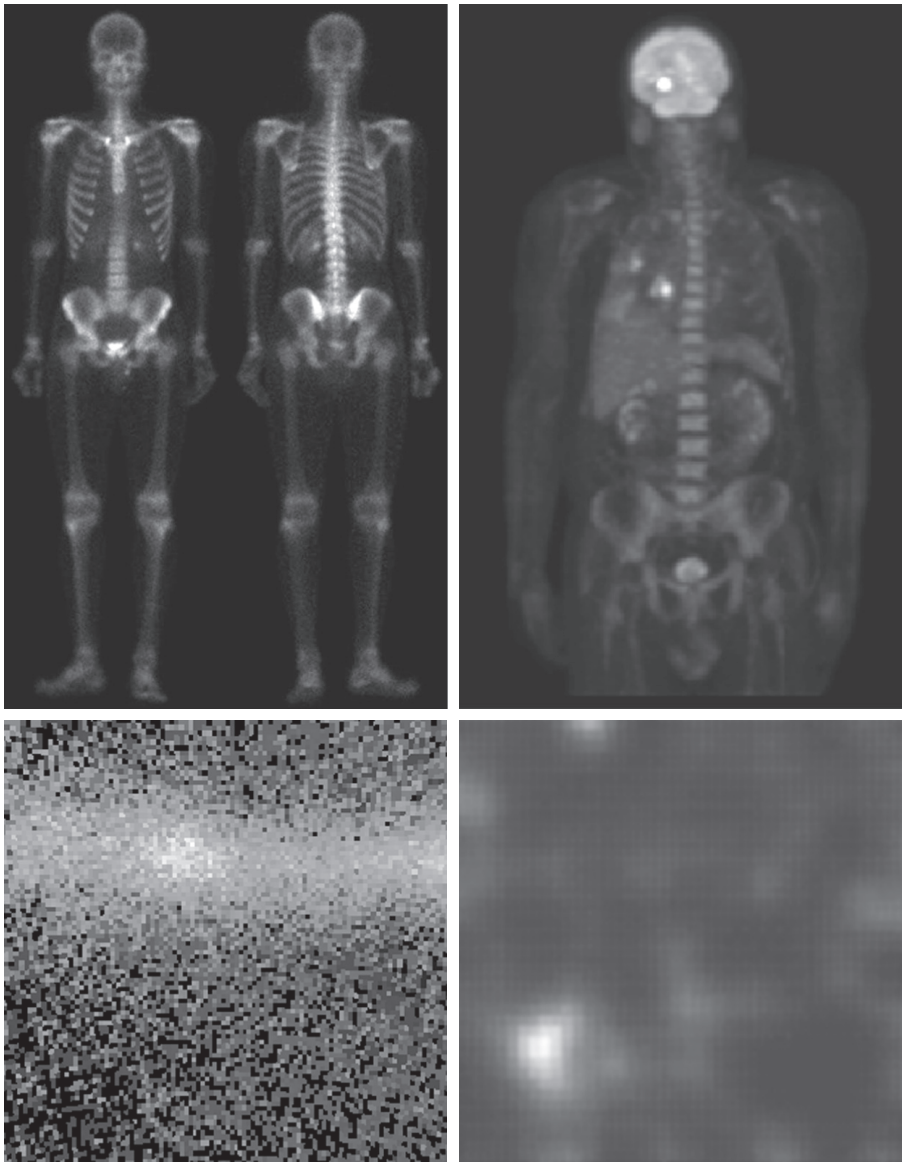


Рис. 1.6. Примеры изображений, полученных с помощью гамма-лучей. (*а*) Снимок скелета. (*б*) ПЭТ-изображение. (*в*) Петля Лебедя. (*г*) Гамма-излучение из клапана реактора (яркое пятно). (Изображения предоставили: (*а*) компания G.E. Medical Systems; (*б*) д-р Майкл Е. Кейси, компания CTI PET Systems; (*в*) профессора Жонг Хи и Дэвид К. Уи, университет шт. Мичиган)

изображение было получено с использованием естественного излучения изображаемого объекта. Наконец, на рис. 1.6(*г*) демонстрируется изображение гамма-излучения из клапана ядерного реактора. В левой нижней части изображения видна область сильной радиации.

1.3.2. Рентгеновские изображения

Рентгеновские лучи — один из самых старых источников электромагнитного излучения, используемых для получения изображений. Хорошо известно применение рентгеновских лучей для медицинской диагностики, однако они также широко используются в промышленности и других областях, в частности астрономии. Рентгеновское излучение для формирования изображений в медицине и промышленности генерируется с помощью рентгеновской трубки — вакуумного прибора с катодом и анодом. Катод находится в нагретом состоянии, вследствие чего испускает свободные электроны, которые с высокой скоростью летят к положительно заряженному аноду. При соударении электронов с ядрами атомов материала анода энергия выделяется в форме рентгеновского излучения. Энергия рентгеновских лучей (часто называемая «жесткостью»), определяющая их проникающую способность, регулируется изменением приложенного к аноду напряжения, а интенсивность излучения (количество рентгеновских лучей) регулируется изменением тока, проходящего через нить накала катода. На рис. 1.7(*а*) показан хорошо знакомый рентгеновский снимок грудной клетки, получаемый при помещении пациента между рентгеновской трубкой и чувствительной к рентгеновскому излучению пленкой. При прохождении рентгеновских лучей через тело пациента их интенсивность изменяется в зависимости от степени поглощения (рассеяния), и окончательный уровень энергии фиксируется на рентгеновской пленке, экспонируя ее почти так же, как лучи света формируют изображение на фотопленке. В цифровой рентгенографии применяются два способа получения цифровых изображений: (1) дискретизация (оцифровка) обычных рентгеновских пленок или (2) непосредственная регистрация прошедших через тело пациента рентгеновских лучей устройством, преобразующим рентгеновское излучение в световое (например с помощью фосфоресцирующего экрана). Полученный световой сигнал затем считывается с помощью цифровой системы, работающей в оптическом диапазоне. Вопросы дискретизации изображений подробнее рассматриваются в главах 2 и 4.

Другое важное применение рентгеновских изображений — ангиография, которая является одним из видов контрастной рентгенографии. Эта процедура используется для получения изображений кровеносных сосудов; такие изображения называются *ангиограммами*. В артерию или вену в паховой области вводится катетер (тонкая гибкая трубка), который продвигается вдоль сосуда, пока не достигнет обследуемой зоны. Затем через катетер впрыскивается контрастное вещество, хорошо поглощающее рентгеновские лучи. Благодаря этому усиливается контраст рентгеновского изображения кровеносных сосудов, что позволяет врачу-радиологу видеть аномалии кровоснабжения или места закупорки сосудов. На рис. 1.7(*б*) демонстрируется ангиограмма аорты, на которой виден катетер, введенный через крупный кровеносный сосуд внизу слева. Обратим внимание на высокий контраст изображения крупного сосуда в направлении потока контрастного вещества к почкам, которые тоже видны на изображении. Как описано в главе 2, в ангиографии широко используется цифровая обработка изображений, в частности вычитание изображений с целью дальнейшего повышения контраста исследуемых кровеносных сосудов.

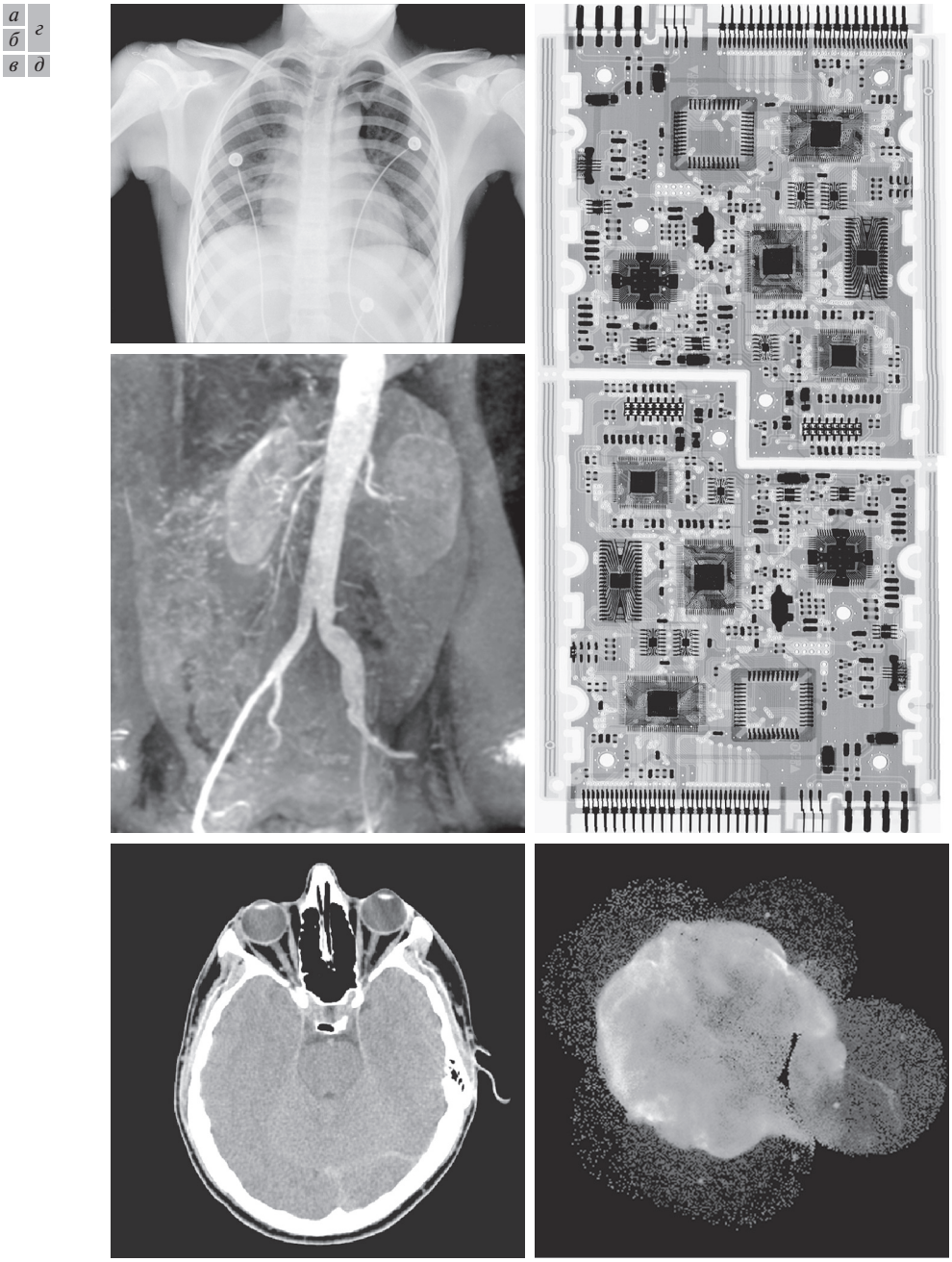


Рис. 1.7. Примеры рентгеновских изображений. (*а*) Рентгенограмма грудной клетки. (*б*) Ангиограмма аорты. (*в*) Компьютерная томограмма головы. (*г*) Печатные платы. (*д*) Петля Лебедя. (Изображения предоставили: (*а*, *в*) д-р Дэвид Р. Пикенс, департамент радиологии медицинского центра университета Вандербилта; (*б*) д-р Томас Р. Гест, отделение анатомии медицинской школы университета шт. Мичиган; (*г*) Джозеф Е. Пассенте, компания Lixi, Inc.; (*д*) агентство NASA)

Еще одно важное применение рентгеновских лучей для формирования изображений в медицине — это компьютерная томография. Благодаря высокому разрешению и возможности трехмерного представления, компьютерная томография с момента своего первого появления в начале 1970-х гг. произвела революцию в медицине. Как отмечалось в разделе 1.2, каждое КТ-изображение передает поперечный срез тела пациента. При продольном перемещении пациента формируется множество таких срезов, которые в совокупности образуют трехмерное представление внутреннего строения тела с продольным разрешением, пропорциональным количеству срезов. На рис. 1.7(*в*) показано изображение типичного среза, получаемого при компьютерной томографии головы.

Аналогичная технология используется и в промышленном производстве, хотя там обычно применяется рентгеновское излучение с большей энергией. На рис. 1.7(*г*) приведено рентгеновское изображение печатной платы радиоэлектронного прибора. Подобные изображения, будучи лишь одним из сотен возможных промышленных применений рентгеновских изображений, используются для контроля печатных плат на наличие дефектов, таких как отсутствие деталей или разрывы контактных дорожек. Промышленная компьютерная томография применима, когда детали проницаемы для рентгеновских лучей, что очевидно в случае пластмассовых узлов, но возможна даже и при контроле крупных изделий вроде твердотопливных реактивных двигателей. На рис.1.7(*д*) показан пример применения рентгеновских изображений в астрономии. Здесь изображена та же Петля Лебедя, что и на рис. 1.6(*в*), но на этот раз в диапазоне рентгеновского излучения.

1.3.3. Изображения в ультрафиолетовом диапазоне

Ультрафиолетовый «свет» находит разнообразные применения, в частности в литографии, производственном контроле, микроскопии, лазерной технике, биологических и астрономических наблюдениях. Мы проиллюстрируем использование изображений ультрафиолетового диапазона на примерах из области микроскопии и астрономии.

Ультрафиолетовое освещение используется во флуоресцентной микроскопии — одном из наиболее быстро развивающихся направлений микроскопии. Явление флуоресценции было открыто в середине XIX в., когда впервые было замечено, что минерал флюорит (плавиковый шпат) излучает свет при направлении на него ультрафиолетового излучения. Сами по себе ультрафиолетовые лучи невидимы, но при столкновении фотона ультрафиолетового излучения с электроном атома флуоресцентного материала электрон переходит на более высокий энергетический уровень. Последующее возвращение возбужденного электрона на нижний уровень сопровождается излучением фотона с меньшей энергией, что соответствует видимому (ближе к красному) диапазону спектра. Принцип работы флуоресцентного микроскопа заключается в облучении подготовленного препарата ярким активизирующим освещением и последующем выделении значительно более слабого флуоресцентного свечения. Таким образом, глаз наблюдателя или другой детектор будет воспринимать только вторичное излучение. Свечение флуоресцирующих участков должно наблюдаться на темном фоне, чтобы обеспечивался достаточный для их обнаружения кон-

а *б*
в

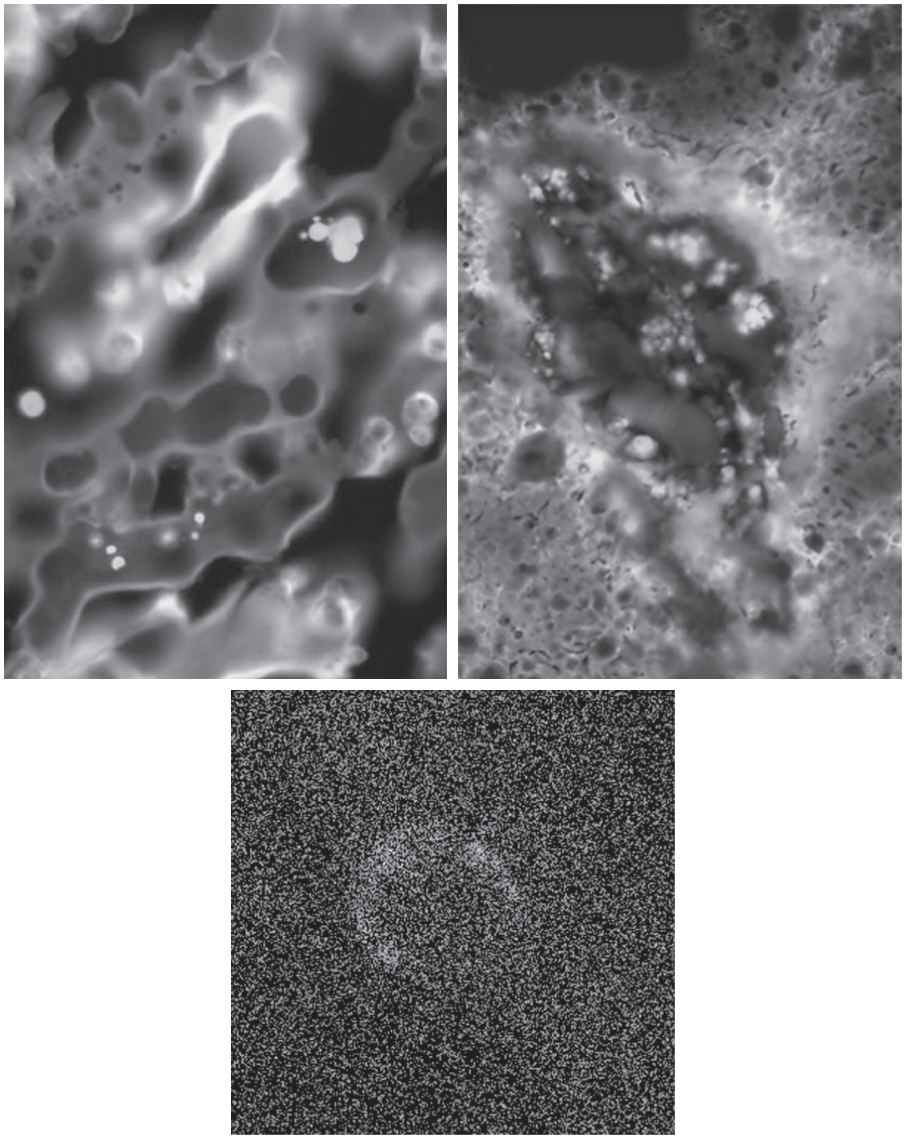


Рис. 1.8. Примеры изображений в ультрафиолетовом диапазоне. (*а*) Нормальное зерно. (*б*) Зерно, зараженное головней. (*в*) Петля Лебедя. (Изображения предоставили: (*а*, *б*) д-р Майкл У. Дэвидсон, университет шт. Флорида; (*в*) агентство NASA)

траст. Чем темнее фон, изготовленный из нефлуоресцирующего материала, тем выше эффективность прибора.

Флуоресцентная микроскопия — прекрасный метод исследования материалов, обладающих флуоресцирующими свойствами, либо в естественной форме (первичная флуоресценция), либо в результате обработки флуоресцирующими химикатами (вторичная флуоресценция). Рис. 1.8(*а*, *б*) демонстрируют типичные возможности флуоресцентной микроскопии. На рис. 1.8(*а*) по-

казан полученный с помощью флуоресцентного микроскопа снимок здорового зерна, а на рис. 1.8(*б*) — снимок зерна, зараженного головней — заболеванием зерновых и бобовых культур, трав, а также луковичных растений, вызываемым более чем 700 видами паразитических грибков. Особенно опасна головня злаковых культур, поскольку зерновые — один из важнейших источников пищи людей. В качестве иллюстрации другой области применения на рис. 1.8(*в*) показано изображение Петли Лебедя в высокочастотной полосе ультрафиолетового диапазона.

1.3.4. Изображения в видимом и инфракрасном диапазонах

Учитывая, что видимый диапазон электромагнитного спектра для нас наиболее привычен, неудивительно, что область использования изображений этого диапазона оказывается намного шире, чем всех остальных вместе взятых. Инфракрасные изображения часто используются совместно с видимыми, поэтому для иллюстрации мы объединили оба эти диапазона в одном разделе. В нижеследующем обсуждении в качестве примеров областей использования будут обсуждаться: световая микроскопия, астрономия, дистанционное зондирование, промышленность и правоохранительная деятельность.

На рис. 1.9 показано несколько примеров изображений, полученных с помощью оптического микроскопа. Диапазон примеров простирается от фармацевтики и микроскопических методов производственного контроля до определения характеристик материалов. Даже в пределах одной микроскопии множество возможных прикладных областей слишком обширно, чтобы его детально описать. Несложно вообразить варианты обработки, которые могут потребоваться в применении к таким изображениям, от улучшения их визуального качества до проведения различных измерений.

Еще одна важная область обработки видимых изображений — дистанционное зондирование земной поверхности, охватывающее обычно несколько зон в видимом и инфракрасном диапазонах спектра. В табл. 1.1 перечислены такие *тематические зоны*, в которых осуществляет зондирование спутник LANDSAT,

Таблица 1.1. Тематические зоны американского спутника LANDSAT

№	Наименование	Длины волн (мкм)	Характеристики и назначение
1	Видимый синий цвет	0,45—0,52	Максимальное проникание воды
2	Видимый зеленый цвет	0,52—0,60	Измерение плотности растительного покрова
3	Видимый красный цвет	0,63—0,69	Различение формы растительности
4	Ближнее ИК-излучение	0,76—0,90	Съемка береговой линии и распределения биомассы
5	Средний ИК-диапазон	1,55—1,75	Содержание влаги в почве и растительности
6	Тепловое ИК-излучение	10,4—12,5	Влажность почвы и температурная карта
7	Средний ИК-диапазон	2,08—2,35	Поиск полезных ископаемых

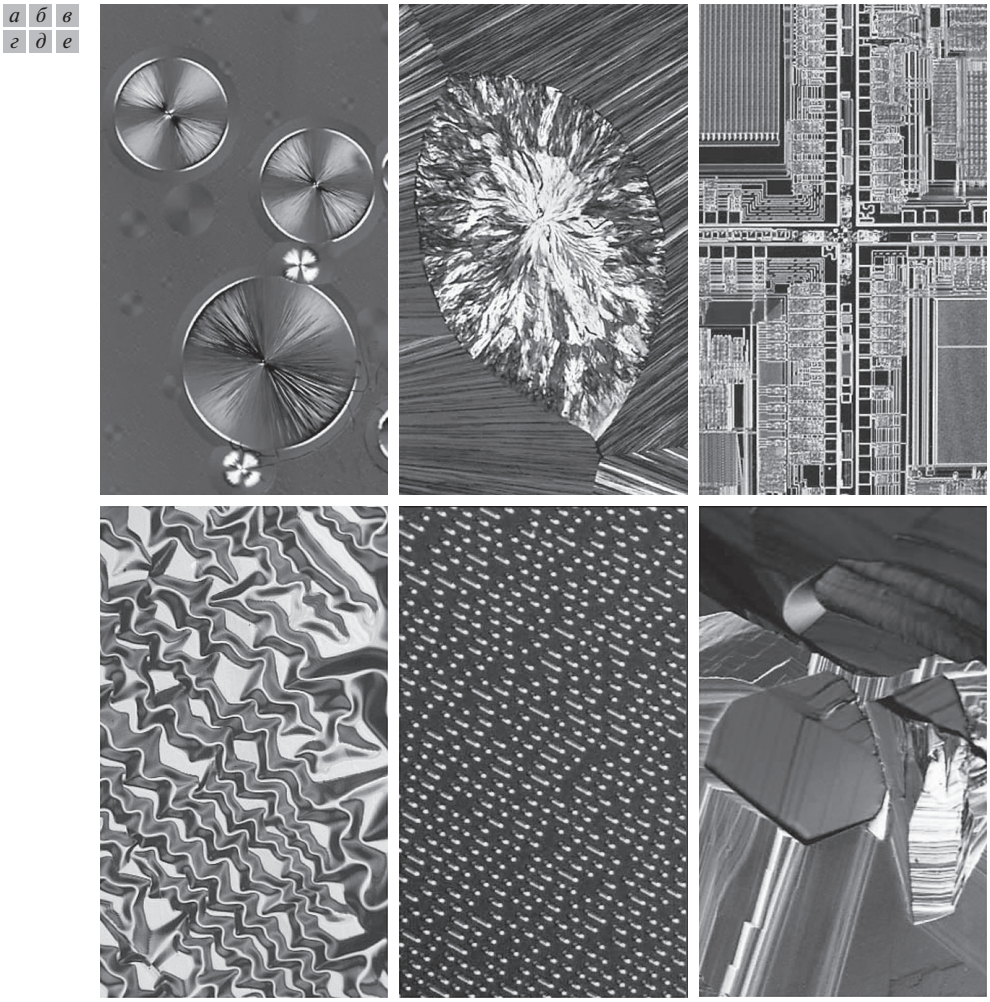


Рис. 1.9. Примеры изображений в оптической микроскопии. (*а*) Таксол (противораковый препарат), увеличение 250×. (*б*) Холестерин, увеличение 40×. (*в*) Микропроцессор, увеличение 60×. (*г*) Тонкая пленка окиси никеля, увеличение 600×. (*д*) Поверхность музыкального компакт-диска, увеличение 1750×. (*е*) Органический сверхпроводящий материал, увеличение 450×. (Изображения предоставил д-р Майкл У. Дэвидсон, университет шт. Флорида)

запущенный агентством NASA. Главная задача LANDSAT состоит в получении и передаче изображений Земли из космоса с целью глобального экологического мониторинга. Интервалы спектра представлены длинами волн в микронах (мкм); напомним, что 1 мкм = 10^{-6} м (длины волн, отвечающих различным диапазонам электромагнитного спектра, подробно обсуждаются в главе 2). Обратим внимание на указанные в таблице характеристики и назначение каждой из тематических зон.

Чтобы получить начальное представление о возможностях таких *многозональных* изображений, взглянем на рис. 1.10, где приведено по одному изображе-

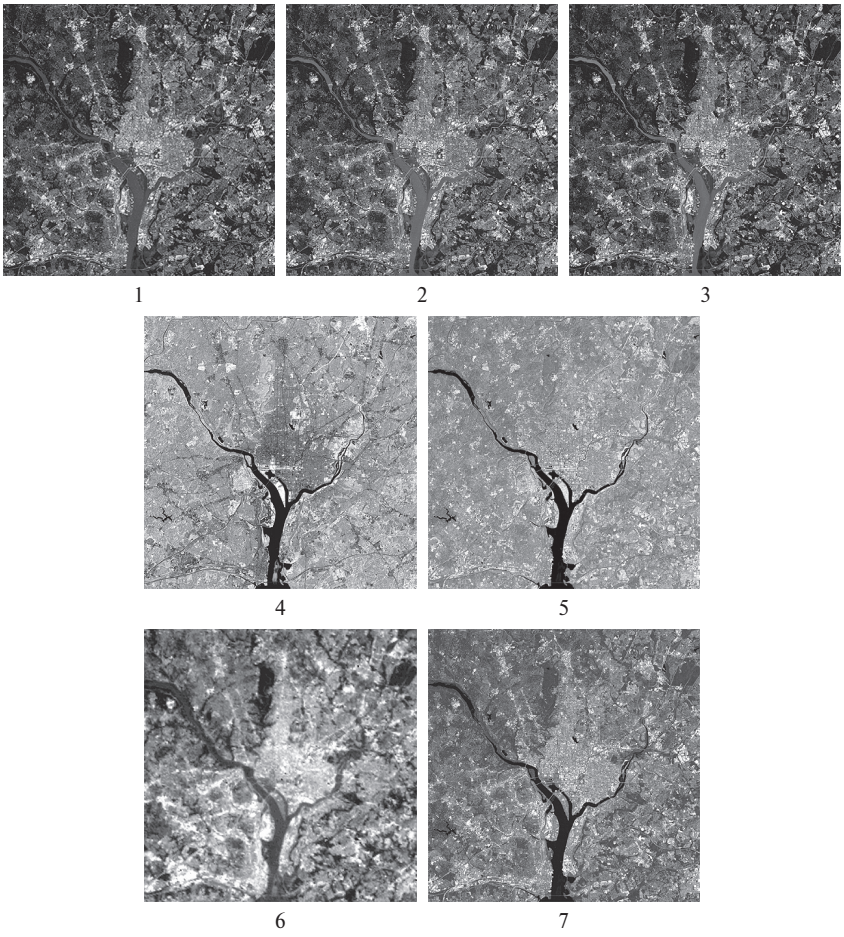


Рис. 1.10. Полученные со спутника LANDSAT изображения г. Вашингтон, округ Колумбия. Номера снимков соответствуют номерам тематических зон в табл. 1.1. (Изображения предоставлены агентством NASA)

нию для каждой из зон таблицы. Изображен район г. Вашингтон, округ Колумбия; видны здания, дороги, участки растительности и протекающая через город крупная река Потомак. Изображения населенных пунктов часто (и давно) используются для оценки роста численности населения, динамики загрязнений и прочих факторов, вредно влияющих на экологию. Примечательна разница между видимым и инфракрасным изображениями на этих снимках. Обратим, например, внимание, насколько хорошо выделяется река на фоне берегов в изображениях, соответствующих 4-й и 5-й зонам.

Наблюдение за погодой и составление прогнозов также является важным применением многозональных спутниковых изображений. Например, на рис. 1.11 приводится изображение урагана Катрина — одного из самых разрушительных ураганов в западном полушарии за последнее время. Оно было получено спутником Национальной океанографической и атмосферной службы США (NOAA) с помощью датчиков, работающих в видимом и инфракрасном

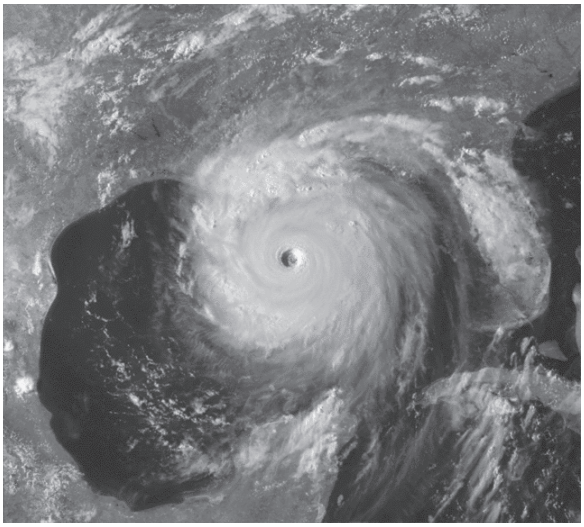


Рис. 1.11. Спутниковое изображение урагана Катрина, полученное 29 августа 2005 г. (Изображение предоставлено службой NOAA)

диапазонах. На этом снимке хорошо виден так называемый «глаз» урагана (соответствует центру циклона).

Рис. 1.12 и 1.13 демонстрируют применение инфракрасных изображений. Эти снимки были получены инфракрасной системой регистрации изображений, установленной на спутнике DMSP, запущенном по оборонно-метеорологической программе службы NOAA, и представляют собой фрагменты обширного набора данных «Ночные огни Земли» — глобального реестра населенных пунктов. Устройство регистрации инфракрасных изображений у этого спутника работает в диапазоне длин волн 10,0—13,4 мкм и обладает уникальной способностью фиксировать находящиеся на земной поверхности слабые источники ближнего инфракрасного излучения, в том числе города, поселки, деревни, газовые факелы и пожары. Даже не будучи специалистом в обработке изображений, легко представить себе компьютерную программу, которая использовала бы такие изображения для оценки относительной доли электроэнергии, потребляемой в различных районах Земли.

Важной областью применения изображений, регистрируемых в видимом диапазоне, является автоматический контроль выпускаемой продукции. На рис. 1.14 даны несколько примеров такого применения. Рис. 1.14(*а*) демонстрирует плату контроллера дисковода CD-ROM. Для подобных изделий типичной задачей обработки изображений может быть контроль наличия всех компонентов (на данном примере черный квадрат в правой верхней части изображения демонстрирует отсутствие микросхемы). На рис. 1.14(*б*) показана упаковка таблеток. Здесь задача состоит в компьютерном визуальном контроле отсутствия пустых мест в упаковке. Рис. 1.14(*в*) иллюстрирует пример обработки изображений для выявления недостаточно заполненных бутылок на производственной линии. На рис. 1.14(*г*) показана прозрачная пластмассовая деталь с недопустимым содержанием пузырьков воздуха. Обнаружение подобных аномалий составляет важную область промышленного контроля

Рис. 1.12. Инфракрасные спутниковые изображения американского континента. Для наглядности рядом приведена небольшая карта. (Изображение предоставлено службой NOOA)

различных материалов, например тканей и дерева. На рис. 1.14(*д*) изображен пакет кукурузных хлопьев, проходящий контроль по цвету и наличию брака в виде отдельных подгоревших хлопьев. Наконец, на рис. 1.14(*е*) показано изображение внутриглазного имплантата (вживляемого в глаз искусственного хрусталика) при специальном структурированном освещении. Эта техноло-

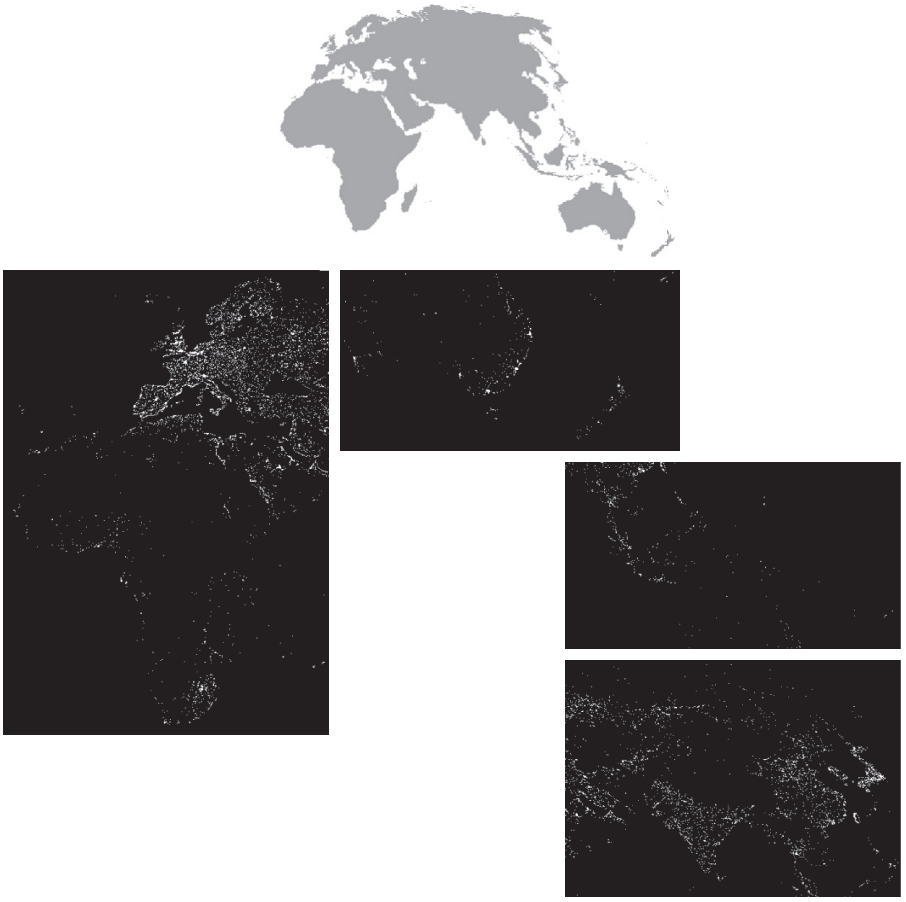
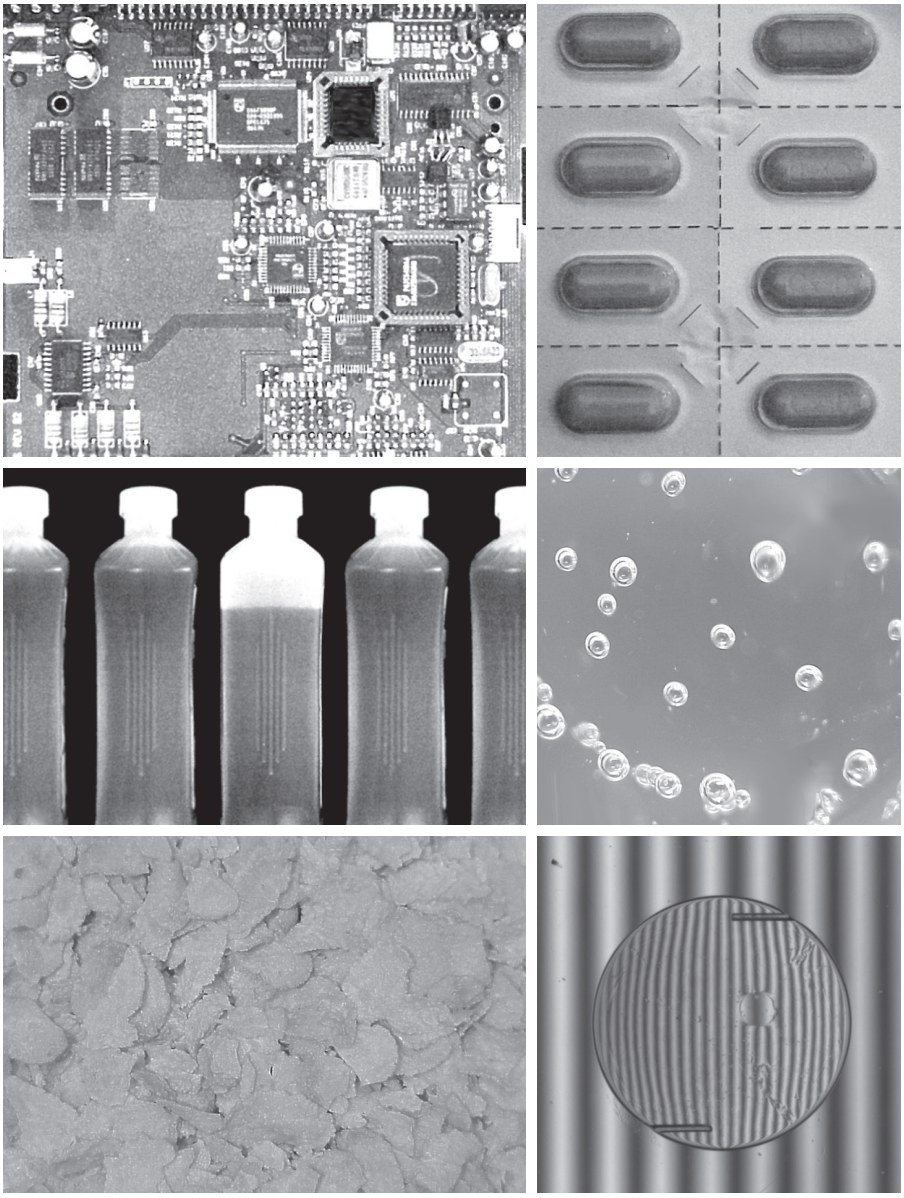


Рис. 1.13. Инфракрасные спутниковые изображения населенных регионов на других континентах. Для наглядности рядом приведена небольшая карта. (Изображение предоставлено службой NOOA)

гия применяется для простоты визуального обнаружения плоских деформаций имплантата. Следы в положениях «1 час» и «5 часов» (по аналогии с часовым циферблатом) — повреждения, оставленные пинцетом, а большинство других мелких пятнышек на изображении созданы пылинками и остатками материала. Цель данного вида контроля состоит в автоматическом обнаружении бракованных или поврежденных имплантатов перед упаковкой готовой продукции.

В качестве заключительной иллюстрации обработки изображений видимого спектра рассмотрим рис. 1.15. На рис. 1.15(*а*) изображен отпечаток большого пальца. Изображения отпечатков пальцев в массовом порядке подвергаются компьютерной обработке как с целью их улучшения, так и для поиска признаков, помогающих автоматически выбирать из базы данных похожие отпечатки. На рис. 1.15(*б*) приведено изображение бумажной купюры. Цифровая обработка таких изображений находит применение при автоматическом подсчете наличности и в правоохранительной деятельности для чтения номеров купюр с целью



а б
в г
д е

Рис. 1.14. Некоторые примеры промышленной продукции, часто контролируемой с помощью цифровой обработки оптических изображений. (*а*) Печатная плата контроллера. (*б*) Упаковка таблеток. (*в*) Бутылки. (*г*) Пузырьки воздуха в изделии из прозрачной пластмассы. (*д*) Кукурузные хлопья. (*е*) Изображение искусственного хрусталика. (Изображение (*е*) предоставил Пит Сайтс, Perceptics Corporation)

их прослеживания и идентификации. Два изображения транспортных средств, показанные на рис. 1.15(*в*, *г*), являются примерами автоматического чтения регистрационных номеров. Светлые прямоугольники указывают области, в кото-

Рис. 1.15. Некоторые дополнительные примеры обработки изображений видимого спектра. (*а*) Отпечаток пальца. (*б*) Бумажная купюра. (*в*, *г*) Автоматическое чтение номерных знаков. (Изображения предоставили: (*а*) Национальный институт стандартов и технологии США; (*в*, *г*) д-р Хуан Херрера, Perceptics Corporation)

рых система обработки изображений опознала номер транспортного средства, а в черных прямоугольниках выводятся результаты автоматического распознавания этой системой содержимого номера. Чтение автомобильных номеров и другие применения автоматического распознавания символов широко используются для контроля дорожного движения и надзора правоохранительных органов.

1.3.5. Изображения в микроволновом диапазоне

Изображения микроволнового диапазона применяются главным образом в радиолокации. Уникальным качеством радиолокации является возможность получения изображения любого района независимо от условий освещения и погоды. Микроволновое излучение некоторых диапазонов способно проникать

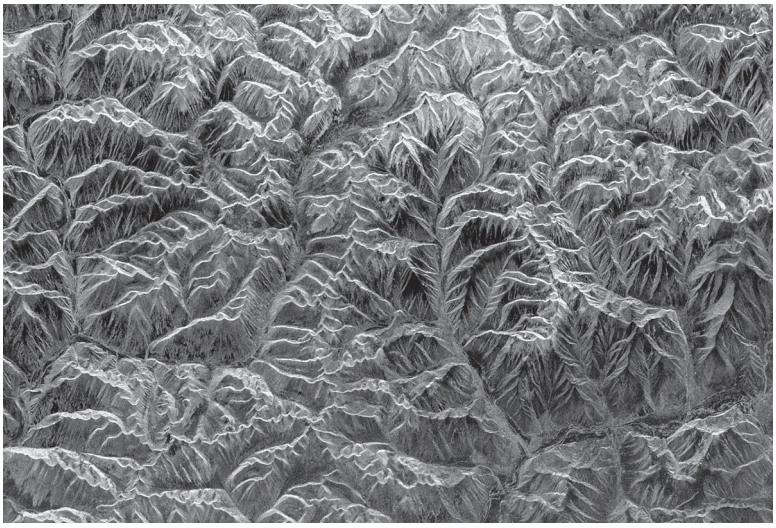


Рис. 1.16. Космическое радиолокационное изображение горного массива на юго-востоке Тибета. (Изображение предоставлено агентством NASA)

даже сквозь облака, растительность, лед и сухой песок. Во многих случаях радиолокация остается единственным способом исследования труднодоступных районов Земли. Применяемый для получения изображения радиолокатор работает аналогично фотоаппарату со вспышкой, в том смысле, что он использует собственный источник освещения (микроволновые импульсы), которое направляется на снимаемый участок поверхности. Роль объектива фотоаппарата в радиолокаторе играет антенна, сигнал от которой проходит через компьютерную систему, осуществляющую регистрацию и обработку изображения. Радиолокационное изображение отображает распределение интенсивностей отраженной энергии микроволнового диапазона, которую уловила антенна локатора.

На рис. 1.16 показано полученное из космоса радиолокационное изображение труднодоступного горного массива в юго-восточном Тибете, приблизительно в 90 км к востоку от г. Лхаса. В правом верхнем углу видна широкая долина реки Лхаса, населенная тибетскими фермерами, разводящими яков; в этой долине расположена деревня Менба. Высота гор в этом районе достигает 5800 м над уровнем моря, а дно долины располагается на высоте около 4300 м. Обратим внимание на четкость изображения и качество воспроизведения деталей, независимо от облаков и других атмосферных неоднородностей, которые обычно мешают получить изображение сходного качества в оптическом диапазоне.

1.3.6. Изображения в диапазоне радиоволн

Как и в случае изображений, получаемых на противоположной стороне электромагнитного спектра (гамма-лучи), основными областями применения изображений в диапазоне радиоволн выступают медицина и астрономия. В меди-

а *б*

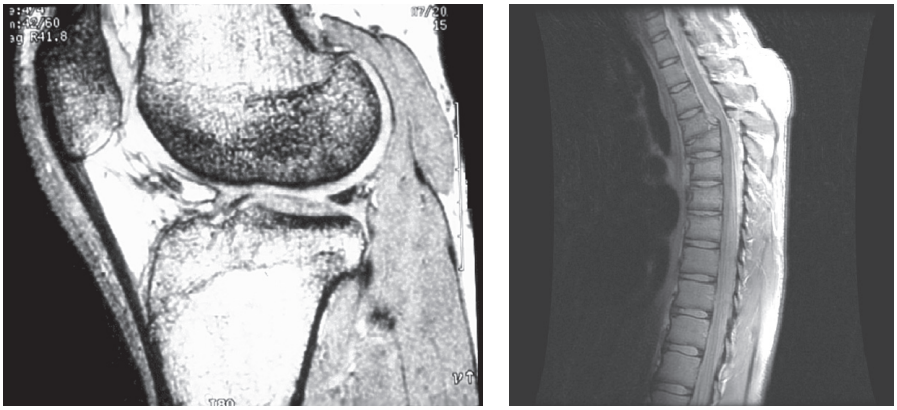


Рис. 1.17. ЯМР-изображения человеческого (*а*) колена и (*б*) позвоночника. (Изображения предоставили: (*а*) д-р Томас Р. Гест, отделение анатомии медицинской школы университета шт. Мичиган; (*б*) д-р Дэвид Р. Пикенс, департамент радиологии медицинского центра университета Вандербилта)

цине радиоволны используются для получения изображений методом ядерного магнитного резонанса (ЯМР). По этой технологии пациента помещают в сильное магнитное поле, и через его тело пропускают радиоволны в форме коротких импульсов. В ответ на каждый такой импульс ткани тела пациента реагируют, излучая свои радиоволновые сигналы. Место возникновения и сила этих сигналов регистрируются компьютерной системой обработки, генерирующей двумерное изображение среза тела пациента. С помощью ЯМР можно получить срез вдоль любой плоскости. На рис. 1.17 показаны ЯМР-изображения человеческого коленного сустава и позвоночника.

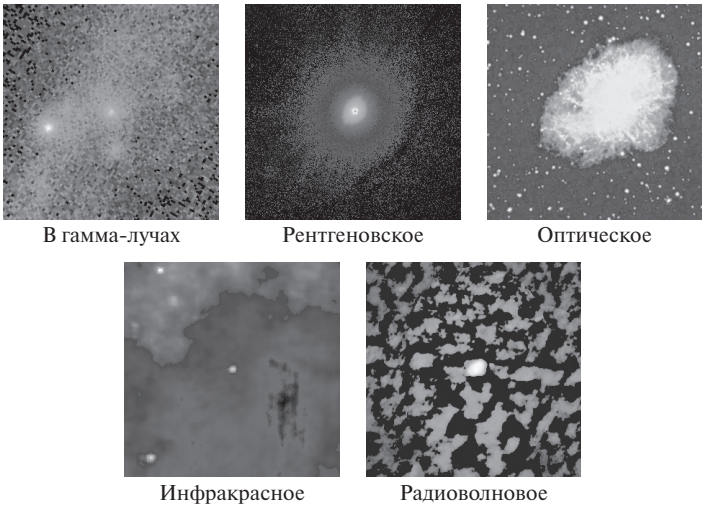


Рис. 1.18. Изображения пульсара Крабовидной туманности (находится в центре снимков) в различных диапазонах электромагнитного спектра. (Изображения предоставлены агентством NASA)

Крайний справа снимок на рис. 1.18 демонстрирует изображение пульсара Крабовидной туманности в диапазоне радиоволн. Интересно сравнить его с приведенными на этом рисунке изображениями того же района, полученными в других обсуждавшихся ранее диапазонах электромагнитного спектра. Заметим, что каждое изображение дает свой, совершенно отличающийся вид этого пульсара.

1.3.7. Примеры, иллюстрирующие другие способы формирования изображений

Хотя чаще всего используются изображения, полученные в электромагнитном спектре, существует ряд других важных способов формирования изображений. В частности, в этом разделе мы обсудим акустические изображения, электронную микроскопию и искусственные изображения, синтезированные с помощью компьютеров.

Построение изображений с помощью звуковых волн находит применение в геологических изысканиях, промышленности и медицине. В геологии используются звуковые колебания с частотами у нижней границы звукового спектра (до сотен герц), а в других областях для получения изображений применяются ультразвуковые колебания с частотами порядка мегагерц (миллионов герц). Наиболее важные коммерческие применения обработки изображений в геологии касаются поисков нефти и других полезных ископаемых. При формировании изображения, несущего информацию о земных недрах, один из основных методов состоит в использовании тяжелого грузовика и большой плоской стальной платформы. Грузовик давит на землю через платформу и одновременно является источником вибраций в спектре частот до 100 Гц. Мощность и скорость распространения отраженных звуковых волн определяются геологическим составом грунта под поверхностью. В результате компьютерного анализа этих звуковых колебаний строится цифровое изображение.

В морской геологии для получения изображений в звуковом диапазоне обычно используют источник энергии в виде пары пневмопушек, буксируемых позади судна. Отраженные звуковые волны детектируются гидрофонами, помещенными внутрь кабелей, которые либо также буксируются судами, либо укладываются на океанское дно или вертикально подвешиваются к буям. Пневмопушки поочередно выстреливают, создавая импульс давления порядка 150 атмосфер. Картина отраженных звуковых волн в совокупности с постоянным движением судна (что дает продольную составляющую) используется для формирования трехмерной карты состава земной коры под дном океана.

На рис. 1.19 изображено поперечное сечение известной трехмерной модели, на которой проверяются характеристики алгоритмов построения изображений по данным сейсморазведки. Стрелка указывает на углеводородный пласт (место залегания нефти и/или газа). Указываемое место выглядит ярче окружающих пластов, поскольку изменения плотности в этом месте оказываются выше. Анализ изображений при сейсморазведке состоит в поиске подобных «ярких пятен», соответствующих вероятным нефтяным и газовым месторождениям. Вышележащие пласты также выделяются своей яркостью, однако в этих случаях изменения яркости в поперечном направлении не столь сильны. Многие

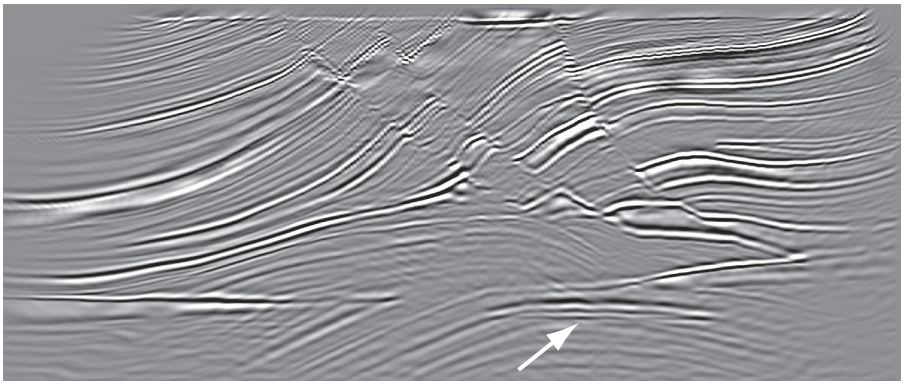


Рис. 1.19. Поперечное сечение модели данных сейсморазведки. Стрелкой указан пласт углеводородов (место залегания нефти и/или газа). (Изображение предоставил д-р Кертис С. Обер, Sandia National Laboratories)

алгоритмы реконструкции данных сейсморазведки испытывают трудности при обнаружении отмеченной области именно из-за ошибок, возникающих в вышележащих областях.

Хотя ультразвуковые изображения широко используются в промышленности, наиболее известно применение этой технологии в медицине, особенно в акушерстве, где изображения еще не рожденных детей изучаются на предмет отсутствия аномалий их развития. Дополнительным результатом такого исследования является определение пола будущего ребенка. Ультразвуковые изображения формируются следующим образом:

1. Ультразвуковая система (состоящая из компьютера, ультразвукового зонда с излучателем и приемником и дисплея) передает в тело ультразвуковые импульсы высокой частоты (от 1 до 5 МГц).
2. Звуковые волны проходят сквозь тело пациента, и на границах между тканями (например между жидкостью и мягкой тканью, мягкой тканью и костью скелета) происходит частичное отражение. Часть звуковых волн отражается обратно в сторону зонда, часть волн затухает, а остальные распространяются дальше, пока не достигнут следующей границы раздела и снова частично отразятся и т. д.
3. Отраженные волны улавливаются приемником зонда и передаются в компьютер.
4. Исходя из времени прихода каждого эхо-сигнала и известной скорости звука в тканях (1500 м/с), компьютер вычисляет расстояние от зонда до соответствующей границы ткани или внутреннего органа.
5. Вычисленные расстояния и интенсивности принятых отраженных сигналов выводятся на дисплее в виде двумерного изображения.

В типичных ультразвуковых исследованиях ежесекундно генерируются и принимаются миллионы звуковых импульсов и эхо-сигналов. Зонд можно двигать вдоль поверхности тела и наклонять, получая изображения в различных проекциях. На рис. 1.20 приводятся несколько примеров таких изображений.

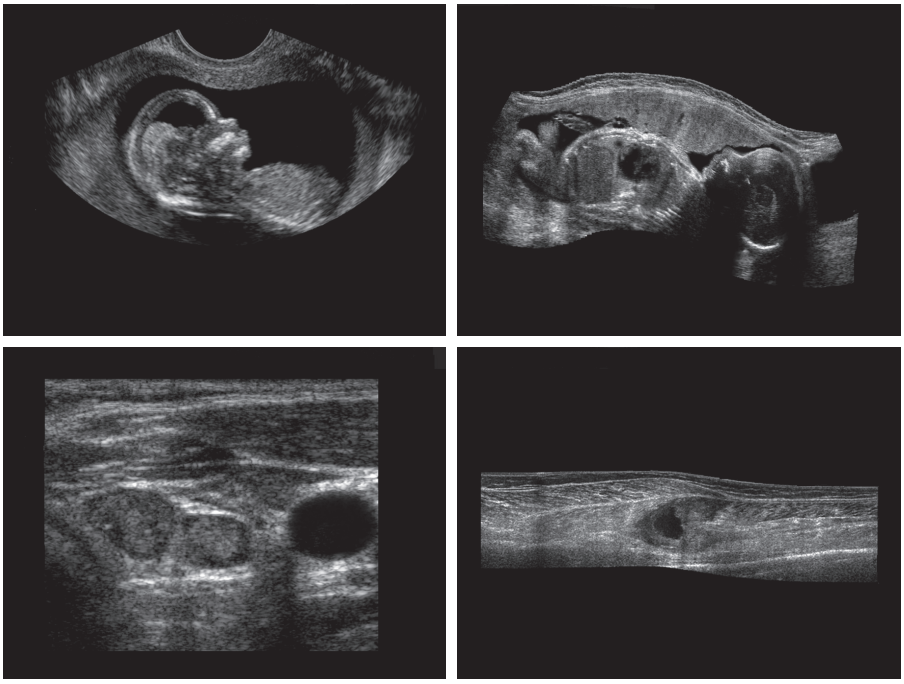


Рис. 1.20. Примеры ультразвуковых изображений. (*а*) Ребенок. (*б*) Ребенок в другой проекции. (*в*) Щитовидная железа. (*г*) Мышечные слои с заметным повреждением. (Изображения предоставила группа по ультразвуку компании Siemens Medical Systems, Inc.)

Мы продолжим обсуждение других способов получения изображений на примерах из электронной микроскопии. Электронный микроскоп действует аналогично оптическому с той разницей, что вместо световых лучей для получения изображения исследуемого объекта применяется сфокусированный пучок электронов. Работа электронного микроскопа складывается из следующих основных шагов. Источник испускает поток электронов, которые, благодаря приложенному положительному напряжению, движутся с ускорением в направлении исследуемого образца. С помощью металлических щелевых диафрагм и магнитных линз этот поток ограничивается и фокусируется, образуя тонкий пучок, сфокусированный на образце. Внутри облучаемого образца происходит взаимодействие, оказывающее влияние на прохождение пучка электронов, эффект от которого обнаруживается и преобразуется в изображение аналогично тому, как свет отражается или поглощается объектами наблюдаемой сцены. Перечисленные основные шаги имеют место во всех электронных микроскопах независимо от их типа.

Принцип работы *просвечивающего электронного микроскопа* (ПЭМ) во многом аналогичен проектору слайдов. Проектор направляет (передает) пучок света на слайд; при проходе этого пучка сквозь слайд свет модулируется содержимым слайда. После этого пучок света проецируется на экран, формируя увеличенное изображение слайда. ПЭМ работает точно так же, за исключением того, что сквозь образец, играющий роль слайда, направляется пучок электронов. Часть

пучка, прошедшая сквозь исследуемый образец, проецируется на экран из фосфоресцирующего материала. Взаимодействие электронов с этим материалом приводит к появлению света и, следовательно, видимого изображения.

Сканирующий (*растровый*) *электронный микроскоп* (СЭМ или РЭМ) осуществляет действительное сканирование образца электронным пучком и запись отраженного результата взаимодействия электронного пучка с каждой точкой поверхности образца. Полное изображение формируется путем растрового сканирования образца электронным пучком аналогично телевизионной развертке. Отраженные электроны при попадании на фосфоресцирующий экран создают на нем видимое изображение. СЭМ лучше подходят для объемных образцов, тогда как для ПЭМ необходим очень тонкий образец.

Электронные микроскопы способны дать очень большое увеличение. Если в оптической микроскопии кратность увеличения ограничена приблизительно 1000×, то в электронной микроскопии достигается увеличение 10000× и более. На рис. 1.21 показаны полученные с помощью СЭМ два изображения образцов с дефектами, возникшими вследствие температурных перегрузок.

Завершим обсуждение различных способов формирования изображений, кратко рассмотрев изображения, полученные не от какого-то физического объекта или явления, а сгенерированные компьютером. Примечательным примером изображений, синтезированных с помощью компьютеров, являются *фракталы* [Lu, 1997]. По существу, фракталы представляют собой не что иное, как повторяющееся воспроизведение некоторого исходного образа по определенным математическим правилам. Например, *мозаика* из квадратных элементов является одним из простейших способов генерации фрактальных изображений. Квадрат можно разделить на четыре квадратные подобласти, каждую из которых, в свою очередь, можно разбить снова на четыре еще более мелких квадрата и так далее. В зависимости от сложности правил заполнения каждой

а *б*

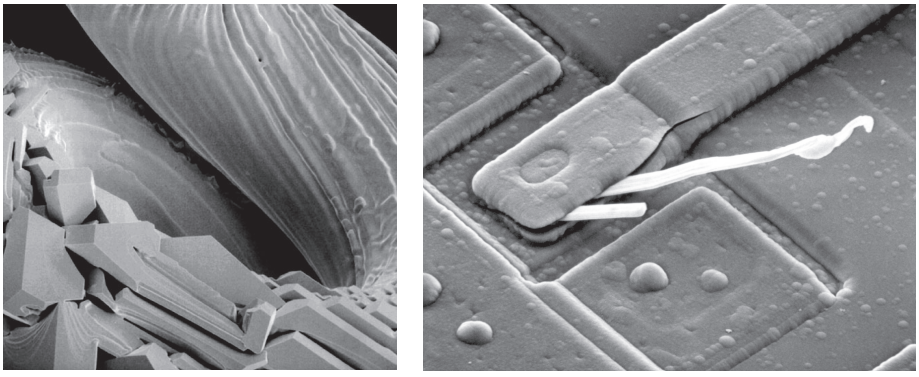
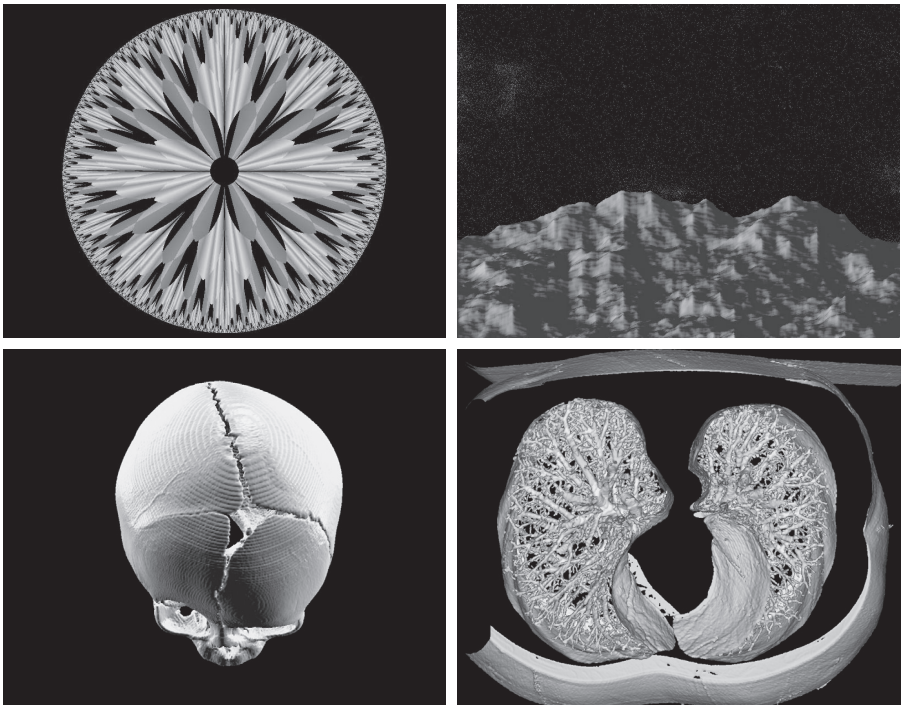


Рис. 1.21. (*а*) Изображение поврежденной при перегреве вольфрамовой нити накаливания, полученное на СЭМ при увеличении 250× (обратите внимание на осколки в левом нижнем углу). (*б*) СЭМ-изображение вышедшей из строя интегральной микросхемы при увеличении 2500×. Белые нити — это полосы окалины, возникшие при тепловом разрушении. (Изображения предоставили: (*а*) Майкл Шаффер, факультет геологии университета шт. Орегон, г. Юджин; (*б*) д-р Джон М. Хьюдек, университет Мак-Мастер, г. Гамильтон, шт. Онтарио, Канада)



а *б*
в *г*

Рис. 1.22. (*а*, *б*) Фрактальные изображения. (*в*, *г*) Изображения, построенные по трехмерным компьютерным моделям показанных объектов. (Изображения предоставили: (*а*, *б*) Мелисса Д. Байнд, Swarthmore College; (*в*, *г*) агентство NASA)

квадратной подобласти, таким способом могут быть получены некоторые красивые мозаичные изображения. Разумеется, геометрия может быть произвольной. Например, фрактальное изображение может вырастать из одной центральной точки — такой пример изображен на рис. 1.22(*а*). Рис. 1.22(*б*) демонстрирует другое фрактальное изображение («лунный ландшафт»), являющееся интересной аналогией некоторых космических снимков, приведенных в качестве иллюстрации в предыдущих разделах.

Фрактальные изображения, при всей их художественности, несут в себе элементы математического описания процесса «выращивания» изображения из его более мелких элементов в соответствии с некоторыми правилами. Иногда они находят применение в качестве случайных текстур. Более систематический подход к генерации изображений с помощью компьютеров состоит в построении трехмерных моделей объектов. Именно эта важная область, находящаяся на стыке обработки изображений и компьютерной графики, является основой для построения множества систем трехмерной визуализации (например авиационных тренажеров). Рис. 1.22(*в*, *г*) демонстрируют примеры изображений, синтезированных с помощью компьютерных программ. Поскольку объекты моделировались как трехмерные, с помощью плоской проекции трехмерного объема можно построить изображения этих объектов в любом ракурсе. Изображения подобного вида используются в медицинском образовании, а также как

основа для решения других прикладных задач, например в криминалистике, судебной медицине или для создания спецэффектов.

1.4. Основные стадии цифровой обработки изображений

Было бы полезно разделить изложенный в последующих главах материал на две большие категории, упомянутые в разделе 1.1: методы, в которых на входе и на выходе имеются изображения, и методы, где на вход поступают изображения, а на выходе возникают признаки и атрибуты, выделенные на основании этих изображений. Такая организация материала книги сведена в схему, изображенную на рис. 1.23. Эта схема не подразумевает, что к изображению применяется каждый из описанных процессов, напротив, целью было донести принципы всех методов обработки, которые могут применяться к изображениям в различных целях и, возможно, с различными получаемыми результатами. Проводимое в данном разделе обсуждение можно рассматривать как краткий обзор материала, представленного в остальной части книги.

Регистрация изображения — первый из процессов, показанных на рис. 1.23. Обсуждение, проведенное в разделе 1.3, дает некоторую информацию относительно возможных источников цифровых изображений, однако значительно более подробно эта тема рассматривается в главе 2, где также вводится ряд базовых понятий, относящихся к цифровым изображениям и используемых далее

Рис. 1.23. Основные стадии цифровой обработки изображений. Внутри блоков указаны главы, в которых рассматривается соответствующий материал

на протяжении всей книги. Заметим, что регистрация изображения может оказаться предельно простой, как в случае, когда исходное изображение уже представлено в цифровой форме. В общем случае стадия регистрации изображения включает некоторую предобработку, например масштабирование.

Улучшение изображения — это процесс манипулирования изображением, в результате которого оно становится более подходящим для конкретного применения, чем оригинал. Здесь важно слово «конкретного», поскольку оно с самого начала устанавливает, что методы улучшения изображений являются проблемно-ориентированными. Так, например, метод, который весьма полезен для улучшения рентгеновских изображений, может оказаться не лучшим подходом для улучшения спутниковых изображений, снятых в инфракрасном диапазоне электромагнитного спектра.

Общей «теории» улучшения изображений не существует. Если изображение обрабатывается с целью визуальной интерпретации, то оценку, насколько хорошо работает конкретный метод, дает в конечном счете наблюдатель. Методы улучшения настолько разнообразны и используют так много различных подходов к обработке изображения, что трудно собрать осмысленную совокупность подходящих для улучшения методов в одной главе, не проводя отдельное обширное исследование. Для новичков в области обработки изображений все связанное с улучшением изображений представляется визуально привлекательным, интересным и относительно простым для понимания. По указанным выше причинам мы используем улучшение изображений в качестве примеров, когда вводим в рассмотрение новые концепции в части главы 2, а также в главах 3 и 4. Материал двух последних глав охватывает многие методы, традиционно применяемые для улучшения изображений. Таким образом, привлекая примеры из области улучшения изображений для представления новых методов обработки изображений уже в этих начальных главах, мы не только избавляемся от необходимости включения в книгу особой главы об улучшении изображений, но, что еще важнее, эффективно вовлекаем начинающих в детали методов обработки изображений на ранней стадии изучения. Однако, как будет видно по мере освоения остальной части книги, введенный в этих главах материал применим для намного более широкого класса задач, чем только лишь улучшение изображений.

Восстановление изображений — это область, также связанная с повышением визуального качества изображения, однако, в отличие от собственно улучшения, критерии которого субъективны, восстановление изображения является объективным в том смысле, что методы восстановления изображений опираются на математические или вероятностные модели искажений изображения. Напротив, улучшение изображений основано на субъективных предпочтениях человеческого восприятия, которые связаны с тем, *что* именно считается «хорошим» результатом улучшения.

Обработка цветных изображений приобрела особую важность в связи со значительным расширением использования цветных изображений в Интернете. В главе 6 излагается ряд фундаментальных понятий, относящихся к цветовым моделям и основным видам цифровых преобразований цветов. Цвет также используется в последующих главах как основа для выделения из изображения некоторых интересующих признаков.

Вейвлеты образуют фундамент для представления изображений в нескольких масштабах одновременно. В частности, этот аппарат используется в книге применительно к сжатию данных изображения, а также для построения пирамидального представления, при котором изображение поэтапно разбивается на все более мелкие фрагменты.

Сжатие, как следует из самого названия, относится к методам уменьшения объема памяти, необходимого для хранения изображения, или сужения полосы пропускания канала, требуемой для его передачи. Хотя техника запоминающих устройств за последнее десятилетие была значительно усовершенствована, этого нельзя сказать в отношении пропускной способности линий связи. Это особенно справедливо по отношению к информации в Интернете, где изобразительная составляющая является существенным элементом содержимого. Со сжатием изображений знакомы (возможно, не отдавая себе в этом отчета) большинство пользователей компьютеров, встречающих в именах графических файлов определенные расширения; например, jpg используется в стандарте сжатия изображений, разработанном Объединенной группой экспертов по фотографии (Joint Photographic Experts Group — JPEG).

Морфологическая обработка связана с инструментами для извлечения таких компонентов изображения, которые могут быть полезны для представления и описания формы. Приведенный в этой главе материал дает основы перехода от процессов, имеющих на выходе *изображение*, к процессам, имеющим на выходе *атрибуты изображения*, как это указывалось в разделе 1.1.

Сегментация разделяет изображение на составные части или объекты. В целом автоматическая сегментация принадлежит к числу самых трудных задач цифровой обработки изображений. Излишне подробная сегментация направляет процесс на длинный путь решения задачи, требуя идентификации объектов по отдельности. С другой стороны, недостаточно подробная или же неверная сегментация почти неизбежно приведет к возникновению ошибок на финальной стадии обработки. В общем, чем точнее сегментация, тем больше шансов на успех при распознавании.

Представление и описание почти всегда следуют непосредственно за этапом сегментации, на выходе которого обычно имеются лишь необработанные данные о пикселях, которые либо образуют границу области (т. е. дается множество пикселей, отделяющих одну область изображения от другой), либо представляют все точки самих областей. В обоих случаях необходимо преобразовать данные в форму, пригодную для компьютерной обработки. Первое решение, которое следует принять, — должны ли эти данные представляться в форме только границ областей или областей целиком. Представление в виде границ подходит для тех случаев, когда в центре внимания находятся внешние характеристики формы областей, например углы и изгибы. Представление в виде областей является предпочтительным, если акцент делается на внутренних свойствах объектов, например текстуре или форме скелета. В некоторых приложениях эти представления дополняют друг друга. Выбор способа представления — лишь часть принятия решения по преобразованию «сырых» пиксельных данных в подходящую для дальнейшей компьютерной обработки форму. Должен быть еще указан метод описания данных, при котором бы выдвигались на передний план интересующие признаки. *Построение описания*, иначе называемое *выбором*

признаков, связано с выделением атрибутов, которые бы выражали интересующую информацию в количественном виде или бы могли служить основой для различения классов объектов.

Распознавание представляет собой процесс, который присваивает некоторому объекту идентификатор (например «транспортное средство») на основании его описателей. Как подробно разъяснялось в разделе 1.1, мы считаем, что сфера цифровой обработки изображений заканчивается разработкой методов распознавания отдельных объектов.

До настоящего момента ничего не говорилось о необходимости априорных знаний или, в терминах рис. 1.23, о взаимосвязи между *базой знаний* и модулями обработки. На самом деле знание о проблемной области (т. е. база знаний) некоторым образом закодировано внутри самой системы обработки изображений. Это знание может быть очень простым, как, например, детальное указание участков изображения, где должна находиться интересующая информация, что позволит сузить область ее поиска. База знаний может быть и очень сложной, как, например, взаимосвязанный список всех наиболее вероятных дефектов в задаче контроля материалов либо база данных спутниковых изображений некоторого района с высоким разрешением в прикладных задачах мониторинга. Помимо того, что база знаний руководит работой каждого модуля обработки, она также управляет взаимодействием между модулями. Эта особенность показана на рис. 1.23 с помощью двунаправленных стрелок между обрабатывающими модулями и базой знаний.

Хотя мы не обсуждаем в этом месте задачу визуализации изображений, важно иметь в виду, что на выходе любой из показанных на рис. 1.23 стадий может выполняться отображение результатов обработки. Отметим также, что не во всех прикладных задачах обработки изображений требуется вся сложность взаимодействия, подразумеваемого рис. 1.23. На самом деле во многих случаях даже не все эти модули необходимы. Например, улучшение изображений для визуальной интерпретации человеком редко нуждается в использовании каких-либо других стадий из числа показанных на рис. 1.23. В общем случае, однако, чем выше сложность задачи обработки изображений, тем большее число процессов требуется привлекать для решения этой задачи.

1.5. Компоненты системы обработки изображений

Еще в середине 1980-х гг. большинство поступающих на рынок образцов систем обработки изображений были относительно крупными периферийными устройствами, которые соединялись со столь же громоздкими компьютерами. Позднее — в конце 1980-х — начале 1990-х гг. — на рынке произошел переход к аппаратуре обработки изображений в виде одиночной платы, конструктивно совместимой с какой-либо из ставших стандартом магистралей и пригодной для установки в индустриальные и персональные компьютеры. Наряду со снижением стоимости такой аппаратуры рынок также стал катализатором для возникновения значительного числа новых компаний, ориентированных на разработку программного обеспечения специально для обработки изображений.

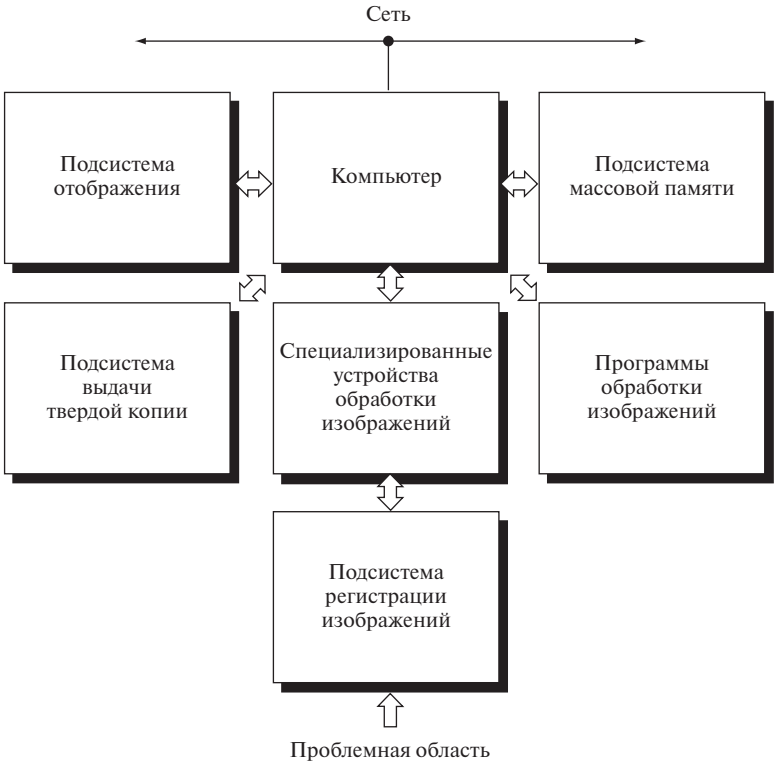


Рис. 1.24. Компоненты универсальной системы обработки изображений

Хотя для массовой обработки изображений большого размера (например спутниковых) еще производятся крупные системы цифровой обработки, общая тенденция направлена в сторону миниатюризации и оснащения обычных малых компьютеров специализированным оборудованием для решения задач обработки изображений. На рис. 1.24 изображены основные компоненты, из которых состоит типичная *универсальная* система цифровой обработки изображений, а ниже обсуждаются функции каждого из ее компонентов начиная с системы регистрации.

Что касается *регистрации*, то для получения цифровых изображений в общем случае необходимы два элемента. Первый из них — это *чувствительный элемент* (*сенсор*), т. е. физическое устройство, обладающее чувствительностью к тому виду излучаемой объектом энергии, который мы хотим отобразить. Второй элемент, *аналого-цифровой преобразователь* (АЦП), представляет собой устройство, преобразующее аналоговый выходной сигнал чувствительного элемента в цифровую форму[5]. Например, в цифровой видеокамере элементы светочувствительной матрицы вырабатывают электрический сигнал, пропорцио-

[5] Здесь необходимо отметить, что для преобразования непрерывного сигнала в цифровую форму необходимы два, вообще говоря, независимых процесса: *дискретизация*, т. е. пространственное разложение непрерывного сигнала на некоторое число отсчетов, и квантование — перевод непрерывного значения сигнала каждого из отсчетов в конечный диапазон значений. — *Прим. ред. перев.*

нальный силе света. Цифровой преобразователь трансформирует эти сигналы в цифровые данные. Более подробно эта тема рассматривается в главе 2.

Специализированные устройства для обработки изображений обычно включают вышеупомянутый цифровой преобразователь, а также оборудование, с помощью которого выполняются другие элементарные операции, как например, арифметико-логическое устройство (АЛУ), которое позволяет выполнять арифметические и логические операции параллельно для всего изображения. Один из вариантов использования АЛУ — локальное усреднение изображений одновременно с оцифровкой — может быть полезен для снижения уровня шума. Оборудование такого типа иногда называют *подсистемой предобработки* (или *препроцессором*); ее отличительной характеристикой является высокая скорость работы. Иначе говоря, этот блок выполняет функции обработки данных, требующие высокой производительности (например оцифровка и усреднение видеоизображений со скоростью 25 кадров в секунду), с чем не справляется типичный управляющий компьютер системы.

Под *компьютером* в системе обработки изображений подразумевается универсальная ЭВМ в диапазоне от обычного персонального компьютера (ПК) до суперкомпьютера. В специализированных приложениях для достижения требуемой производительности иногда используются компьютеры специальной конструкции, однако мы рассматриваем здесь именно универсальную систему обработки изображений. В таких системах практически любой хорошо оснащенный ПК пригоден для решения задач обработки изображений, не требующих работы в реальном масштабе времени.

Программное обеспечение для обработки изображений состоит из специализированных модулей, выполняющих конкретные операции. В развитых пакетах программ имеются также средства, позволяющие пользователю самостоятельно разрабатывать программы, которые как минимум запускают в работу специализированные модули системы. Более сложные программные пакеты позволяют сочетать вызов этих модулей с обычными операторами какого-либо из универсальных языков программирования.

Наличие *массовой памяти* большого объема обязательно для практических задач обработки изображений. Для хранения изображения размером 1024×1024 пикселя, в котором яркость каждого пикселя представляется 8-битовой величиной, необходим один мегабайт памяти, если не используются средства сжатия изображений. При работе с тысячами или даже миллионами изображений наличие достаточной внешней памяти в системе обработки изображений может оказаться проблематичным. Цифровые запоминающие устройства для задач обработки изображений делятся на три основные категории: (1) временная память для краткосрочного использования в ходе обработки, (2) внешняя память, обладающая относительно коротким временем обращения, и (3) архивная память, для которой характерны редкие обращения. Емкость запоминающих устройств измеряется в байтах (8 бит), килобайтах (тысяча байтов), мегабайтах (миллион байтов), гигабайтах (миллиард байтов) и терабайтах (триллион байтов)[6].

---

[6] Зачастую при выражении емкости памяти в байтах коэффициент увеличения при переходе в каждый следующий класс принимается равным $2^{10} = 1024$, а не $10^3 = 1000$, что, впрочем, здесь несущественно. — *Прим. перев.*

Одним из вариантов реализации временной памяти может быть оперативная память компьютера. Другой вариант состоит в использовании специальных плат, называемых *буферами кадров*, которые хранят одно или более изображений, обеспечивая высокую скорость чтения/записи, обычно соответствующую частоте кадров видеосигнала (например 25 кадр/с). Этот способ позволяет практически мгновенно выполнять *увеличение* изображения либо сдвигать его в вертикальном (*прокрутка*) или горизонтальном (*панорамирование*) направлениях. Буферы кадров обычно расположены в показанном на рисунке блоке «Специализированные устройства обработки изображений». Внешняя память, как правило, представлена накопителями на магнитных или оптических дисках и характеризуется частыми обращениями к хранящейся информации; т. е. важнейшим для нее является фактор быстродействия. Напротив, обращение к архивной памяти за информацией происходит редко, но требуется очень большая емкость памяти. В запоминающих устройствах, используемых в качестве архивной памяти, в качестве носителей информации обычно применяются магнитные ленты и оптические диски внутри многодисковых хранилищ с автоматической установкой и сменой дисков (jukebox).

Графические дисплеи, используемые в настоящее время, в основном оснащаются электронно-лучевыми трубками по типу телевизионных, предпочтительно с плоским экраном. Сигнал на монитор подается с платы отображения (видеоадаптера), входящей в состав компьютера. В редких случаях к системе отображения предъявляются такие требования, которым не отвечают встроенные видеоадаптеры современных компьютеров. Иногда необходим стереоскопический режим отображения; это может достигаться с помощью закрепляемой на голове гарнитуры с двумя малогабаритными дисплеями, встроенными в оправу, похожую на защитные очки, в которые и смотрит пользователь.

К числу *устройств получения твердых копий* относятся лазерные и струйные принтеры, устройства термопечати, пленочные фотокамеры и цифровые устройства, например оптические диски[7]. Максимальное разрешение достигается при выводе на пленку, однако для письменных и печатных материалов более естественным носителем является бумага. Для показа в ходе презентаций изображения выводятся на прозрачную пленку или цифровой носитель (если используется подключаемый к компьютеру проектор). Последний вариант постепенно становится общепринятым стандартом презентации изображений.

Соединение с *телекоммуникационной сетью* уже стало почти подразумеваемой функцией в любой сегодняшней компьютерной системе. Учитывая большие объемы данных, связанные с задачами обработки изображений, важнейшим фактором для передачи изображений является пропускная способность сети. В локальных сетях и на выделенных каналах телекоммуникации трудностей обычно не возникает, однако обмен информацией с удаленными пунктами через Интернет далеко не всегда оказывается столь же эффективным. К счастью, в результате развития оптоволоконных сетей и других технологий широкополосной связи положение в этой сфере быстро исправляется.

---

[7] Оптические диски, вообще говоря, относятся к устройствам хранения информации. — *Прим. перев.*

Заключение

Главной целью материала этой главы было показать в исторической перспективе истоки возникновения цифровой обработки изображений и, что более важно, нынешние и будущие возможности применения этой технологии. Хотя эти вопросы невозможно исчерпывающе рассмотреть в пределах одной главы, у читателя должно остаться ясное ощущение широты области практических применений цифровой обработки изображений. Поскольку последующие главы книги посвящены теории и приложениям обработки изображений, мы привели большое число примеров, чтобы более четко представлять перспективы и пользу от применения таких методов. Закончив изучение последней главы книги, читатель должен прийти к такому уровню понимания предмета, который является фундаментом для большинства работ, ныне проводимых в этой области.

Ссылки и литература для дальнейшего изучения

В конце последующих глав даются относящиеся к конкретным темам каждой из этих глав ссылки на общий список литературы, приведенный в конце книги. Однако в этой главе использована другая форма, чтобы собрать в одном месте всю совокупность научных журналов, публикующих материалы по обработке изображений и смежным вопросам. Приводится также список книг, по которым читатель легко может представить себе историческую и современную перспективу деятельности в этой области. Таким образом, приводимый в этой главе список литературных источников следует рассматривать как доступный путеводитель по изданной литературе в области обработки изображений.

К числу наиболее важных реферируемых журналов, публикующих статьи по обработке изображений и смежным вопросам, относятся: *IEEE Transactions on Image Processing; IEEE Transactions on Pattern Analysis and Machine Intelligence; Computer Vision, Graphics, and Image Processing* (до 1991 г.); *Computer Vision and Image Understanding; IEEE Transactions on Systems, Man and Cybernetics; Artificial Intelligence; Pattern Recognition; Pattern Recognition Letters; Journal of the Optical Society of America* (до 1984 г.); *Journal of the Optical Society of America — A: Optics, Image Science and Vision; Optical Engineering; Applied Optics — Information Processing; IEEE Transactions on Medical Imaging; Journal of Electronic Imaging; IEEE Transactions on Information Theory; IEEE Transactions on Communications; IEEE Transactions on Acoustics, Speech and Signal Processing; Proceedings of the IEEE;* а также выпуски журнала *IEEE Transactions on Computers* до 1980 г. Также представляют интерес публикации Международного общества по оптической технике (SPIE).

Следующие книги, перечисленные в обратном хронологическом порядке (и с уклоном в сторону более свежих публикаций), содержат материал, дополняющий наш взгляд на цифровую обработку изображений. Эти книги дают общую картину обсуждаемой области за последние 30 с лишним лет и были выбраны так, чтобы представить разнообразие существующих трактовок. В списке присутствуют и учебники, излагающие фундаментальный материал, и руковод-

ства по применению определенных методов, и, наконец, научные монографии, представляющие достигнутый уровень исследований в этой области.

Prince J. L., Links J. M. [2006]. *Medical Imaging, Signals, and Systems*, Prentice Hall, Upper Saddle River, NJ.

Bezdek J. C. et al. [2005]. *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*, Springer, New York.

Davies E. R. [2005]. *Machine Vision: Theory, Algorithms, Practicalities*, Morgan Kaufmann, San Francisco, CA.

Rangayyan R. M. [2005]. *Biomedical Image Analysis*, CRC Press, Boca Raton, FL.

Umbaugh S. E. [2005]. *Computer Imaging: Digital Image Analysis and Processing*, CRC Press, Boca Raton, FL.

Gonzalez R. C., Woods R. E., Eddins S. L. [2004]. *Digital Image Processing Using MATLAB*, Prentice Hall, Upper Saddle River, NJ. [Имеется перевод: Гонсалес Р., Вудс Р., Эддинс С. *Цифровая обработка изображений в среде MATLAB*. — М.: Техносфера, 2006.

Snyder W. E., Qi Hairong [2004]. *Machine Vision*, Cambridge University Press, New York.

Klette R., Rosenfeld A. [2004]. *Digital Geometry — Geometric Methods for Digital Picture Analysis*, Morgan Kaufmann, San Francisco, CA.

Won C. S., Gray R. M. [2004]. *Stochastic Image Processing*, Kluwer Academic/Plenum Publishers, New York.

Soille P. [2003]. *Morphological Image Analysis: Principles and Applications*, 2nd ed., Springer-Verlag, New York.

Dougherty E. R., Lotufo R. A. [2003]. *Hands-on Morphological Image Processing*, SPIE — The International Society for Optical Engineering, Bellingham, WA.

Gonzalez R. C., Woods R. E. [2002]. *Digital Image Processing*, 2nd ed., Prentice Hall, Upper Saddle River, NJ.

Forsyth D. F., Ponce J. [2002]. Computer Vision — A Modern Approach, Prentice Hall, Upper Saddle River, NJ.

Duda R. O., Hart P. E., Stork D. G. [2001]. *Pattern Classification*, 2nd ed., John Wiley & Sons, New York.

Pratt W. K. [2001]. *Digital Image Processing*, 3rd ed., John Wiley & Sons, NY. [Имеется перевод 1-го издания: Прэтт У. *Цифровая обработка изображений*. — М.: Мир, 1982. Кн. 1 и 2.]

Ritter G. X., Wilson J. N. [2001]. *Handbook of Computer Vision Algorithms in Image Algebra*, CRC Press, Boca Raton, FL.

Shapiro L. G., Stockman G. C. [2001]. *Computer Vision*, Prentice Hall, Upper Saddle River, NJ.

Dougherty E. R. (ed.) [2000]. *Random Processes for Image and Signal Processing*, IEEE Press, New York.

Etienne E. K., Nachtegael M. (eds.). [2000]. *Fuzzy Techniques in Image Processing*, Springer-Verlag, New York.

Goutsias J., Vincent L., Bloomberg D. S. (eds.). [2000]. *Mathematical Morphology and Its Applications to Image and Signal Processing*, Kluwer Academic Publishers, Boston, MA.

Mallot A. H. [2000]. *Computational Vision*, The MIT Press, Cambridge, MA.

Marchand-Maillet S., Sharaiha Y. M. [2000]. *Binary Digital Image Processing: A Discrete Approach*, Academic Press, New York.

Mitra S. K., Sicuranza G. L. (eds.) [2000]. *Nonlinear Image Processing*, Academic Press, New York.

Edelman S. [1999]. *Representation and Recognition in Vision*, The MIT Press, Cambridge, MA.

Lillesand T. M., Kiefer R. W. [1999]. *Remote Sensing and Image Interpretation*, John Wiley & Sons, New York.

Mather P. M. [1999]. *Computer Processing of Remotely Sensed Images: An Introduction*, John Wiley & Sons, New York.

Petrou M., Bosdogianni P. [1999]. *Image Processing: The Fundamentals*, John Wiley & Sons, UK.

Russ, J.C. [1999]. *The Image Processing Handbook*, 3rd ed., CRC Press, Boca Raton, FL.

Smirnov A. [1999]. *Processing of Multidimensional Signals*, Springer-Verlag, New York.

Sonka M., Hlavac V., Boyle R. [1999]. *Image Processing, Analysis, and Computer Vision*, PWS Publishing, New York.

Haskell B. G., Netravali A. N. [1997]. *Digital Pictures: Representation, Compression, and Standards*, Perseus Publishing, New York.

Jahne B. [1997]. *Digital Image Processing: Concepts, Algorithms, and Scientific Applications*, Springer-Verlag, New York.

Castleman K. R. [1996]. *Digital Image Processing*, 2nd ed., Prentice Hall, Upper Saddle River, NJ.

Geladi P., Grahn H. [1996]. *Multivariate Image Analysis*, John Wiley & Sons, NY.

Bracewell R. N. [1995]. *Two-Dimensional Imaging*, Prentice Hall, Upper Saddle River, NJ.

Sid-Ahmed M. A. [1995]. *Image Processing: Theory, Algorithms, and Architectures*, McGraw-Hill, NY.

Jain R., Rangachar K., Schunk B. [1995]. *Computer Vision*, McGraw-Hill, New York.

Mitiche A. [1994]. *Computational Analysis of Visual Motion*, Perseus Publishing, New York.

Baxes G. A. [1994]. *Digital Image Processing: Principles and Applications*, John Wiley & Sons, New York.

Gonzalez R. C., Woods R. E. [1992]. *Digital Image Processing*, Addison-Wesley, Reading, MA.

Haralick R. M., Shapiro L. G. [1992]. *Computer and Robot Vision*, vols. 1 & 2, Addison-Wesley, Reading, MA.

Pratt W. K. [1991] *Digital Image Processing*, 2nd ed., Wiley-Interscience, New York. [Имеется перевод 1-го издания: Прэтт У. *Цифровая обработка изображений*. — М.: Мир, 1982. Кн. 1 и 2.]

Lim J. S. [1990]. *Two-Dimensional Signal and Image Processing*, Prentice Hall, Upper Saddle River, NJ.

Jain A. K. [1989]. *Fundamentals of Digital Image Processing*, Prentice Hall, Upper Saddle River, NJ.

Schalkoff R. J. [1989]. *Digital Image Processing and Computer Vision*, John Wiley & Sons, New York.

Giardina C. R., Dougherty E. R. [1988]. *Morphological Methods in Image and Signal Processing*, Prentice Hall, Upper Saddle River, NJ.

Levine M. D. [1985]. *Vision in Man and Machine*, McGraw-Hill, New York.

Serra J. [1982]. *Image Analysis and Mathematical Morphology*, Academic Press, New York.

Ballard D. H., Brown C. M. [1982]. *Computer Vision*, Prentice Hall, Upper Saddle River, NJ.

Fu K. S. [1982]. *Syntactic Pattern Recognition and Applications*, Prentice Hall, Upper Saddle River, NJ. [Имеется перевод более ранней книги: Фу К. *Структурные методы в распознавании образов.* — М.: Мир, 1977.]

Nevatia R. [1982]. *Machine Perception*, Prentice Hall, Upper Saddle River, NJ.

Pavlidis T. [1982]. *Algorithms for Graphics and Image Processing*, Computer Science Press, Rockville, MD.

Rosenfeld R., Kak A. C. [1982]. *Digital Picture Processing*, 2nd ed., vols. 1 & 2, Academic Press, New York.

Hall E. L. [1979]. *Computer Image Processing and Recognition*, Academic Press, New York.

Gonzalez R. C., Thomason M. G. [1978]. *Syntactic Pattern Recognition: An Introduction*, Addison-Wesley, Reading, MA.

Andrews H. C., Hunt B. R. [1977]. *Digital Image Restoration*, Prentice Hall, Upper Saddle River, NJ.

Pavlidis T. [1977]. *Structural Pattern Recognition*, Springer-Verlag, New York, 1977.

Tou J. T., Gonzalez R. C. [1974]. *Pattern Recognition Principles*, Addison-Wesley, Reading, MA, 1974.

Andrews H. C. [1970]. *Computer Techniques in Image Processing*, Academic Press, New York.

Литература, добавленная при переводе

Ниже приводится список литературы по вопросам цифровой обработки изображений, который, по нашему мнению, будет способствовать расширению кругозора читателя данной книги. Из российских специализированных периодических изданий, посвященных рассматриваемой проблематике, следует отметить журнал «*Компьютерная оптика*», и журнал *Pattern Recognition and Image Analysis*; к сожалению, последний публикуется издательством Pleiades Publishing исключительно на английском языке. К числу отечественных периодических изданий, в которых достаточно регулярно публикуются научные работы по данной тематике, относятся журналы «*Автометрия*», «*Исследование Земли из Космоса*» и «*Кибернетика*». Большинство современных результатов публикуются в трудах различных научных конференций, симпозиумов, ведомственных изданий, или же в виде сборников статей, издаваемых институтами Российской Академии наук или иными научными и учебными учреждениями. Из их числа без сомнения следует отметить такие сборники как «*Иконика*» и «*Цифровая оптика*», издававшиеся Институтом проблем передачи информации РАН, а также сбор-

ники «*Вопросы кибернетики*», издаваемые научным советом по комплексной проблеме «Кибернетика» РАН.

Ранее (до начала 1990 гг.) издавался перевод ежемесячного журнала «*Труды института инженеров по электротехнике и радиоэлектронике* (*ТИИЭР*)», в котором с достаточной регулярностью появлялись как отдельные статьи, так и тематические выпуски, посвященные обработке изображений. Без сомнения, особого внимания заслуживают следующие тематические выпуски ТИИЭР.

Сокращение избыточности. — ТИИЭР, 1967, т. 55, № 3.

Цифровая обработка изображений. — ТИИЭР, 1972, т. 60, № 7.

Распознавание образов и обработка изображений. — ТИИЭР, 1979, т. 67, № 5.

Цифровое кодирование. — ТИИЭР, 1980, т. 68, № 7.

Обработка изображений. — ТИИЭР, 1981, т. 69, № 5.

Системы видеосвязи. — ТИИЭР, 1985, т. 73, № 4 и № 5.

Реконструктивная и вычислительная томография. — ТИИЭР, 1983, т. 71, № 3.

Из числа изданных на русском языке книг, представляющих интерес при изучении вопросов цифровой обработки изображений (как оригинальных, так и переводных изданий), можно рекомендовать следующие.

Ахмед Н., Рао К. Р. *Ортогональные преобразования при обработке цифровых сигналов.* — М.: Связь, 1980.

Александров В. В., Горский Н. Д. *Представление и обработки изображений: Рекурсивный подход.* — Л.: Наука, 1985.

Бейтс Р., Мак-Доннелл М. *Восстановление и реконструкция изображений.* — М.: Мир, 1989.

Бендат Дж., Пирсол А. *Прикладной анализ случайных данных.* — М.: Мир, 1989.

Бонгард М. М. *Проблема узнавания.* — М.: Наука, 1967.

Быстрые алгоритмы в цифровой обработке изображений: Преобразования и медианные фильтры / Хуанг Т. С., Эклунд Дж.-О., Нуссбаумер Г. Дж. и др. Ред. Хуанг Т. С. — М.: Радио и связь, 1984.

Василенко Г. И. *Теория восстановления сигналов: О редукции к идеальному прибору в физике и технике.* — М.: Сов.радио, 1979.

Василенко Г. И., Тараторин А. М. *Восстановление изображений.* — М: Радио и связь, 1986.

Ватолин Д. С. *Алгоритмы сжатия изображений. Методическое пособие.* М.: МГУ, 1999.

Введение в цифровую фильтрацию / Ред. Богнер Р., Константинидис А. — М.: Мир, 1976.

Виттих В. А., Сергеев В. В., Сойфер В. А. *Обработка изображений в автоматизированных системах научных исследований.* — М.: Наука, 1982.

Возенкрафт Дж., Джейкобс И. *Теоретические основы техники связи.* — М.: Мир, 1968.

Голд Б., Рэйдер Ч. *Цифровая обработка сигналов.* — М.: Сов. радио, 1973.

Гонсалес Р., Вудс Р. *Цифровая обработка изображений*, М.: Техносфера, 2005

Горелик А. Л., Скрипкин В. А. *Методы распознавания: Учебное пособие.* — М.: Высшая школа, 1989.

Гренандер У. *Лекции по теории образов.* — М.: Мир. т. 1: *Синтез образов.* — 1979. т. 2: *Анализ образов.* — 1981. т. 3: *Регулярные структуры.* — 1983.

Даджион Д., Мерсеро Р. *Цифровая обработка многомерных сигналов.* — М.: Мир, 1988.

Добеши И. *Десять лекций по вейвлетам.* — Ижевск: НИЦ «Регулярная и хаотическая динамика», 2001.

Дуда Р., Харт П. *Распознавание образов и анализ сцен.* — М.: Мир, 1976.

Жуков А. И. *Метод Фурье в вычислительной математике.* — М.: Наука, 1992.

Завалишин Н. В., Мучник И. Б. *Модели зрительного восприятия и алгоритмы анализа изображений.* — М.: Наука, 1974.

Ковалевский В. А. *Методы оптимальных решений в распознавании изображений.* — М.: Наука, 1976.

Кодирование и обработка изображений / Ред. Лебедев Д. С., Зяблов В. В. — М.: Наука, 1988.

Коды с обнаружением и исправлением ошибок. — М.: ИЛ, 1956.

Лебедев Д. С., Цуккерман И. И. *Телевидение и теория информации.* М. — Л.: Энергия, 1965.

Марр Д. *Зрение. Информационный подход к изучению представления и обработки зрительных образов.* — М.: Радио и связь, 1987.

Методы компьютерной обработки изображений / Ред. Сойфер. В.А. — М.: Физматлит, 2001.

Методы передачи изображений. Сокращение избыточности / Прэтт У. К., Сакрисон Д. Д., Мусман Х. Г. Д. и др. / Ред. Прэтт У. К. — М.: Радио и связь, 1983.

Минский М., Пейперт С. *Персептроны.* — М.: Мир, 1971.

Нильсон Н. *Искусственный интеллект. Методы поиска решений.* — М.: Мир, 1973.

Нильсон Н. *Обучающие машины.* — М.: Мир, 1967.

Обработка изображений и цифровая фильтрация. / Ред. Хуанг Т. — М.: Мир, 1979.

Обработка изображений при помощи цифровых вычислительных машин / Ред. Эндрюс Г., Инло Л. — М.: Мир, 1973. (См. также ТИИЭР, 1972, т. 60, № 7).

Оппенгейм А. В., Шафер Р. В. *Цифровая обработка сигналов.* — М.: Связь, 1979.

Осовский С. *Нейронные сети для обработки информации.* — М.: Финансы и статистика, 2002.

Павлидис Т. *Алгоритмы машинной графики и обработки изображений.* — М.: Радио и связь, 1986.

Применение цифровой обработки сигналов / Ред. Оппенгейм Э. — М.: Мир, 1980.

Прэтт У. *Цифровая обработка изображений.* — М.: Мир, 1982. Кн. 1 и 2.

Психология машинного зрения / Ред. Уинстон П. — М.: Мир, 1978.

Рабинер Л., Гоулд Б. *Теория и применение цифровой обработки сигналов.* — М.: Мир, 1978.

Распознавание образов при помощи цифровых вычислительных машин / Ред. Хармон Л. — М.: Мир, 1974. (См. также ТИИЭР, 1972, т. 60, № 10).

Реконструкция изображений / Ред. Старк Г. — М.: Мир, 1992.

Розенблатт Ф. *Принципы нейродинамики. Персептрон и теория механизмов мозга*. — М.: Мир, 1966.

Розенфельд А. *Распознавание и обработка изображений с помощью вычислительных машин*. — М.: Мир, 1972.

Столниц Э., ДеРоуз Т., Салезин Д. *Вейвлеты в компьютерной графике. Теория и приложения*. — Ижевск: НИЦ «Регулярная и хаотическая динамика», 2002.

Теория информации и ее приложения. М.: Физматгиз, 1959.

Тихонов А. Н., Арсенин В. Я. *Методы решения некорректных задач*. — М.: Наука, 1979.

Ту Дж., Гонсалес Р. *Принципы распознавания образов*. — М.: Мир, 1978.

Ульман Ш. *Принципы восприятия подвижных объектов*. — М.: Радио и связь, 1983.

Фу К. *Последовательные методы в распознавании образов и обучении машин*. — М.: Наука, 1979.

Фу К. *Структурные методы в распознавании образов*. — М.: Мир, 1977.

Фукунага К. *Введение в статистическую теорию распознавания образов*. — М.: Наука, 1979.

Хермен Г. *Восстановление изображений по проекциям. Основы реконструктивной томографии*. — М.: Мир, 1983.

Хорн Б. К. П. *Зрение роботов*. — М.: Мир, 1989.

Хэмминг Р. В. *Теория кодирования и теория информации*. — М.: Радио и связь, 1983.

Цифровая обработка изображений: Специальный выпуск. — Зарубежная радиоэлектроника. 1985, № 10.

Цифровая обработка изображений в информационных системах: Учебное пособие / Грузман И. С., Киричук В. С. и др. — Новосибирск, НГТУ, 2002.

Цифровая обработка телевизионных и компьютерных изображений / Дворкович А. В., Дворкович В. П., Зубарев Ю. Б. и др. — М.: Международный центр научной и технической информации, 1997.

Цифровое кодирование телевизионных изображений / Цуккерман И. И., Кац Б. М., Лебедев Д. С. и др. — М.: Радио и связь, 1981.

Чисар И., Кёрнер Я. Теория информации: теоремы кодирования для дискретных систем без памяти. — М.: Мир, 1985.

Чуи К. *Введение в вэйвлеты*. — М.: Наука, 2001.

Шашлов Б. А. *Цвет и цветовоспроизведение*. — М.: МГАП «Мир книги», 1995.

Эндрюс Г. *Применение вычислительных машин для обработки изображений*. — М.: Энергия, 1977.

Яне Б. *Цифровая обработка изображений*. — М.: Техносфера, 2007.

Яншин В. В. *Анализ и обработка изображений: принципы и алгоритмы*. — М.: Машиностроение, 1994.

Ярославский Л. П. *Введение в цифровую обработку изображений*. — М.: Сов. радио, 1979.

Ярославский Л. П. *Цифровая обработка сигналов в оптике и голографии: Введение в цифровую оптику*. — М.: Радио и связь, 1987.

ГЛАВА 2

ОСНОВЫ ЦИФРОВОГО ПРЕДСТАВЛЕНИЯ ИЗОБРАЖЕНИЙ

Желающий добиться успеха должен задавать правильные предварительные вопросы.

Аристотель

Введение

Цель этой главы — познакомить читателя с набором основных понятий в области цифровой обработки изображений, которые используются на протяжении всей книги. В разделе 2.1 описывается действие зрительной системы человека, включая механизмы формирования изображения в глазу, и возможности зрения в плане яркостной адаптации и контрастной чувствительности. В разделе 2.2 обсуждается свет и другие составляющие электромагнитного спектра, а также их характеристики в отношении формирования изображений. Раздел 2.3 посвящен обсуждению различных чувствительных элементов и способов регистрации цифровых изображений. В разделе 2.4 вводятся понятия равномерной пространственной дискретизации изображений и квантования по яркости. Кроме того, в этом разделе обсуждаются вопросы представления цифровых изображений и эффекты при изменении частоты дискретизации и количества уровней квантования. Вводятся понятия пространственного и яркостного разрешения и рассматриваются принципы интерполяции изображений. В разделе 2.5 рассматриваются некоторые основные взаимоотношения между пикселями. Наконец, раздел 2.6 служит введением в основной математический аппарат, используемый на протяжении всей книги. Другая цель этого раздела — помочь читателю почувствовать, как эти инструменты используются при решении разнообразных элементарных задач обработки изображений. Область рассмотрения и применения указанных инструментов по мере необходимости расширяется в оставшейся части книги.

2.1. Элементы зрительного восприятия

Хотя цифровая обработка изображений как прикладная дисциплина строится на основе математических и вероятностных формулировок, человеческая интуиция и анализ играют центральную роль при выборе того или иного метода

среди других, и этот выбор часто совершается на основе субъективного визуального оценивания. Поэтому первым шагом нашего путешествия по этой книге будет приобретение элементарных знаний о зрительном восприятии человека. Понимая сложность и широту этой темы, мы сможем охватить лишь самые начальные аспекты зрения человека. В частности, мы опишем физические механизмы и параметры, связанные с процессом формирования изображения и его восприятия человеком. Нам также интересно рассмотреть физические ограничения человеческого зрения с точки зрения факторов, используемых при работе с цифровыми изображениями. Поэтому такие вопросы, как соотношение человеческого и электронного зрения по разрешающей способности и возможности адаптироваться к изменениям освещенности, не только представляют познавательный интерес, но и важны в практическом плане.

2.1.1. Строение человеческого глаза

На рис. 2.1 в упрощенном виде показано горизонтальное сечение человеческого глаза. Глаз имеет почти сферическую форму со средним диаметром около 20 мм. Глаз окружен тремя оболочками: *роговица* со *склерой* образуют внешнюю оболочку, под которой последовательно расположены сосудистая оболочка (*хороидея*) и нейроглиальная оболочка (*сетчатка*). Роговица — это плотная прозрачная ткань, закрывающая переднюю поверхность глаза. Продолжением ее является склера — непрозрачная оболочка, закрывающая остальную часть оптической сферы глаза.

Хороидея расположена непосредственно под склерой. В этой оболочке расположена сеть кровеносных сосудов, обеспечивающих питание глаза. Даже незначительное повреждение хороидеи, часто кажущееся неопасным, может привести к серьезному нарушению зрения из-за воспаления, препятствующего нормальному кровотоку. Поверхностный слой хороидеи сильно пигментирован, что снижает интенсивность попадающего через склеру внешнего света, мешающего восприятию из-за его отражения и рассеяния внутри оптической сферы. Передняя область (зубчатая линия) хороидеи непосредственно вплетена в *цилиарное тело* и *радужную оболочку* (или *радужку*). Отверстие в центре радужной оболочки (*зрачок*) может сужаться или расширяться, регулируя тем самым количество попадающего через роговицу света. Диаметр зрачка может изменяться в пределах от 2 до 8 мм. От цвета пигмента на передней поверхности радужки зависит цвет глаза человека, а пигмент на задней поверхности радужки имеет черный цвет, что также снижает внутреннее рассеяние света.

Хрусталик (*хрусталиковая линза*), состоящий из наружной капсулы и внутрихрусталикового вещества, закреплен внутри глаза с помощью передней и задней порций волокон ресничного пояска хрусталика, которые проходят между отростками цилиарного тела и вплетаются в зубчатую линию хороидеи. Капсула и внутрихрусталиковое вещество состоят из коллагеновых волокон и содержат от 60 % до 70 % воды, около 6 % жиров и больше белков, чем любые другие ткани глаза. Внутрихрусталиковое вещество имеет слабо-желтую пигментацию, которая с возрастом усиливается. Ускоренное помутнение вещества хрусталика, связанное с нарушением его питания, приводит к заболеванию, называемому *катарактой*, при котором ухудшается цветовое восприятие и остро-

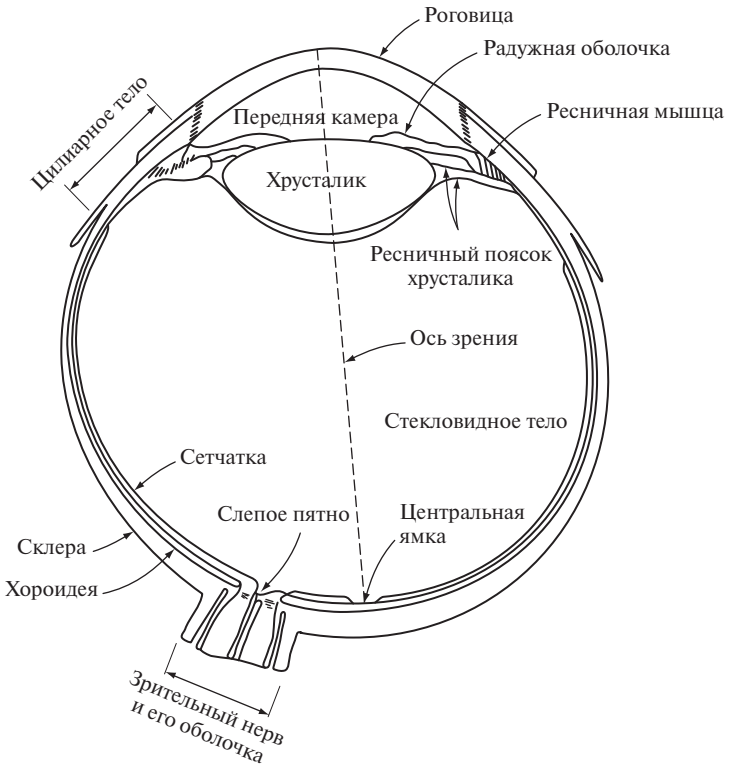


Рис. 2.1. Упрощенная схема глаза человека в разрезе

та зрения. Хрусталик поглощает около 8 % света в видимом диапазоне спектра и практически не пропускает более коротковолновое излучение. Свет инфракрасного и ультрафиолетового диапазонов существенно поглощается белком хрусталика и при высокой интенсивности может привести к необратимому нарушению зрения.

Самая внутренняя оболочка глаза — сетчатка — выстилает изнутри задний отдел глаза. При правильной оптической фокусировке глаза свет от наружного объекта проецируется в виде изображения на сетчатку. Зрительное восприятие образов становится возможным благодаря распределению дискретных светочувствительных клеток (*рецепторов*) по внутренней поверхности сетчатки. Существуют рецепторы двух видов — *колбочки* и *палочки*. В глазу насчитывается от 6 до 7 миллионов колбочек, которые обладают высокой чувствительностью к спектральным составляющим света и располагаются преимущественно в центральной области сетчатки, называемой *желтым пятном*. В центре желтого пятна имеется так называемая *центральная ямка* — область наибольшей остроты зрения. Человек различает мелкие детали изображения в основном благодаря колбочкам, поскольку каждая из них соединена с отдельным нервным окончанием. Наружные мышцы глаза обеспечивают вращение глазного яблока так, чтобы изображение интересующего объекта попадало в область желтого пятна. Колбочки обеспечивают *фотопическое* зрение, или зрение в ярком свете.

Количество палочек в глазу намного больше: по поверхности сетчатки их распределено от 75 до 150 миллионов. Бóльшая, чем у колбочек, область распределения и тот факт, что к одному нервному окончанию присоединено сразу несколько палочек (в среднем около 10), уменьшают возможности различения деталей с помощью этих рецепторов. Палочки позволяют сформировать общую картину всего поля зрения. Они наиболее чувствительны при низких уровнях освещенности и не участвуют в обеспечении функций цветного зрения. Например, предметы, имеющие яркую окраску при дневном свете, при сумеречном освещении выглядят как лишенные цветов образы, поскольку возбуждаются только палочки. Это явление известно как *скотопическое* (или сумеречное) зрение.

Рис. 2.2 иллюстрирует зависимость плотности распределения палочек и колбочек по сетчатке в зависимости от величины угла между зрительной осью и линией, проведенной из центра хрусталика до сетчатки. Изображено горизонтальное сечение правого глаза в месте выхода зрительного нерва. Отсутствие рецепторов в этой области приводит к появлению так называемого *слепого пятна* (см. рис. 2.1). В остальной области сетчатки распределение рецепторов центрально симметрично относительно центра желтого пятна. Из рис. 2.2 видно, что максимальная плотность колбочек наблюдается в центре сетчатки (в центральной ямке), а плотность палочек возрастает от этой точки приблизительно до угла в 20°, после чего плавно снижается вплоть до периферии сетчатки.

Центральная ямка представляет собой углубление круглой формы в сетчатке с диаметром около 1,5 мм. В контексте дальнейшего обсуждения более естественно говорить о прямоугольных массивах чувствительных элементов. В несколько вольной интерпретации можно рассматривать центральную ямку как квадратный массив чувствительных элементов на площади 1,5 × 1,5 мм. Плотность колбочек в этой области сетчатки приблизительно равна 150 тыс. на 1 мм^2, следовательно, общее количество колбочек в области наибольшей остроты зре-

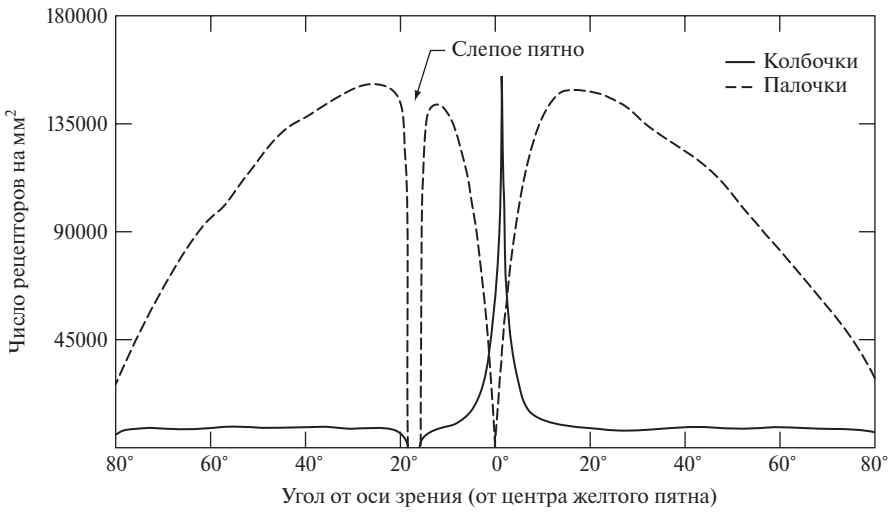


Рис. 2.2. Распределение палочек и колбочек по сетчатке

ния составляет около 337 тыс. элементов. Если рассуждать только в терминах разрешающей способности, то широко применяемые в современной технике светочувствительные матрицы среднего разрешения на основе приборов с зарядовой связью (ПЗС) содержат такое же количество чувствительных элементов при площади кристалла не более 5×5 мм. Однако такое чисто количественное сравнение было бы несколько поверхностным, поскольку не учитывает способность человека объединять зрение с интеллектом и опытом. В дальнейшем будем просто иметь в виду, что глаз человека по своей разрешающей способности вполне сопоставим с современными электронными устройствами получения изображений.

2.1.2. Формирование изображения в глазу

В обычном фотоаппарате объектив имеет постоянное фокусное расстояние, и фокусировка на различную дальность достигается изменением расстояния от объектива до плоскости изображения, в которой расположена фотопленка (или светочувствительная матрица в случае цифровой камеры). Напротив, в человеческом глазу расстояние между линзой (хрусталиком) и местом формирования изображения (сетчаткой) постоянное, а необходимое для получения четкой картины фокусное расстояние достигается за счет вариации формы хрусталика. Это осуществляется с помощью волокон ресничного пояска, придающих хрусталику уплощенную или округленную форму для фокусировки зрения, соответственно, на удаленном или близкорасположенном предмете. Расстояние от центра хрусталика до сетчатки вдоль оси зрения составляет приблизительно 17 мм, а диапазон изменения фокусного расстояния — от 14 мм до 17 мм (последнее значение соответствует расслабленному состоянию ресничной мышцы, когда глаз сфокусирован для рассматривания предметов на удалении более 3 м.

Рис. 2.3 иллюстрирует, как вычислить размеры изображения некоторого объекта на сетчатке. Пусть, например, наблюдатель смотрит на дерево высотой 15 м с расстояния 100 м. Обозначая h высоту изображения дерева на сетчатке (в мм), получаем пропорцию $15/100 = h/17$, откуда $h = 2{,}55$ мм. Как указывалось в разделе 2.1.1, проецирующееся на сетчатку изображение воспринимается главным образом областью желтого пятна. Расположенные в ней рецепторы возбуждаются в соответствии с интенсивностью падающего света, что приводит к преобразованию энергии светового излучения в электрические нервные импульсы, которые в конечном счете декодируются в мозге человека.

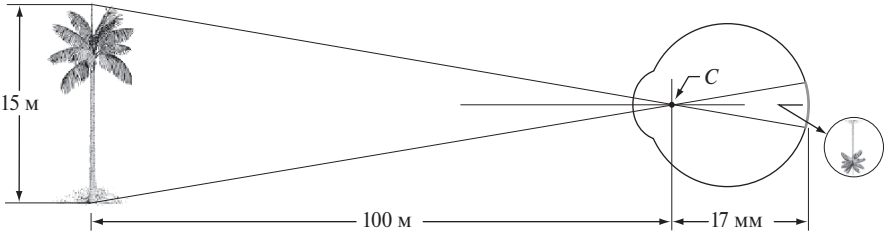


Рис. 2.3. Схематическое изображение глаза, наблюдающего дерево (точка C — оптический центр хрусталика)

2.1.3. Яркостная адаптация и контрастная чувствительность

Поскольку цифровые изображения воспроизводятся как дискретное множество элементов с различной яркостью, при представлении результатов обработки изображений необходимо учитывать способность глаза различать отличающиеся уровни яркости. Зрительная система человека способна адаптироваться к огромному, порядка 10^{10}, диапазону яркостей — от порога чувствительности скотопического зрения до предела ослепляющего блеска. Эксперименты со всей очевидностью показывают, что *субъективная яркость* (т. е. яркость как она воспринимается зрительной системой человека) является логарифмической функцией от физической яркости света, попадающего в глаз. На рис. 2.4 изображен график этой зависимости субъективной яркости от истинной яркости. Длинная сплошная кривая представляет диапазон яркостей, в котором способна адаптироваться зрительная система. При использовании одного фотопического зрения этот диапазон составляет около 10^6. Постепенный переход от скотопического зрения к фотопическому происходит в диапазоне приблизительно от 0,003 до 0,3 кд/м^2 (соответствует диапазону от -3 до -0 на графике), что показано в виде двух ветвей кривой адаптации в этом диапазоне яркостей.

Для правильной интерпретации столь впечатляющего динамического диапазона, изображенного на рис. 2.4, важно понимать, что зрительная система не способна работать во всем этом диапазоне *одновременно.* Вместо этого она охватывает такой большой диапазон за счет изменения общей чувствительности. Это явление известно как *яркостная адаптация*. Общий диапазон одновременно различаемых уровней яркости относительно мал по сравнению со всем диапазоном адаптации. Для любого данного набора внешних условий текущий уровень чувствительности зрительной системы, называемый *уровнем яркостной адаптации*, соответствует некоторой яркости, например точке B_a на рис. 2.4.

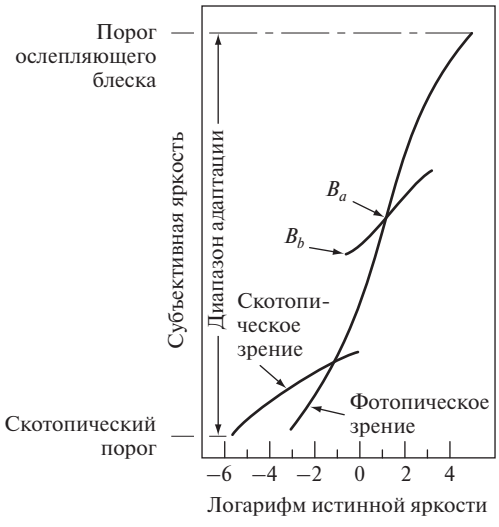


Рис. 2.4. Диапазон субъективно воспринимаемой яркости и конкретный уровень адаптации

Короткая кривая, пересекающая основной график, представляет диапазон субъективной яркости, которую способен воспринимать глаз при адаптации к указанному уровню. Этот диапазон достаточно ограничен: все уровни яркости ниже B_b субъективно воспринимаются зрением как черное и, значит, неразличимы. Верхняя часть этой кривой реально не ограничена, но теряет смысл при большой длине, поскольку при повышении яркости просто повышается уровень адаптации B_a.

Способность зрения различать *изменения* яркости при данном уровне адаптации также представляет значительный интерес. Классический эксперимент для определения способности зрительной системы человека различать разные уровни яркости состоит в том, что испытуемый смотрит на плоский равномерно освещенный экран достаточно больших размеров, такой, что он занимает все поле зрения. Как правило, это рассеиватель из матового стекла, освещаемый со стороны, противоположной наблюдателю, световым источником, яркость I которого можно регулировать. На это равномерное поле накладывается добавочная яркость ΔI в форме кратковременной вспышки в области круглой формы, расположенной в центре равномерно освещенного экрана, как изображено на рис. 2.5.

Если приращение ΔI недостаточно велико (неразличимо), испытуемый говорит «нет», указывая тем самым на отсутствие видимых изменений. По мере увеличения ΔI в какой-то момент он начнет говорить «да», подтверждая тем самым восприятие изменений яркости. Наконец, при достаточно большом значении ΔI испытуемый станет говорить «да» на каждую вспышку. Величина $\Delta I_c / I$, где ΔI_c — величина приращения яркости, различимая в 50 % случаев на фоне яркости I, называется *отношением Вебера*. Малое значение $\Delta I_c / I$ означает, что различаются очень малые относительные изменения яркости, т. е. имеет место «высокая» контрастная чувствительность. Наоборот, большое значение $\Delta I_c / I$ означает, что требуется большое относительное изменение яркости, чтобы его заметить; это говорит о «низкой» контрастной чувствительности.

График зависимости величины $\log(\Delta I_c / I)$ от $\log I$ имеет общую форму, изображенную на рис. 2.6. Эта кривая показывает, что низкая контрастная чувствительность (т. е. большое отношение Вебера) наблюдается при малых уровнях яркости, и контрастная чувствительность заметно возрастает (т. е. отношение Вебера уменьшается) при увеличении фоновой яркости. Наличие двух ветвей кривой отражает тот факт, что при малых уровнях яркости зрение осуществляется благодаря действию палочек, тогда как при больших уровнях яркости (которым соответствует высокая контрастная чувствительность) зрительные функции выполняют колбочки сетчатки.

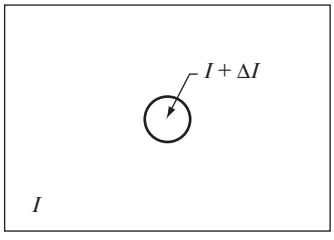


Рис. 2.5. Постановка простого эксперимента для определения характеристик контрастной чувствительности

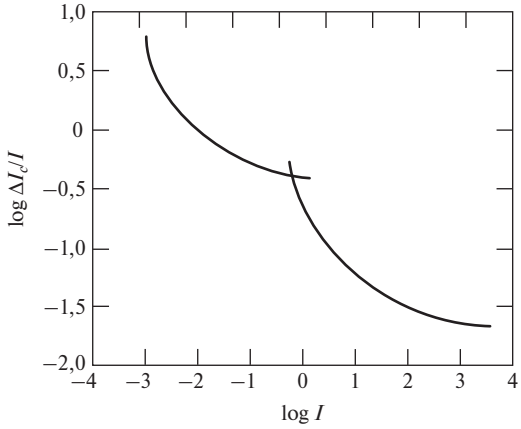


Рис. 2.6. Типичная зависимость отношения Вебера как функции яркости

Если поддерживать фоновую яркость постоянной, а яркость добавочного источника варьировать не вспышками, а ступенчатым изменением яркости от неотличимого до заметного всегда, то типичный наблюдатель способен различить всего 10—20 различающихся ступеней яркости. Грубо говоря, этот результат относится к числу различных уровней яркости, которые человек способен различить в произвольной точке монохромного изображения. Это не означает, что изображение может быть представлено таким небольшим числом градаций яркости, так как по мере движения взгляда по изображению меняется

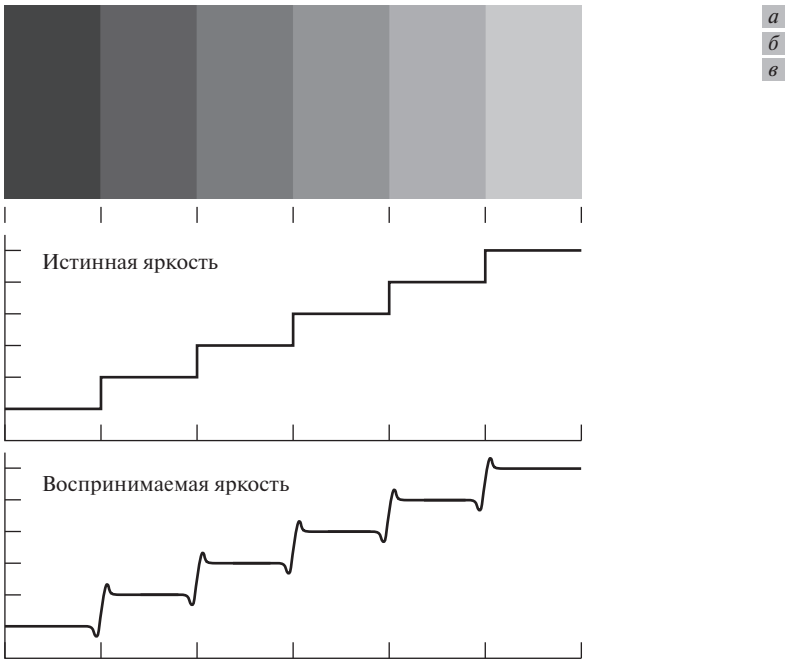


Рис. 2.7. Пример, показывающий, что воспринимаемая яркость не является просто функцией от истинной яркости

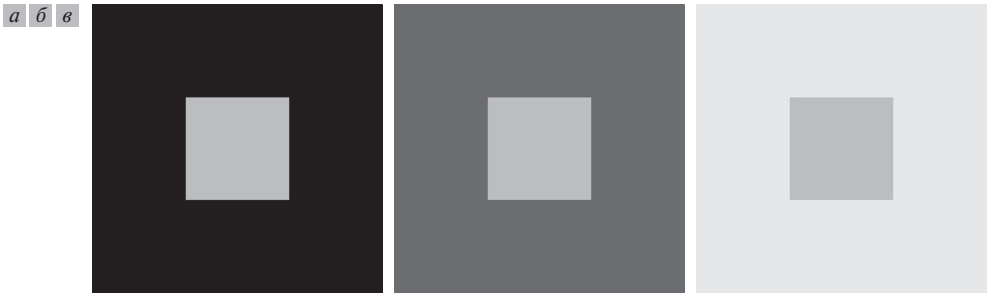


Рис. 2.8. Примеры одновременного контраста. Яркость всех центральных квадратов одинакова, но они кажутся все темнее, чем светлее становится фон

среднее значение яркости фона, что позволяет обнаруживать *различные* множества относительных изменений яркости для каждого нового уровня адаптации. Конечным следствием является способность глаза различать яркости в намного более широком *общем* диапазоне. В действительности, как мы покажем в разделе 2.4.3, глаз способен обнаруживать нежелательные ложные контуры в монохромных изображениях, общий диапазон яркостей которых представляется значительно большим количеством, чем 20 уровней.

Известны два явления, ясно доказывающие, что воспринимаемая яркость не является простой функцией истинной яркости. Первое основывается на том факте, что вблизи границ соседних областей с отличающимися, но постоянными яркостями зрение человека склонно «подчеркивать» яркостные перепады, как бы добавляя несуществующие выбросы яркости, что убедительно демонстрирует пример на рис. 2.7(*а*). Хотя яркость каждой из полос постоянна, мы, кроме действительно ступенчатого изменения яркости, видим характерные выбросы вблизи краев полос (рис. 2.7(*в*)). Эти полосы с кажущимися изменениями яркости на краях называются *полосами Маха* в честь Эрнста Маха, впервые описавшего этот феномен в 1865 г.

Второе явление, называемое *одновременным контрастом*, связано с тем фактом, что воспринимаемая яркость некоторой области не определяется просто ее яркостью, как показывает рис. 2.8. Здесь все центральные квадраты имеют в точности одинаковую яркость, однако зрительно воспринимаются тем более темными, чем светлее фон. Еще более знакомым примером является лист бумаги, который кажется белым, когда он лежит на столе, но может показаться совершенно черным, если им закрывать глаза, глядя на яркое небо.

Другими примерами феноменов человеческого зрительного восприятия являются оптические иллюзии, в которых глаз восполняет несуществующую информацию или ошибочно воспринимает геометрические свойства объектов. Некоторые примеры оптических иллюзий изображены на рис. 2.9. На рис. 2.9(*а*) ясно видны очертания квадрата, вопреки тому факту, что на изображении отсутствуют линии, определяющие такую фигуру. Аналогичный эффект, на этот раз в виде круга, виден на рис. 2.9(*б*); заметим, как всего нескольких линий достаточно для получения иллюзии полного круга. Два горизонтальных отрезка на рис. 2.9(*в*) имеют одинаковую длину, но один кажется короче другого. Наконец, все проведенные под углом 45° линии на рис. 2.9(*г*) параллельны

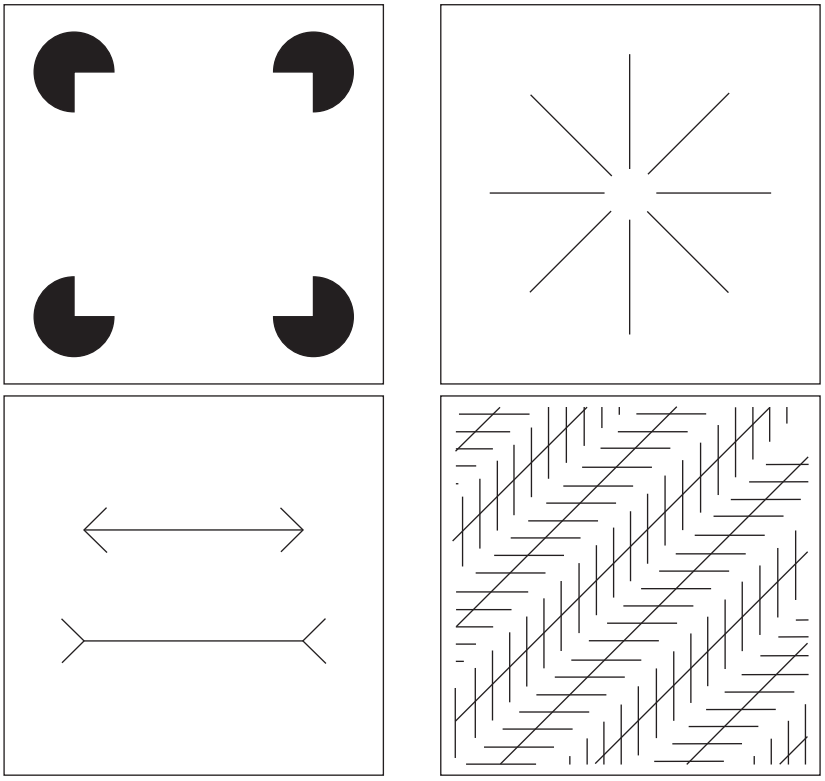


Рис. 2.9. Некоторые хорошо известные оптические иллюзии

и расположены на одинаковых расстояниях друг от друга. Однако штриховка создает иллюзию, что эти линии далеки от параллельности. Оптические иллюзии относятся к числу не вполне понятных характеристик зрительной системы человека.

2.2. Свет и электромагнитный спектр

Электромагнитный спектр был кратко представлен в разделе 1.3; теперь мы рассмотрим эту тему более подробно. В 1666 г. сэр Исаак Ньютон открыл, что при прохождении луча солнечного света сквозь стеклянную призму возникает световой пучок, который имеет не белый цвет, а состоит из непрерывного цветового спектра, цвет которого меняется от фиолетового на одном конце до красного на другом. Как видно из рис. 2.10, диапазон цветов, которые мы воспринимаем как видимый свет, составляет очень малую часть спектра электромагнитного излучения. На одном конце этого спектра находятся радиоволны, длина которых в миллиарды раз превышает длины волн видимого света, а на другом конце — гамма-лучи, длина волны которых в миллионы раз меньше длины световых волн. Компоненты электромагнитного спектра можно выражать в терминах длины волны, частоты колебаний или энергии. Длина волны (λ) и частота (ν) связаны соотношением

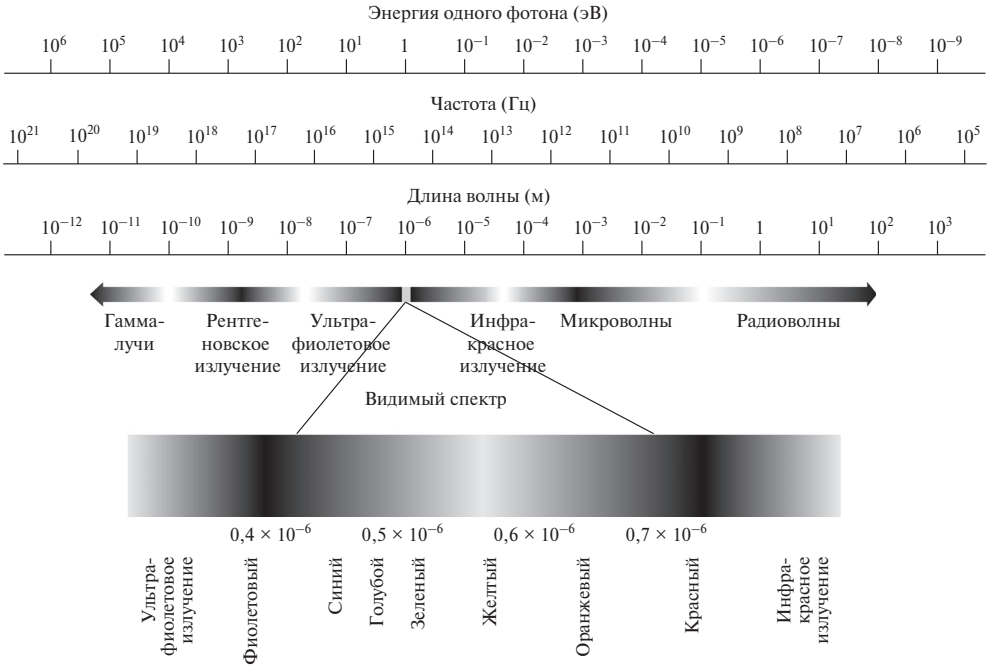


Рис. 2.10. Спектр электромагнитных колебаний. Видимый спектр показан в растянутом виде, но следует подчеркнуть, что он занимает весьма узкий участок всего электромагнитного спектра

$$\lambda = \frac{c}{\nu}, \tag{2.2-1}$$

где c — скорость света ($2{,}998 \cdot 10^8$ м/с). Энергия составляющей электромагнитного спектра определяется выражением

$$E = h\nu, \tag{2.2-2}$$

где h — постоянная Планка. Длина волны измеряется в метрах, но столь же часто в качестве единиц измерения употребляются *микрон* (1 мкм = 10^{-6} м) и *нанометр* (1 нм = 10^{-9} м)[1]. Частота измеряется в герцах (Гц); 1 Гц соответствует колебанию с частотой один период в секунду. Общеупотребительной единицей измерения энергии фотонов является электрон-вольт (эВ).

Электромагнитные волны можно трактовать как распространяющиеся синусоидальные колебания с длиной волны λ (рис. 2.11), а можно — как поток частиц с нулевой массой, движущихся со скоростью света. Каждая такая частица не имеет массы, но обладает определенной энергией и называется *квантом излучения* (*фотоном*). Из соотношения (2.2-2) видно, что энергия пропорциональна частоте, поэтому электромагнитные колебания более высокой частоты (т. е. с более короткой длиной волны) обладают большей энергией фотона. Таким образом, радиоволны характеризуются малой энергией фотона, у микроволн энергия больше, у инфракрасного излучения еще больше, далее энергия фотона

[1] Весьма распространенной единицей измерения длины световой волны является также ангстрем (1 Å = 10^{-10} м). — *Прим. перев.*

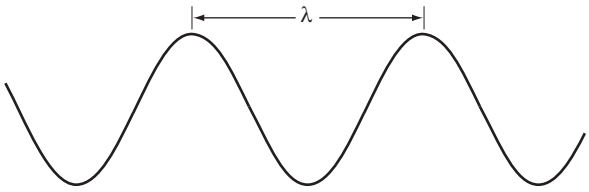


Рис. 2.11. Графическое представление длины волны колебаний

последовательно возрастает для диапазонов видимого спектра, ультрафиолетового излучения, рентгеновских лучей и, наконец, гамма-лучей, обладающих самой большой энергией. Именно по этой причине гамма-излучение так опасно для живых организмов.

Свет является особым видом электромагнитного излучения, которое воспринимается человеческим глазом. Видимый (цветовой) спектр приведен на рис. 2.10 в растянутом виде только для сведения; более подробно цвет рассматривается в главе 6. Этот видимый диапазон электромагнитного спектра охватывает длины волн приблизительно от 0,43 мкм (фиолетовый цвет) до 0,79 мкм (красный цвет). Для удобства цветовой спектр делят на семь широких полос (цветов): фиолетовый, синий, голубой, зеленый, желтый, оранжевый и красный; но это разграничение не резкое, а скорее, один цвет плавно переходит в другой, как показано на рис. 2.10, подобно любой другой составляющей электромагнитного спектра.

Различаемые зрением человека цвета предметов определяются характером света, *отраженного* от этих предметов. Тело, которое отражает свет приблизительно одинаково во всем видимом диапазоне волн, представляется наблюдателю белым, тогда как тело, отражающее свет в каком-то ограниченном диапазоне длин волн, воспринимается с некоторым цветовым оттенком. Например, зеленый предмет в основном отражает свет с длинами волн 500—570 нм, поглощая большинство энергии в других интервалах длин волн.

Свет, лишенный цветовой окраски, называется *монохроматическим* или *ахроматическим*. Единственным параметром такого освещения является его интенсивность, или *яркость*. Для описания монохроматической яркости также используется термин *уровень серого*, поскольку яркость изменяется от черного до белого, с промежуточными серыми оттенками. В дальнейшем термины «яркость» и «уровень серого» будут использоваться в одинаковом значении. Диапазон измеренных значений монохроматической яркости от черного до белого принято называть *шкалой полутонов*, и монохроматические изображения часто называют *полутоновыми изображениями*.

Хроматический (*цветной*) *свет* охватывает, как уже отмечалось, спектр электромагнитного излучения в диапазоне длин волн приблизительно от 0,43 мкм до 0,79 мкм. Помимо частоты, хроматические источники света характеризуются тремя основными величинами: энергетическим потоком, световым потоком и (субъективной) яркостью. *Энергетический поток* — это общее количество энергии, излучаемой источником света, обычно измеряемое в ваттах (Вт). *Световой поток*, измеряемый в люменах (лм), характеризует количество энергии, которое наблюдатель *воспринимает* от светового источника. Например, световой источник, работающий в дальнем инфракрасном диапазоне, может давать

значительный энергетический поток, но наблюдатель его практически не ощущает, так что световой поток такого источника почти нулевой. Наконец, как уже обсуждалось в разделе 2.1, *яркость* описывает субъективное восприятие света и практически не поддается измерению. Она олицетворяет понятие интенсивности в ахроматическом случае и является одним из ключевых факторов при описании цветового ощущения.

Продолжая обсуждение рис. 2.10, заметим, что коротковолновая сторона спектра электромагнитного излучения представлена гамма- и рентгеновскими лучами. Мы уже обсуждали в разделе 1.3.1 важность использования изображений в гамма-лучах для медицины, астрономии и ядерной энергетики. Жесткое (с более высокой энергией) рентгеновское излучение используется для получения изображений в промышленности. Для получения рентгеновских изображений грудной клетки и в стоматологии используется мягкое рентгеновское излучение (с меньшей энергией). Мягкие рентгеновские лучи плавно переходят в дальний ультрафиолетовый диапазон, длинноволновый участок которого, в свою очередь, — в видимый спектр. Двигаясь дальше в сторону увеличения длин волн, мы встретим инфракрасный диапазон, в котором излучается тепло, что делает его полезным для получения изображений на основе тепловой картины объекта. Участок инфракрасного диапазона, соседствующий с видимым спектром, называется *ближним инфракрасным* диапазоном, а противоположный участок — *дальним инфракрасным* диапазоном. Последний плавно переходит в микроволновый диапазон, хорошо известный благодаря кухонным микроволновым печам, но также используемый во многих других целях, в том числе для связи и радиолокации. Наконец, в диапазоне радиоволн осуществляется теле- и радиовещание, а в области высоких энергий этого диапазона проводятся астрономические наблюдения радиосигналов, испускаемых некоторыми звездными телами. Примеры изображений для большинства перечисленных диапазонов излучения были приведены в разделе 1.3.

В принципе, если сконструировать чувствительный элемент, способный обнаруживать излучаемую энергию в некотором диапазоне электромагнитного спектра, то можно получить изображение интересующих объектов в этом диапазоне. Однако важно заметить, что длина электромагнитных волн, используемых для «наблюдения» некоторого объекта, должна быть меньше его размера. Например, размер молекулы воды равен порядка 10^{-10} м, поэтому для исследования этих молекул необходимо применять источник излучения в диапазонах ультрафиолетового или мягкого рентгеновского излучения. Подобные ограничения, наряду с физическими свойствами материала, из которого изготовлен чувствительный элемент, определяют физические пределы возможностей сенсоров, применяемых для регистрации изображений, в частности оптических, инфракрасных или других.

Хотя подавляющее большинство получаемых цифровых изображений основано на энергии излучения электромагнитных волн, это не единственный способ генерации изображений. Например, как говорилось в разделе 1.3.7, отраженные от объектов звуковые волны могут использоваться для построения ультразвуковых изображений. Другие важные источники цифровых изображений — электронные пучки, применяемые в электронной микроскопии, и компьютерный синтез, используемый для визуализации и в компьютерной графике.

2.3. Считывание и регистрация изображения

Большинство интересующих нас изображений есть двумерное отображение наблюдаемой сцены (как правило, двух- или трехмерной), возникающее как результат регистрации лучистой энергии, исходящей из наблюдаемой сцены, с помощью некоторого устройства — сенсора (или совокупности сенсоров одновременно). Мы предполагаем, что регистрируемый сенсором сигнал возникает в результате взаимодействия источника «освещения» с элементами изображаемой «сцены» в условиях эффектов отражения и поглощения энергии этого источника. Мы берем слова *освещение* и *сцена* в кавычки, чтобы подчеркнуть тот факт, что они носят значительно более общий характер, чем в привычной ситуации, когда источник видимого света освещает обычную трехмерную бытовую сцену. Например, освещение не только может порождаться источником друго-

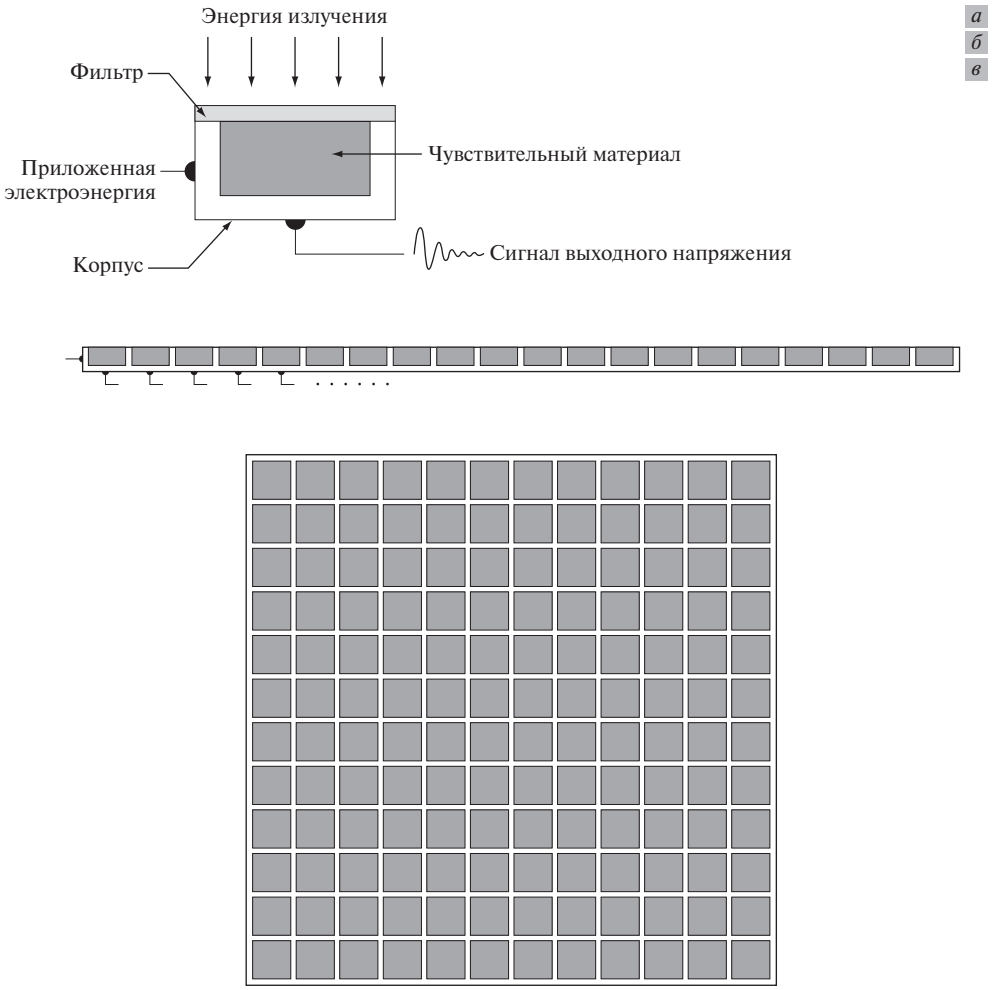


Рис. 2.12. (*а*) Одиночный чувствительный элемент. (*б*) Линейка чувствительных элементов. (*в*) Матрица чувствительных элементов

го диапазона электромагнитного излучения, например радиолокационным, инфракрасным или рентгеновским, но и происходить из менее традиционных источников, например ультразвукового или даже виртуального, синтезированного компьютерной программой. В роли элементов сцены могут выступать знакомые предметы, но вполне могут быть и молекулы, структуры подземных пластов или мозг человека. В зависимости от природы источника и особенностей сцены, энергия освещения отражается от объектов сцены или проходит сквозь них. Примером первого вида может быть свет, отраженный от поверхности предметов. Второй вид взаимодействия имеет место, например, при пропускании рентгеновских лучей через тело пациента для получения диагностического рентгеновского снимка на пленке. В некоторых прикладных задачах отраженная или проходящая энергия направляется на фотопреобразователь (например экран, покрытый флюоресцирующим материалом), который преобразует эту энергию в видимый свет. Такой подход обычен для электронной микроскопии и регистрации изображений в гамма-лучах.

На рис. 2.12 изображены три основные схемы размещения чувствительных элементов (сенсоров), которые используются для преобразования энергии «освещения» в цифровое изображение. Сама идея преобразования очень проста: падающая энергия преобразуется в напряжение благодаря сочетанию материала, обладающего чувствительностью к интересующему виду излучения, и приложенной к нему электрической энергии. В ответ на энергию внешнего излучения такой чувствительный элемент выдает сигнал выходного напряжения, который затем преобразуется в цифровую форму. В этом разделе мы рассмотрим основные способы получения и регистрации изображений, а вопросы дискретизации и квантования изображений обсуждаются в разделе 2.4.

2.3.1. Регистрация изображения с помощью одиночного сенсора

На рис. 2.12(*а*) показаны компоненты одиночного сенсора (чувствительного элемента). Вероятно, наиболее известным сенсором такого типа является фотодиод, изготовленный из полупроводникового материала (кремния), напряжение выходного сигнала которого пропорционально освещенности. Установка фильтра перед чувствительным элементом обеспечивает избирательность сенсора. Например, если установить перед сенсором зеленый пропускающий фильтр, то выходной сигнал будет выше для зеленого участка видимого спектра, чем для всех остальных.

Для получения двумерного изображения с помощью одиночного сенсора необходимо обеспечить его перемещение в двух взаимно перпендикулярных направлениях (по осям x и y) относительно регистрируемой области. На рис. 2.13 изображена конструкция, применяемая в прецизионных сканерах, где пленочный негатив закрепляется на барабане, вращение которого обеспечивает перемещение по одной оси. Одиночный сенсор закреплен на ходовом винте, вращение которого приводит к линейной подаче в перпендикулярном направлении. Поскольку механическим перемещением можно управлять с большой точностью, такой способ позволяет регистрировать изображения с высоким разрешением при небольших затратах (но медленно). Другой вид

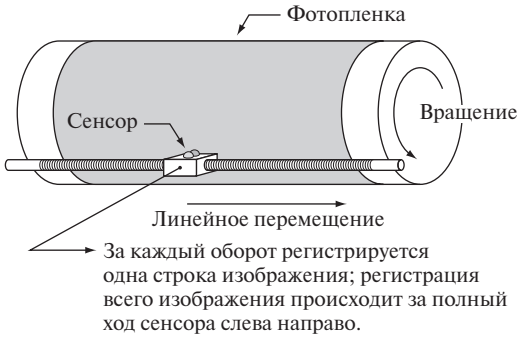


Рис. 2.13. Перемещение одиночного сенсора при регистрации двумерного изображения

механической конструкции аналогичного назначения — это планшет, по которому чувствительный элемент линейно передвигается в двух направлениях. Такие устройства с последовательным механическим сканированием всего поля изображения (как барабанные, так и планшетные) иногда называют *микроденситометрами*.

При другом способе получения изображения с помощью одиночного сенсора используется лазерный источник света, конструктивно совмещенный с сенсором. Механическая развертка осуществляется с помощью движения зеркал, направляющих луч источника на сканируемый плоский объект и возвращающих отраженный луч на чувствительный элемент. Та же конструкция может применяться для регистрации изображений с помощью линеек и матриц чувствительных элементов, что обсуждается в следующих двух параграфах.

2.3.2. Регистрация изображения с помощью линейки сенсоров

Более часто, чем одиночный сенсор, для считывания изображений используется одномерный массив сенсоров, обычно располагаемых вдоль прямой, как это показано на рис. 2.12(*б*). Такая линейка обеспечивает одновременную регистрацию элементов изображения в одном направлении (условно говоря, по строке), а перемещение всей линейки в перпендикулярном направлении позволяет получить все строки изображения (рис. 2.14(*а*)). Подобная конструкция применяется в большинстве планшетных сканеров. Удается изготавливать линейки, состоящие из 4000 и более расположенных в ряд чувствительных элементов. Расположение сенсоров в ряд широко используется при аэрофотосъемке, когда система регистрации устанавливается на самолете, летящем с постоянной скоростью и на неизменной высоте над интересующим районом. Одномерные линейки сенсоров, чувствительных к излучениям в различных участках электромагнитного спектра, располагаются перпендикулярно направлению полета. В каждый момент времени линейка сенсоров регистрирует одну строку изображения, а движение всей системы в перпендикулярном направлении позволяет заполнить второе измерение двумерного изображения. Для проекции сканируемой области на линейку сенсоров применяются линзы или другие фокусирующие устройства.

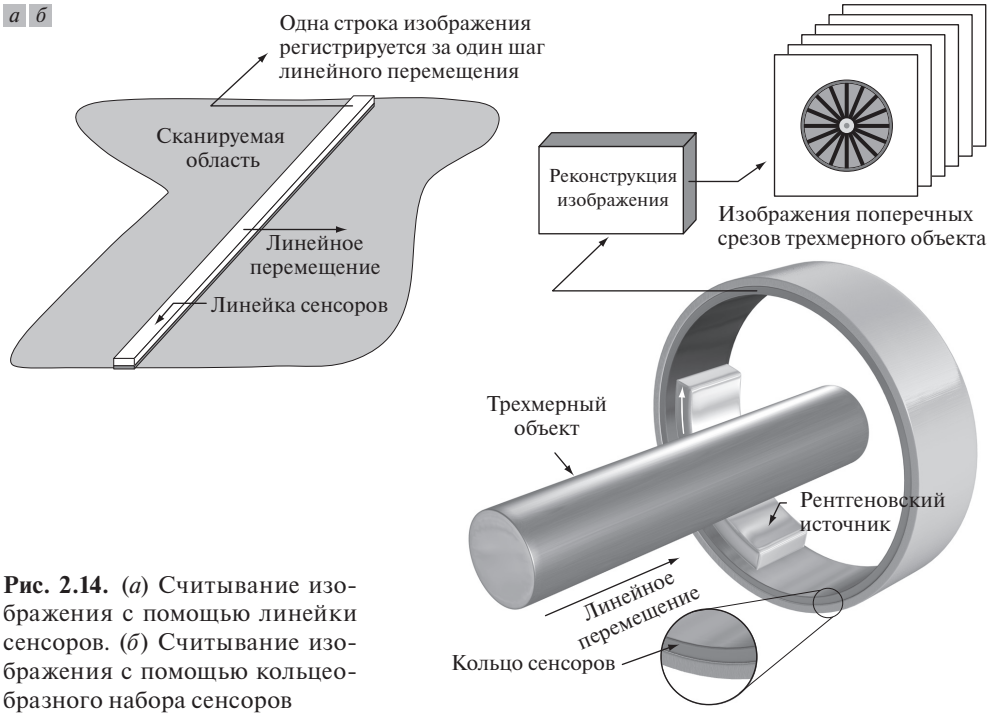


Рис. 2.14. (*а*) Считывание изображения с помощью линейки сенсоров. (*б*) Считывание изображения с помощью кольцеобразного набора сенсоров

Кольцеобразные наборы сенсоров применяются в медицине и промышленности для получения изображений поперечного сечения («срезов») трехмерных объектов, как показано на рис. 2.14(*б*). Вращающийся рентгеновский источник освещает объект, а расположенные на противоположной стороне кольца детекторы рентгеновского излучения улавливают энергию рентгеновских лучей, прошедших сквозь объект. Таков основной принцип получения изображений в компьютерной томографии (КТ), ранее упоминавшейся в разделах 1.2 и 1.3.2. Важно отметить, что выходные сигналы сенсоров подлежат обработке с помощью алгоритмов реконструкции, задача которых состоит в преобразовании регистрируемых данных в осмысленные изображения поперечных срезов (см. раздел 5.11). Другими словами, изображение среза не может быть получено простой регистрацией принимаемых сигналов одновременно с движением источника, а необходима значительная по объему вычислений обработка этих первичных данных. Трехмерное представление исследуемого объекта, состоящее из серии последовательных срезов, полученных с некоторым шагом, генерируется путем перемещения объекта в направлении, перпендикулярном к плоскости кольца. Существуют и другие способы регистрации изображений с использованием принципа КТ, но на базе иных физических процессов, в частности получение изображений методом ядерного магнитного резонанса (ЯМР) и позитронной эмиссионной томографии (ПЭТ). В них используются источники освещения и чувствительные элементы других типов, отличается и вид получаемых изображений, но принципиально эти способы регистрации изображений в значительной степени основаны на базовой схеме, показанной на рис. 2.14(*б*).

2.3.3. Регистрация изображения с помощью матрицы сенсоров

На рис. 2.12(*в*) изображено расположение отдельных сенсоров в форме двумерного массива (матрицы). Многочисленные электромагнитные и некоторые ультразвуковые устройства ввода данных сегодняшних систем обработки изображений используют именно матрицу сенсоров. Такая же конструкция находится внутри подавляющего числа цифровых камер, в которых типичным чувствительным элементом является матрица на основе приборов с зарядовой связью (ПЗС), которые выпускаются в виде монолитной конструкции, объединяющей 4000×4000 элементов (и более) с широким диапазоном чувствительных свойств. ПЗС-матрицы широко используются в цифровых фото- и видеокамерах, а также других светочувствительных приборах. Отклик каждого сенсора пропорционален интегралу световой энергии, попадающей на его поверхность за время экспозиции; это свойство используется в астрономии и других приложениях, где требуется получать изображения с низким уровнем шума. Уменьшение шума достигается за счет того, что чувствительным элементам дают возможность интегрировать принимаемый световой сигнал в течение минут или даже часов. Коль скоро изображенная на рис. 2.12(*в*) матрица сенсоров двумерна, ее главное достоинство состоит в том, что можно сформировать сразу все изображение, если сфокусировать на поверхности матрицы отвечающий ему пространственный поток лучистой энергии. Легко видеть, что в таком случае отпадает необходимость в механическом перемещении сенсоров, как это было в рассмотренных выше случаях одиночного сенсора или линейки таких сенсоров.

> В некоторых случаях регистрируется изображение самого источника, например при получении снимков Солнца.

Рис. 2.15 иллюстрирует главный способ использования матриц сенсоров. Здесь показано, что энергия, излучаемая источником освещения, отражается от объекта сцены (но, как отмечалось в начале этого раздела, энергия также может и проникать сквозь объекты сцены). Первая функция, выполняемая системой формирования изображения (рис. 2.15(*в*)), состоит в том, чтобы собрать поступающую энергию и сфокусировать ее на плоскости изображения. Если для освещения используется источник видимого света, то на входе системы формирования изображения имеется оптическая линза, которая проецирует наблюдаемую сцену на плоскость изображения (рис. 2.15(*г*)). Совмещенная с этой плоскостью чувствительная матрица генерирует набор выходных сигналов, каждый из которых пропорционален интегралу световой энергии, принятой соответствующим сенсором. С помощью цифровой и аналоговой электроники эти выходные сигналы поочередно преобразуются в комплексный видеосигнал. Тот факт, что регистрация двумерного сигнала осуществляется дискретно расположенными в пространстве сенсорами, обеспечивает пространственную дискретизацию сигнала; квантование его осуществляется в следующем блоке системы формирования изображения. На выходе ее получается цифровое изображение, схематически показанное на рис. 2.15(*д*). Преобразование изображения в цифровую форму является темой раздела 2.4.

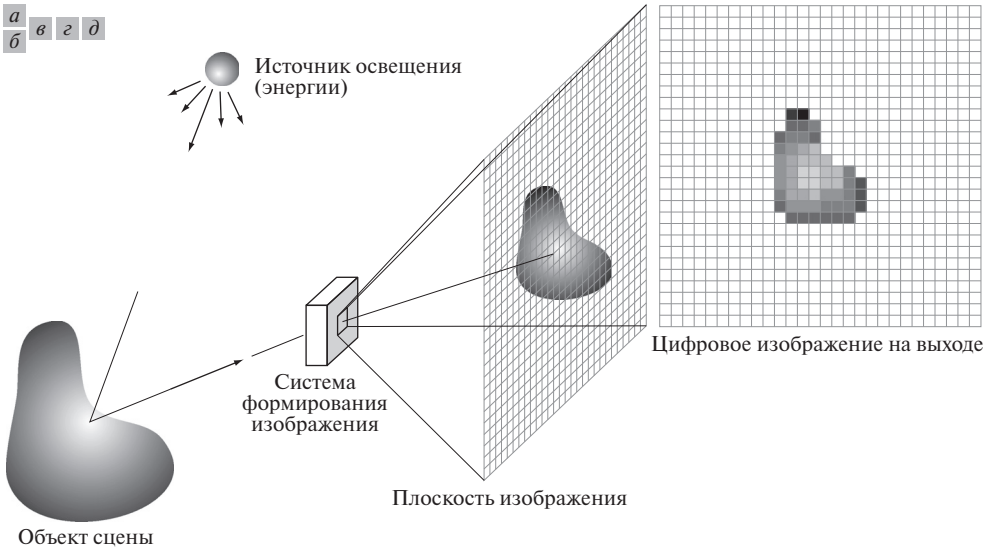


Рис. 2.15. Процесс регистрации цифрового изображения (пример). (*а*) Источник энергии («освещения»). (*б*) Элемент сцены. (*в*) Система формирования изображения. (*г*) Проекция сцены на плоскость изображения. (*д*) Оцифрованное изображение

2.3.4. Простая модель формирования изображения

Как уже говорилось в разделе 1.1, мы рассматриваем изображение как двумерную функцию вида $f(x, y)$. Значение функции f в точке с пространственными координатами (x, y) является положительной скалярной величиной, физический смысл которой определяется источником изображения. Если изображение генерируется в результате физического процесса, его значения пропорциональны энергии излучения некоторого физического источника, например энергии электромагнитных колебаний, вследствие чего функция $f(x, y)$ должна быть ненулевой и конечной, т. е.

$$0 < f(x, y) < \infty. \qquad (2.3\text{-}1)$$

Яркость элементов изображения может принимать отрицательные значения в ходе обработки или в результате интерпретации. Например, в радиолокационных изображениях объекты, движущиеся в направлении локатора, часто интерпретируются как имеющие отрицательную скорость, а объекты, удаляющиеся от локатора, — как имеющие положительную скорость. Таким образом, изображение поля скоростей может кодироваться элементами как с положительными, так и отрицательными значениями. При хранении и визуализации изображений диапазон яркостей обычно преобразуется так, чтобы минимальному отрицательному значению соответствовала нулевая яркость (см. раздел 2.6.3 относительно преобразования масштаба яркостей).

Функцию $f(x, y)$ можно характеризовать двумя компонентами: (1) величиной светового потока, который падает на наблюдаемую сцену от источника,

и (2) относительной долей светового потока, отраженного от объектов этой сцены. Мы будем называть эти компоненты *освещенностью* и *коэффициентом отражения*, обозначая их соответственно $i(x, y)$ и $r(x, y)$. Произведение этих функций дает функцию изображения:

$$f(x,y) = i(x,y)\, r(x,y), \tag{2.3-2}$$

где

$$0 < i(x,y) < \infty \tag{2.3-3}$$

и

$$0 < r(x,y) < 1. \tag{2.3-4}$$

Соотношение (2.3-4) указывает, что коэффициент отражения может меняться в пределах от 0 (полное поглощение) до 1 (полное отражение). Природа функции $i(x, y)$ зависит от источника освещения, тогда как функция $r(x, y)$ определяется свойствами объектов изображаемой сцены. Примечательно, что приведенные выражения в равной мере применимы также и к изображениям, сформированным в проходящем освещении (сквозь наблюдаемый объект), как, например, при рентгеноскопии грудной клетки. В подобном случае в качестве функции $r(x, y)$ мы имеем дело с *коэффициентом пропускания* вместо коэффициента отражения, но пределы ее изменения будут те же, что и в (2.3-4), и функция изображения формируется по той же модели — как произведение (2.3-2).

Пример 2.1. Некоторые типичные значения освещенности и коэффициента отражения.

■ Значения, указанные в соотношениях (2.3-3) и (2.3-4), представляют собой теоретические границы. Ниже приводятся средние числовые значения, иллюстрирующие типичный интервал изменения функции $i(x, y)$ для видимого света. В ясный день солнце создает на земной поверхности освещенность 90000 лм/м^2 и выше, а в пасмурную погоду эта величина падает до 10000 лм/м^2. Безоблачной ночью в полнолуние освещенность земной поверхности составляет около 0,1 лм/м^2. В типичных служебных помещениях поддерживается уровень освещенности порядка 1000 лм/м^2. Типичные значения коэффициента отражения (т. е. функции $r(x, y)$) составляют: 0,01 для черного бархата; 0,65 для нержавеющей стали; 0,80 для поверхности стены, окрашенной в ровный белый цвет; 0,90 для посеребренной металлической поверхности и 0,93 для снега. ■

Обозначим величину яркости (уровня серого) монохроматического изображения в произвольной точке (x_0, y_0)

$$\ell = f(x_0, y_0). \tag{2.3-5}$$

Из соотношений (2.3-2) — (2.3-4) видно, что ℓ лежит в некотором интервале

$$L_{\min} \le \ell \le L_{\max}. \tag{2.3-6}$$

Теоретически к границам этого интервала предъявляются только требования, чтобы $L_{\min}$ было положительно, а $L_{\max}$ — конечно. На практике $L_{\min} = i_{\min} r_{\min}$, а $L_{\max} = i_{\max} r_{\max}$. С учетом вышеприведенных типичных значений освещенности в служебных помещениях и коэффициента отражения можно ожидать типич-

ных пределов $L_{min} \approx 10$ и $L_{max} \approx 1000$ для изображений, наблюдаемых в помещениях в отсутствие дополнительного освещения.

Интервал $[L_{min}, L_{max}]$ называется *диапазоном яркостей*. На практике его обычно сдвигают по числовой оси, получая интервал $[0, L-1]$, края которого принимаются за уровень черного ($\ell = 0$) и уровень белого ($\ell = L-1$). Все промежуточные значения в этом интервале соответствуют некоторым оттенкам серого при изменении от черного до белого.

2.4. Дискретизация и квантование изображения

Из сказанного в предыдущем разделе ясно, что, несмотря на многочисленные возможные способы регистрации изображений, задача всегда одна и та же: сформировать цифровое изображение на основе данных, воспринимаемых чувствительными элементами. От большинства сенсоров поступает аналоговый выходной сигнал в форме непрерывно меняющегося напряжения, форма и амплитуда которого связаны с регистрируемым физическим явлением. Чтобы получить цифровое изображение, необходимо преобразовать непрерывно поступающий сигнал в цифровую форму. Эта операция включает в себя два процесса — *дискретизацию* и *квантование*.

> В этом разделе дискретизация обсуждается на интуитивном уровне. Более глубокое рассмотрение проводится в главе 4.

2.4.1. Основные понятия, используемые при дискретизации и квантовании

Главный принцип, лежащий в основе дискретизации и квантования изображений, проиллюстрирован на рис. 2.16. Здесь (рис. 2.16(*а*)) приведено исходное изображение $f(x, y)$, которое мы хотим преобразовать в цифровую форму. Изображение непрерывно по координатам x и y, а также по амплитуде. Чтобы преобразовать эту функцию в цифровую форму, необходимо представить ее отсчетами по обеим координатам и по амплитуде. Представление координат в виде конечного множества отсчетов называется *дискретизацией*, а представление амплитуды значениями из конечного множества — *квантованием*.

Изображенная на рис. 2.16(*б*) одномерная функция представляет собой график изменения значений яркости непрерывного изображения вдоль отрезка AB на рис. 2.16(*а*). Случайные отклонения на графике вызваны наличием шумов в изображении. Чтобы дискретизовать эту функцию, разобьем отрезок AB на равные интервалы, как показано засечками на рис. 2.16(*в*) внизу. Значения в выбранных точках отсчета представлены небольшими квадратиками на графике функции. Построенный набор значений в точках дискретизации описывает функцию в виде совокупности ее дискретных отсчетов, однако сами эти значения пока еще охватывают весь непрерывный диапазон яркостей (по вертикали). Чтобы построить цифровую функцию, диапазон яркостей также необходимо преобразовать в дискретные величины (*проквантовать*). Справа

на рис. 2.16(*в*) изображена шкала яркостей, разбитая на восемь дискретных интервалов от черного до белого. Засечки указывают конкретное значение яркости, присвоенное каждому интервалу. Квантование непрерывных значений яркости в точках дискретизации осуществляется простым сопоставлением каждому отсчету одного из восьми отмеченных значений — того, к которому ближе найденное значение яркости. В результате совместных операций дискретизации и квантования возникает отвечающий одной строке изображения дискретный набор цифровых отсчетов, показанный на рис. 2.16(*г*). Выполняя такую процедуру построчно, с верхней по нижнюю строки, получаем двумерное цифровое изображение. Рис. 2.16 наводит на мысль, что, помимо числа применяемых дискретных уровней, получаемая точность квантования сильно зависит от шумовой составляющей дискретизованного сигнала.

Выполнение дискретизации описанным выше способом предполагает, что нам доступно непрерывное по обеим координатам и по яркости изображение. На практике, однако, способ оцифровки определяется конструкцией сенсорного устройства, применяемого для регистрации изображения. Если изображение формируется одиночным сенсором в сочетании с механическим его перемещением (рис. 2.13), выходной сигнал сенсора квантуется, как описано выше, а дискретизация определяется выбором шагов механического перемещения сенсора в процессе сбора данных. Механическое перемещение может выполняться с очень

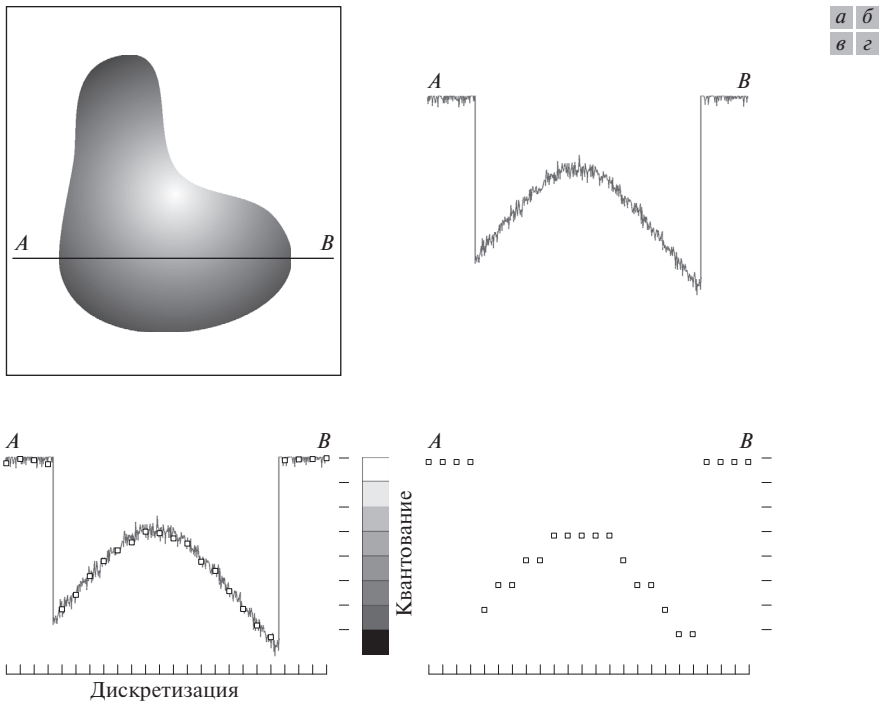


Рис. 2.16. Формирование цифрового изображения. (*а*) Непрерывное изображение. (*б*) Профиль вдоль линии сканирования между точками *A* и *B* на непрерывном изображении, который используется для иллюстрации понятий дискретизации и квантования. (*в*) Дискретизация и квантование. (*г*) Цифровое представление строки изображения

а *б*

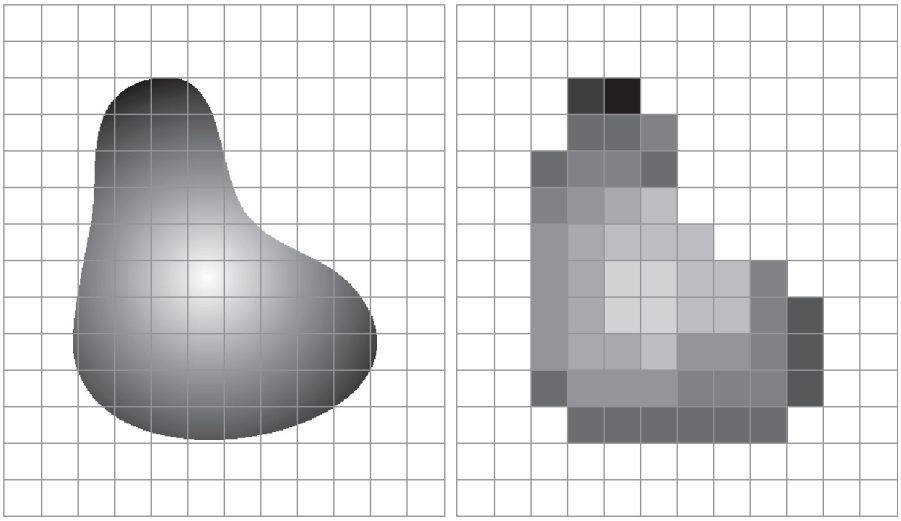


Рис. 2.17. (*а*) Проекция непрерывного изображения на матрицу чувствительных элементов. (*б*) Результат дискретизации и квантования изображения

высокой точностью, так что в принципе почти нет пределов для уменьшения шага дискретизации, однако на практике предельная точность дискретизации определяется другими факторами, например качеством оптической системы.

Если для формирования изображения используется линейка сенсоров, то число сенсоров в ней определяет предел дискретизации изображения по одному направлению. Механическим перемещением в другом направлении можно управлять и с более высокой точностью, но нет особого смысла пытаться повышать частоту дискретизации в одном направлении, коль скоро в другом направлении она жестко ограничена. Выходные сигналы всех элементов линейки подвергаются однотипному квантованию, в результате чего строится цифровое изображение.

В случае регистрации изображения с помощью матрицы сенсоров механическое перемещение отсутствует и пределы дискретизации изображения по обоим направлениям определяются числом сенсоров в матрице. Квантование их выходных сигналов осуществляется так же, как и раньше. Рис. 2.17 иллюстрирует этой случай. На рис. 2.17(*а*) показано непрерывное изображение, спроецированное на плоскость сенсорной матрицы, а рис. 2.17(*б*) демонстрирует то же изображение после дискретизации и квантования. Ясно, что качество получаемого представления в большой степени зависит от числа отсчетов в разбиении и числа уровней квантования. Однако, как мы увидим в разделе 2.4.3, при выборе значений этих количественных параметров важно учитывать содержательное наполнение изображения.

2.4.2. Представление цифрового изображения

Пусть $f(s, t)$ представляет непрерывную функцию изображения от двух непрерывных переменных s и t. Преобразуем эту функцию в *цифровое изображение* с помощью операций дискретизации и квантования, как описано в предыду-

щем разделе. Предположим, что непрерывное изображение преобразовано в двумерный массив $f(x, y)$ — матрицу из M строк и N столбцов, где (x, y) — дискретные координаты. Для ясности обозначений и большего удобства мы будем использовать для этих координат целочисленные значения: $x = 0, 1, 2,..., M - 1$ и $y = 0, 1, 2,..., N - 1$, принимая за начало координат левый верхний угол изображения, где $(x, y) = (0, 0)$. Следующим значением координат вдоль первой строки изображения будет точка $(x, y) = (0, 1)$. Важно иметь в виду, что обозначение (0, 1) используется лишь для указания на второй отсчет в первой строке и *не означает*, что это фактические значения физических координат точек дискретизации. В общем случае значение изображения в произвольной координатной точке (x, y) обозначается $f(x, y)$, где x и y — целые числа. Область действительной координатной плоскости, охватываемая координатами изображения, называется *пространственной областью*, а x и y — *пространственными переменными* или *пространственными координатами*.

Как видно из рис. 2.18, есть три основных способа представления $f(x, y)$. На рис. 2.18(*а*) представлен график функции, у которого две оси определяют поло-

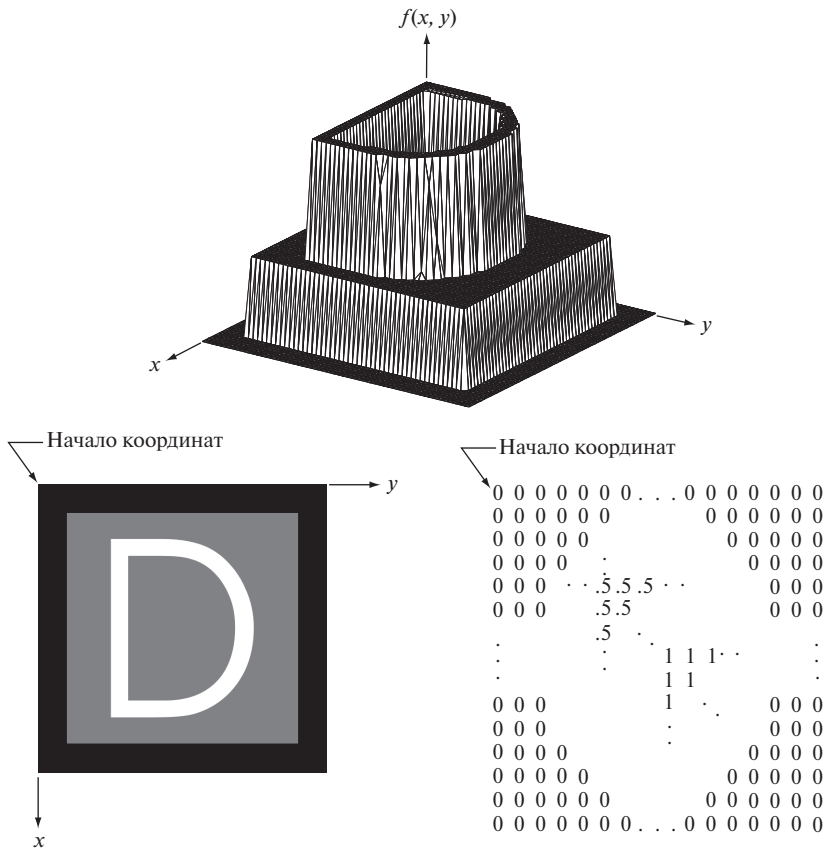


Рис. 2.18. (*а*) График изображения как объемной поверхности. (*б*) Отображение в форме массива видимых яркостей. (*в*) Изображение в виде числовой матрицы (значения 0, 0,5 и 1 представляют соответственно черное, серое и белое)

жение в пространственной области, а на третьей оси откладываются значения яркости f как функции двух пространственных координат x и y. Хотя в этом примере по графику можно представить себе структуру изображения, сложные изображения обычно содержат большое число деталей, которые трудно интерпретировать на основании подобных графиков. Такое представление полезно для работы с множествами полутонов, элементы которых представляются тройками вида (x, y, z), где x и y — пространственные координаты, а z — значение яркости f в точке с координатами (x, y). Это представление будет использоваться в разделе 2.6.4.

Представление на рис. 2.18(*б*) намного привычнее. Оно отображает $f(x, y)$ в таком виде, как изображение выглядит на мониторе или фотографии. Здесь яркость каждой точки пропорциональна значению f в этой точке. На данном рисунке есть только три значения яркости, взятые через равные промежутки. Если нормализовать яркость на интервал [0, 1], то каждая точка изображения имеет одно из значений — 0, 0,5 или 1. Монитор или принтер просто преобразуют эти три значения в черное, серое или белое соответственно, что и показано на рис. 2.18(*б*). Третье представление изображает численные значения $f(x, y)$ в виде матрицы. В этом примере f имеет размеры 600×600 элементов, т. е. 360 000 чисел. Ясно, что печать всей матрицы была бы громоздкой и малоинформативной. Однако при разработке алгоритмов такое представление вполне полезно, если в числовой форме печатаются и анализируются только фрагменты изображения, как показано на рис. 2.18(*в*).

Из сказанного можно заключить, что наиболее полезны представления, изображенные на рис. 2.18(*б*) и (*в*). Графические дисплеи позволяют мгновенно увидеть результаты. Числовые массивы применяются для обработки изображений и разработки алгоритмов. Представление числовой матрицы размерами $M \times N$ можно записать в форме соотношения

$$f(x,y)=\begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1,N-1) \end{bmatrix}. \qquad (2.4\text{-}1)$$

Обе части этого равенства — эквивалентные способы количественного представления цифрового изображения. В правой части стоит матрица действительных чисел. Каждый элемент этой матрицы называется *элементом изображения* или *пикселем*. Далее повсюду будут употребляться термины *изображение* и *пиксель* для указания на цифровое изображение и его элементы.

В ряде случаев для обозначения цифрового изображения и его элементов бывает полезно использовать более традиционную матричную запись:

$$\mathbf{A}=\begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,N-1} \\ \vdots & \vdots & & \vdots \\ a_{M-1,0} & a_{M-1,1} & \cdots & a_{M-1,N-1} \end{bmatrix}. \qquad (2.4\text{-}2)$$

Ясно, что $a_{ij} = f(x = i, y = j) = f(i, j)$, поэтому матрицы (2.4-1) и (2.4-2) идентичны. Можно даже представлять изображение как вектор $\mathbf{v}$, например, формируя вектор-столбец размерами $MN \times 1$ так, что первые M элементов $\mathbf{v}$ совпадают

с первым столбцом матрицы **A**, следующие M — со вторым столбцом и т. д. Альтернативно для формирования такого вектора мы может использовать строки матрицы **A**. Любое представление будет правильным, коль скоро обработка изображения ведется соответственно.

Возвращаясь ненадолго к рис. 2.18, отметим, что начало координат расположено в левом верхнем углу цифрового изображения, ось x направлена вниз, а ось y — вправо. Такое общепринятое представление основано на том факте, что многие графические дисплеи (например телевизионные мониторы) осуществляют развертку изображения, начиная с левого верхнего угла и двигаясь затем вправо вдоль каждой строки. Еще важнее то, что первым элементом матрицы считается элемент в левом верхнем углу, так что выбор в качестве начала координат этой точки является математически осмысленным. Следует иметь в виду, что такое представление является привычной правосторонней[2] декартовой системой координат, в которой просто оси направлены вниз и вправо, а не вправо и вверх.

Иногда может быть полезно выражать операции дискретизации и квантования в более формальных математических терминах. Пусть Z и R обозначают соответственно множества целых и действительных чисел. Процесс дискретизации можно рассматривать как разбиение плоскости xy на множество ячеек прямоугольной сетки, координаты центра каждой ячейки которой суть элементы декартова произведения Z^2, т. е. множества всех пар (z_i, z_j), где z_i и z_j — элементы множества Z. Следовательно, $f(x, y)$ — цифровое изображение, если (x, y) суть целочисленные пары из Z^2, и функция f приписывает каждой паре координат (x, y) конкретное значение яркости, т. е. действительное число из множества R. Такое сопоставление, осуществляемое функцией f, отвечает описанному выше процессу дискретизации. Если (как обычно в этой и последующих главах) значения яркости также целочисленные и вместо множества R используется Z, тогда цифровое изображение становится двумерной цифровой функцией, у которой как обе координаты, так и значения — целые числа.

> Для вычислений или разработки алгоритмов часто бывает полезно отобразить шкалу L значений яркости на интервал [0, 1], так что они перестают быть целочисленными. Однако при хранении и визуализации изображений в большинстве случаев полученные значения масштабируются обратно на интервал целых значений $[0, L - 1]$.

Для выполнения процесса оцифровки изображения необходимо принять решения относительно значений M и N, а также числа дискретных уровней (градаций) яркости L. Для M и N не устанавливается специальных ограничений помимо того, что они должны быть положительными целочисленными значениями. Однако значение L по соображениям удобства построения оборудования для хранения и квантования обычно выбирают равным целочисленной степени двойки:

$$L = 2^k. \tag{2.4-3}$$

---

[2] Напомним, что правосторонней называется такая система координат (x, y, z), для которой при вытянутом указательном пальце правой руки вдоль оси x и согнутом под прямым углом среднем пальце вдоль оси y большой палец направлен вдоль оси z. Как видно из рис. 2.18(*a*), система координат изображения именно такая.

Мы предполагаем, что дискретные уровни яркости расположены с постоянным шагом (т.е. используется равномерное квантование) и принимают целые значения в интервале [0, $L - 1$]. Иногда интервал значений яркости неформально называют динамическим диапазоном изображения. Этот термин используется по-разному в различных областях. Здесь мы определяем *динамический диапазон* системы регистрации изображений как отношение максимально возможного значения измеренной яркости к минимальному уровню яркости, обнаруживаемому этой системой. Как правило, верхний предел определяется *насыщением*, а нижний предел — *шумом* (см. рис. 2.19). По существу, динамический диапазон устанавливает минимальный и максимальный уровни яркости, которые система способна представлять и, следовательно, могут присутствовать в изображении. С этим понятием тесно связан *контраст* изображения, который мы определяем как разность между максимальным и минимальным уровнями яркости в данном изображении. Если заметная доля пикселей обладает большим динамическим диапазоном, можно ожидать, что такое изображение имеет высокий контраст. Наоборот, изображение с малым динамическим диапазоном обычно выглядит тусклым, размытым и серым. Более подробно эта тема обсуждается в главе 3.

Общее количество бит b, необходимое для хранения цифрового изображения, определяется по формуле

$$b = M \times N \times k . \qquad (2.4\text{-}4)$$

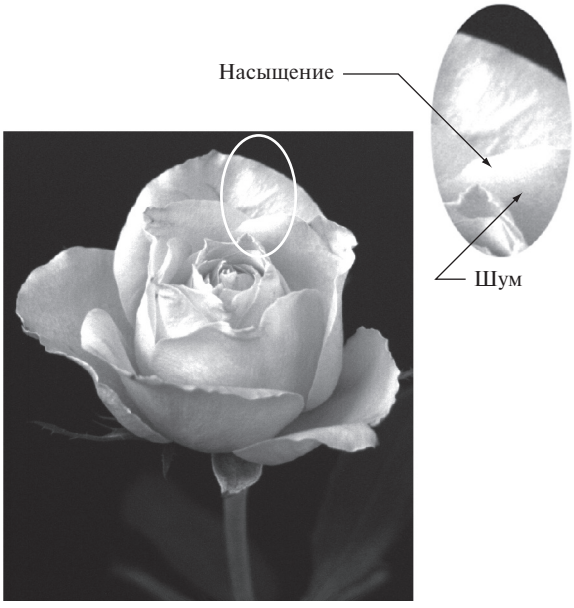


Рис. 2.19. Изображение, в котором присутствуют насыщение и шум. Насыщение определяется тем максимальным уровнем, выше которого все уровни яркости неразличимы (обрезаются). Обратите внимание, что вся область насыщения имеет высокий *постоянный* уровень яркости. Шум в данном случае проявляется в виде зернистой текстуры. Шум, особенно в темных областях изображения (например на стебле розы), маскирует минимально различимый уровень истинной яркости

Таблица 2.1. Число бит для хранения изображения при различных значениях N и k. L — число градаций яркости

N/k	1 ($L=2$)	2 ($L=4$)	3 ($L=8$)	4 ($L=16$)
32	1.024	2.048	3.072	4.096
64	4.096	8.192	12.288	16.384
128	16.384	32.768	49.152	65.536
256	65.536	131.072	196.608	262.144
512	262.144	524.288	786.432	1.048.576
1024	1.048.576	2.097.152	3.145.728	4.194.304
2048	4.194.304	8.388.608	12.582.912	16.777.216
4096	16.777.216	33.554.432	50.331.648	67.108.864
8192	67.108.864	134.217.728	201.326.592	268.435.456

N/k	5 ($L=32$)	6 ($L=64$)	7 ($L=128$)	8 ($L=256$)
32	5.120	6.144	7.168	8.192
64	20.480	24.576	28.672	32.768
128	81.920	98.304	114.688	131.072
256	327.680	393.216	458.752	524.288
512	1.310.720	1.572.864	1.835.008	2.097.152
1024	5.242.880	6.291.456	7.340.032	8.388.608
2048	20.971.520	25.165.824	29.369.128	33.554.432
4096	83.886.080	100.663.296	117.440.512	134.217.728
8192	335.544.320	402.653.184	469.762.048	536.870.912

В случае квадратного изображения $M = N$ и это равенство приобретает вид

$$b = N^2 k\,. \tag{2.4-5}$$

В табл. 2.1 приводится число бит, необходимых для хранения квадратных изображений при различных значениях N и k. Количество градаций яркости, соответствующее каждому значению k, указано в скобках. Если пиксели изображения могут принимать 2^k значений яркости, то такое изображение часто называют «k-битовым»; например, изображение с возможными 256 градациями яркости называют восьмибитовым. Из таблицы видно, что для хранения 8-битовых изображений размерами 1024×1024 и более элементов требуется существенный объем памяти.

2.4.3. Пространственное и яркостное разрешения

Интуитивно понятно, что пространственное разрешение — это размер мельчайших различимых деталей на изображении. Количественно *пространственное разрешение* можно определять многими способами; к наиболее употребительным относятся *число пар линий на единицу длины* и *число точек (пикселей) на единицу длины*. Предположим, что построен чертеж, состоящий из чередующихся

черных и белых вертикальных линий, каждая шириной W единиц длины (W может быть меньше 1). Таким образом, ширина *пары линий* составляет $2W$, и на единице длины размещается $1/2W$ таких пар. Например, если ширина линии равна 0,1 мм, то на единицу длины (миллиметр) приходится пять пар линий. Широко используемое определение разрешения состоит именно в указании максимального числа *различимых* пар линий на единицу длины; например 100 пар линий на миллиметр. Число точек на единицу длины — мера разрешения изображения, которая повсеместно применяется в полиграфии и издательском деле. В США эту характеристику обычно выражают в *точках на дюйм* (dpi). Представление о качестве дают следующие цифры: газеты печатаются с разрешением 75 dpi, журналы — 133 dpi, глянцевые брошюры — 175 dpi, а эта страница книги (имеется в виду англоязычный оригинал — *прим. ред. перев.*) напечатана с разрешением 2400 dpi.

В предыдущем абзаце главное то, что содержательная мера пространственного разрешения должна определяться по отношению к пространственным единицам измерения. Размеры изображения сами по себе не дают полной информации. Сказать, например, что изображение имеет разрешение 1024×1024 пикселя — не показательно, если при этом не указаны пространственные размеры, занимаемые этим изображением. И размеры изображения сами по себе полезны только для сравнения возможностей получения изображений. Например, можно ожидать, что цифровой фотоаппарат с 20-мегапиксельной светочувствительной ПЗС-матрицей обеспечит более высокую детальность изображения, чем 8-мегапиксельный фотоаппарат, в предположении, что оба аппарата оснащены сравнимыми объективами и сравниваются снимки, сделанные с одинакового расстояния.

Яркостным (или *полутоновым*) *разрешением* аналогично называется мельчайшее различимое изменение яркости. Если при дискретизации изображений имеется относительно большая свобода действий при выборе числа отсчетов (т. е. частоты дискретизации), то при выборе числа градаций яркости приходится в значительной степени учитывать особенности аппаратуры; по этим причинам число градаций обычно выбирается равным степени 2. Наиболее частым решением является выбор 8-битового представления (256 градаций яркости), но в некоторых приложениях используется 16 бит, если необходимо иметь более точное представление полутонов. Квантование уровней яркости с точностью 32 бит используется редко. Иногда можно встретить системы, в которых квантование уровней яркости изображения проводится с точностью 10 или 12 бит, но это скорее исключение, чем правило[3]. В отличие от пространственного разрешения, которое имеет смысл только если дается в отношении к единице длины, *яркостным разрешением* принято называть число бит, используемых при квантовании величины яркости. Например, об изображении, у которого яркость квантуется на 256 градаций, обычно говорят как об имеющем яркостное разрешение 8 бит. Поскольку фактически различимые изменения яркости зависят

[3] Отметим, что *точность квантования* яркости, определяемую используемым АЦП, во многих случаях путают с *точностью представления*, которая в значительной степени зависит от способа хранения. Весьма часто сигнал квантуется на 2^6, 2^{10}, 2^{12} или 2^{14} градаций, но сохраняется при этом с точностью 8 или 16 бит соответственно. — *Прим. ред. перев.*

не только от уровней шума и насыщения, но и от возможностей человеческого восприятия (см. раздел 2.1), то заявление, что изображение имеет яркостное разрешение 8 бит, означает лишь то, что формирующая его система способна квантовать величину яркости через равные ступени, каждая величиной в 1/256 максимальной амплитуды яркости.

Следующие два примера иллюстрируют влияние по отдельности пространственного и яркостного разрешений на степень детализации изображения. Позже в этом разделе мы обсудим, как эти два параметра совместно определяют ощущаемое качество цифрового изображения.

Пример 2.2. Демонстрация эффектов при уменьшении пространственного разрешения цифрового изображения.

■ Рис. 2.20 демонстрирует эффекты, возникающие в результате снижения пространственного разрешения цифрового изображения. Изображения рис. 2.20(*а*)—(*г*) приведены с разрешением 1250, 300, 150 и 72 dpi соответственно. Естественно, по числу элементов изображения с более низким разрешением меньше оригинала. Например, изображение при 72 dpi представляет собой матрицу размерами 213×162, тогда как размеры оригинала — 3692×2812 пикселей. Для большей наглядности сравнения все изображения с меньшими размерами были увеличены до размеров оригинала (метод увеличения рассматривается в разделе 2.4.4). В некотором смысле это эквивалентно рассмотрению уменьшенных изображений с более близкого расстояния, чтобы объективно судить о сравнительной степени их детальности.

Между рис. 2.20(*а*) и (*б*) имеются небольшие видимые отличия, наиболее заметные в виде легких искажений формы большой черной стрелки, но по большей части изображение на рис. 2.20(*б*) вполне приемлемо. Фактически 300 dpi — это типичное минимальное пространственное разрешение в книгоиздании, так что здесь не следовало ожидать больших видимых отличий. На рис. 2.20(*в*) начинает проявляться ухудшение визуального качества изображения (например вид круглых краев хронометра и малой стрелки в правой части, которая указывает на 60). Рис. 2.20(*г*) демонстрирует заметное ухудшение большинства деталей изображения. Как говорится в разделе 4.5.4, при печати с таким низким разрешением в полиграфии применяется ряд хитростей (таких как локальное изменение размеров печатной точки), которые позволяют получить значительно лучшие результаты, чем показаны на рис. 2.20(*г*). Кроме того, как мы увидим в разделе 2.4.4, приведенные на рис. 2.20 результаты можно улучшить за счет выбора метода интерполяции. ■

Пример 2.3. Типичные эффекты при изменении числа градаций яркости в цифровом изображении.

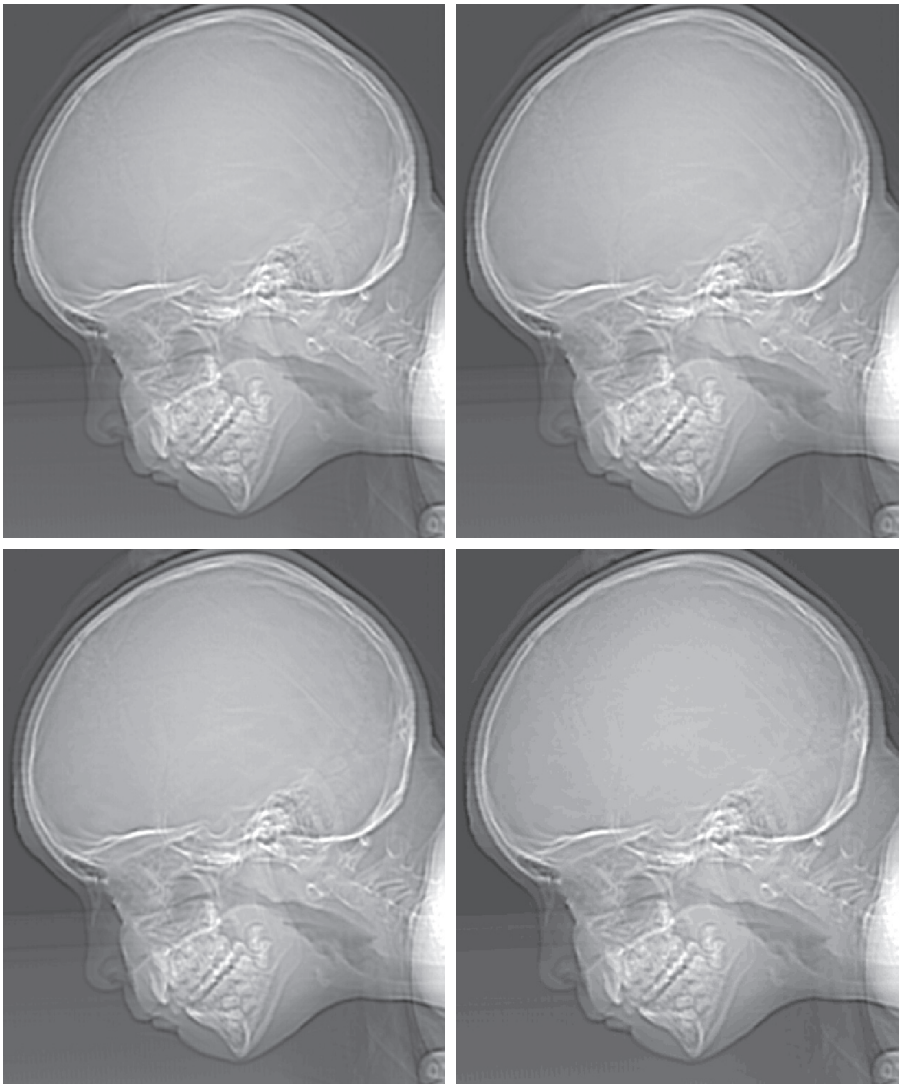
■ В этом примере мы сохраняем число отсчетов дискретизации постоянным, но постепенно уменьшаем число уровней квантования с 256 до 2, двигаясь по степеням 2. На рис. 2.21(*а*) показан полученный методом проекционной КТ рентгеновский снимок размерами 452×374 пикселей с 256 градациями яркости ($k = 8$). Подобные изображения получаются при закреплении рентгеновского источника в некотором положении, что дает двумерное изображение в нужном

а *б*
в *г*

Рис. 2.20. Типичные эффекты при уменьшении пространственного разрешения. Приведены изображения с разрешением (*а*) 1250 dpi, (*б*) 300 dpi, (*в*) 150 dpi и (*г*) 72 dpi. Тонкие рамки добавлены для удобства, они не являются частью изображений

направлении. Проекционные КТ-изображения используются для настройки параметров компьютерного томографа, например угла наклона, количества срезов и диапазона.

Изображения на рис. 2.21(*б*—*з*) были получены путем уменьшения числа бит представления с $k = 7$ до $k = 1$ при сохранении постоянного пространственного разрешения 452×374 пикселей. Изображения с 256, 128 и 64 градациями яркости



а *б*
в *г*

Рис. 2.21. (*а*) Изображение размерами 452×374 пикселей с 256 градациями яркости. (*б*)—(*г*) Изображения того же пространственного разрешения, представленные с 128, 64 и 32 градациями яркости

оказываются визуально неотличимыми и в равной мере применимыми. Однако на 32-градационном изображении (рис. 2.21(*г*)) появляются почти незаметные мелкие рубчатые структуры в области постоянной или мало меняющейся яркости, в частности внутри черепной коробки. Этот эффект, вызванный использованием недостаточного числа градаций яркости в областях плавных переходов полутонов, называется появлением *ложных контуров*, поскольку эти линии напоминают контурные линии горизонталей на топографической карте. Ложные контуры становятся отчетливо видны на изображениях, представленных с равномерным разбиением всего яркостного диапазона на 16 градаций и менее, как видно из рис. 2.21(*д*—*з*).

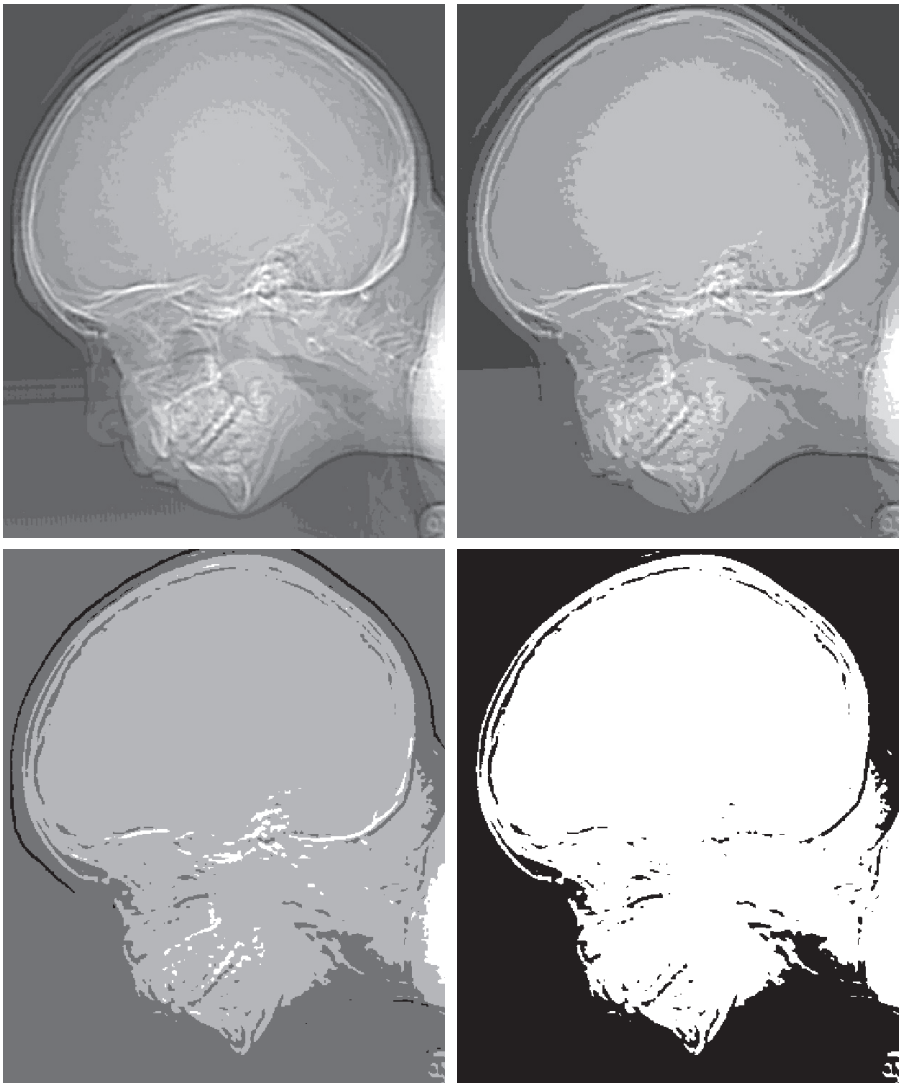


Рис. 2.21. (Продолжение.) (*д*)—(*з*) Изображения, представленные с 16, 8, 4 и 2 градациями яркости. (Исходное изображение предоставил д-р Дэвид Р. Пикенс, департамент радиологии медицинского центра университета Вандербилта)

В качестве очень грубого эмпирического правила можно считать, что минимальные пространственное и яркостное разрешение, при котором цифровое изображение, напечатанное с размерами порядка 5×5 см, будет относительно свободным от дефектов типа ложных контуров и ступенчатости, составляет около 256×256 пикселей с 64 градациями яркости. ■

Результаты, показанные на примерах 2.2 и 2.3, иллюстрируют влияние независимого изменения значений N и k на качество изображения. Но это дает лишь частичный ответ на вопрос о влиянии N и k на свойства цифровых изображений,

а б в

Рис. 2.22. (*а*) Изображение с малым числом деталей. (*б*) Изображение со средним уровнем детализации. (*в*) Изображение с относительно большим количеством деталей. (Изображение (*б*) предоставлено Массачусетским технологическим институтом)

поскольку мы пока не рассматривали возможные взаимосвязи между этими двумя параметрами. В ранних исследованиях [Huang, 1965] делалась попытка измерить экспериментально, как влияет на качество изображения одновременное изменение значений N и k. Эксперимент состоял из набора субъективных тестов, в которых использовались изображения, подобные приведенным на рис. 2.22. Женское лицо относится к классу изображений с относительно малым содержанием деталей, снимок толпы, напротив, содержит большое число деталей, а изображение кинооператора занимает промежуточное положение по степени наличия деталей.

Генерировался набор изображений указанных трех классов для различных значений N и k, после чего наблюдателю предлагалось упорядочить их в соответствии с субъективно ощущаемым качеством. Результаты эксперимента представлялись в форме так называемых *кривых равного предпочтения* на плоскости Nk (рис. 2.23 демонстрирует типичные образцы кривых равного предпочтения для изображений, приведенных на рис. 2.22). Каждая точка на плоскости Nk соответствует изображению со значениями параметров N и k, равными координатам этой точки. Точки, лежащие на кривой равного предпочтения, соответствуют изображениям одинакового субъективно воспринимаемого качества. В описываемых экспериментах было обнаружено, что кривые равного предпочтения имеют тенденцию к смещению вправо и вверх, однако форма их для каждого из трех рассматриваемых классов изображений отличается и в целом подобна изображенной на рис. 2.23. В целом результаты не являются неожиданными, поскольку сдвиг кривой вправо и вверх означает просто увеличение значений N и k, что неминуемо ведет к повышению качества изображения.

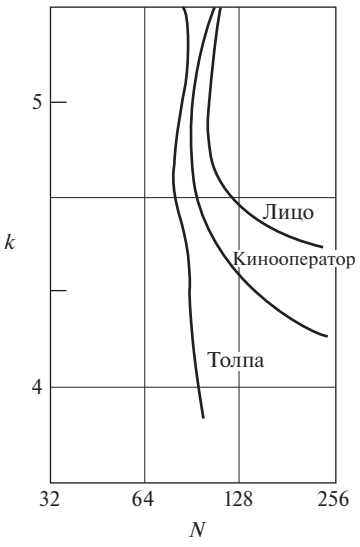


Рис. 2.23. Образцы кривых равного предпочтения для трех типов изображений, показанных на рис. 2.22

В контексте нашего обсуждения наиболее важным представляется тот факт, что кривые равного предпочтения приобретают все более вертикальную форму по мере увеличения степени детализации изображения. Такой результат наводит на мысль, что для изображений с большим количеством деталей может оказаться достаточным лишь небольшое число градаций яркости. Например, приведенная на рис. 2.23 кривая равного предпочтения для изображения толпы почти вертикальна. Это показывает, что для фиксированного значения N воспринимаемое качество изображения такого типа почти не зависит от числа использованных градаций яркости (в показанном на рис. 2.23 диапазоне). Интересно также отметить, что для других двух типов изображений визуальное качество остается одинаковым в некоторых интервалах, где пространственное разрешение увеличивается при одновременном уменьшении числа градаций яркости. Наиболее вероятная причина такого результата состоит в том, что при уменьшении k повышается видимый контраст изображения, а этот эффект часто субъективно воспринимается человеком как улучшение качества изображения.

2.4.4. Интерполяция цифрового изображения

Интерполяция является основным инструментом, широко используемым при решении таких задач, как увеличение и уменьшение изображений, их поворот и коррекция геометрических искажений. Главная цель этого раздела — ввести в рассмотрение интерполяцию и ее использование для изменения (уменьшения или увеличения) размеров цифрового изображения, что по существу означает *повторную дискретизацию* (*передискретизацию*) исходного изображения. Применение интерполяции в таких приложениях, как поворот и коррекция геометрических искажений, рассматривается в разделе 2.6.5. Мы также вернемся к этой теме в главе 4, где повторная дискретизация цифрового изображения обсуждается более подробно.

В своей основе *интерполяция* — это процесс, при котором имеющиеся данные используются для оценки значений в неизвестных точках. Начнем обсуждение этой темы с простого примера. Предположим, что имеется изображение размерами 500×500 пикселей, которое необходимо увеличить в 1,5 раза до размеров 750×750 пикселей. Простейший способ визуально наблюдать увеличение — это вообразить сетку размерами 750×750 элементов с таким же шагом, что и у пикселей исходного изображения, и затем сжимать ее вплоть до точного совпадения краев с краями исходного изображения. Очевидно, что шаг сжатой сетки 750×750 будет меньше одного пикселя исходного изображения. Чтобы присвоить значение яркости любому элементу наложенного изображения, найдем ближайший к нему пиксель исходного изображения и припишем его яркость данному элементу сетки 750×750. Присвоив так значения яркости всем элементам наложенной сетки, после растяжения ее до первоначальных размеров получим увеличенное цифровое изображение.

Такой способ называется *интерполяцией по ближайшему соседу*, поскольку в нем каждому пикселю нового изображения присваивается значение яркости ближайшего соседнего пикселя в исходном изображении (вопросы соседства пикселей рассматриваются в разделе 2.5). Этот подход отличается простотой,

но, как мы увидим позже в данном разделе, он часто приводит к появлению на изображении нежелательных артефактов вроде значительного искажения прямолинейных контуров и поэтому редко применяется на практике. Более подходящим методом является *билинейная интерполяция*, в которой для присвоения яркостей элементам нового изображения используются четыре ближайшие соседа данной точки. Пусть (x, y) — координаты точки нового изображения, которой нужно присвоить значение яркости (эту точку можно представить себе как точку ранее упоминавшейся наложенной сетки). Обозначим этот уровень яркости $v(x, y)$. В случае билинейной интерполяции он задается соотношением

$$v(x, y) = ax + by + cxy + d, \tag{2.4-6}$$

где коэффициенты *a, b, c, d* находятся из системы четырех линейных уравнений с четырьмя неизвестными, выписанной для четырех ближайших соседей точки (x, y). Как мы скоро увидим, билинейная интерполяция дает значительно лучшие результаты, чем интерполяция по ближайшему соседу, ценой умеренного усложнения вычислений.

> Заметим, что, вопреки названию, билинейная интерполяция *не является* линейной ввиду наличия члена *xy*, однако она линейна отдельно по каждой переменной при фиксированном значении другой переменной.

Следующей по уровню сложности является *бикубическая интерполяция*, в которой участвуют шестнадцать ближайших соседей рассматриваемой точки. Значение яркости, которое будет приписано точке (x, y), определяется соотношением

$$v(x, y) = \sum_{i=0}^{3} \sum_{j=0}^{3} a_{ij} x^i y^j, \tag{2.4-7}$$

где шестнадцать коэффициентов a_{ij} находятся из системы 16 уравнений с 16 неизвестными, выписанной для шестнадцати ближайших соседей точки (x, y). Заметим, что соотношение (2.4-7) сводится к виду (2.4-6), если в обеих суммах стоят пределы суммирования от 0 до 1. В общем случае при бикубической интерполяции мелкие детали изображения сохраняются лучше, чем при билинейной, почему первая стандартно используется в коммерческих программах обработки изображений, таких как Adobe Photoshop и Corel Photopaint.

Пример 2.4. Сравнение методов интерполяции для целей увеличения и уменьшения изображений.

■ На рис. 2.24(*а*) снова повторено изображение из рис. 2.20(*г*), которое было получено уменьшением разрешения 1250 dpi исходного изображения на рис. 2.20(*а*) до 72 dpi (в результате чего размеры изображения уменьшились с 3692×2812 до 213×162 пикселей) и последующим увеличением при печати до первоначальных размеров. Чтобы получить изображение рис. 2.20(*г*), в обоих случаях, при уменьшении и увеличении, использовалась интерполяция по ближайшему соседу. Как уже говорилось, показанный на рис. 2.24(*а*) результат довольно плохого качества. На рис. 2.24(*б*) и (*в*) приведены результаты аналогичной процедуры, но с использованием соответственно билинейной и бикубической

а б в
г д е

Рис. 2.24. (*а*) Изображение, уменьшенное до разрешения 72 dpi и затем увеличенное до исходных размеров 3692×2812 пикселей с помощью интерполяции по ближайшему соседу. Это изображение повторяет рис. 2.20(*г*). (*б*) Изображение, уменьшенное и увеличенное с помощью билинейной интерполяции. (*в*) То же, что и (*б*), но с использованием бикубической интерполяции. (*г*)—(*е*) Аналогичная последовательность для случая, когда разрешение уменьшается до 150 dpi вместо 72 dpi (рис. 2.24(*г*) совпадает с рис. 2.20(*в*)). Сравните рис. 2.24(*д*) и (*е*), особенно последний, с исходным изображением на рис. 2.20(*а*)

интерполяции при уменьшении и увеличении. Результат, полученный с помощью билинейной интерполяции, значительно лучше по сравнению с интерполяцией по ближайшему соседу. Резкость изображения, полученного методом бикубической интерполяции, несколько выше, чем при билинейной интерполяции. На рис. 2.24(*г*) снова повторено изображение из рис. 2.20(*в*), полученное из исходного путем снижения разрешения до 150 dpi методом интерполяции по ближайшему соседу (при уменьшении и при увеличении). Обсуждая выше этот рисунок, мы говорили, что при уменьшении разрешения до 150 dpi начинает ухудшаться визуальное качество изображения. Рис. 2.24(*д*) и (*е*) демонстрируют результаты использования соответственно билинейной и бикубической ин-

терполяции. Несмотря на снижение исходного разрешения 1250 dpi до 150 dpi, эти два последние изображения вполне сопоставимы с оригиналом, что говорит о мощи двух обсуждаемых методов интерполяции. Как и прежде, результаты бикубической интерполяции отличаются большей резкостью. ■

Для интерполяции может использоваться большее число соседей, и существуют более сложные методы, например с использованием сплайнов и вейвлетов, что в некоторых случаях может дать лучшие результаты, чем описанные методы интерполяции. Хотя сохранение мелких деталей оказывается исключительно важным соображением при генерации изображений в трехмерной компьютерной графике [Watt, 1993; Shirley, 2002] и при обработке медицинских изображений [Lehmann et al., 1999], однако ввиду повышенной вычислительной сложности применение такого подхода в универсальных системах обработки изображений редко бывает оправданным, и обычно для операций увеличения и уменьшения ограничиваются билинейной или бикубической интерполяцией.

2.5. Некоторые фундаментальные отношения между пикселями

В этом разделе мы рассмотрим некоторые важные взаимосвязи между элементами цифрового изображения. Как указывалось выше, мы будем обозначать изображение в виде функции $f(x, y)$. Ссылаясь в пределах этого раздела на конкретные пиксели, мы будем пользоваться строчными буквами, например p и q.

2.5.1. Соседи отдельного элемента

У элемента изображения p с координатами (x, y) имеются четыре соседа по *вертикали* и *горизонтали*, координаты которых даются выражениями

$$(x+1, y), (x-1, y), (x, y+1), (x, y-1).$$

Это множество пикселей называется *четверкой соседей p* и обозначается $N_4(p)$. Каждый его элемент находится на единичном расстоянии от (x, y); если же точка (x, y) лежит на краю изображения, то некоторые из соседей оказываются за пределами изображения. Этот вопрос обсуждается в главе 3.

Четыре соседа p по *диагонали* имеют координаты

$$(x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1)$$

и обозначаются $N_D(p)$. Вместе с четверкой соседей эти точки образуют так называемую *восьмерку соседей*, обозначаемую $N_8(p)$. Как и выше, некоторые точки множеств $N_D(p)$ и $N_8(p)$ могут оказаться за пределами изображения, если точка (x, y) лежит на его краю.

2.5.2. Смежность, связность, области и границы

Пусть V — множество значений яркости, используемое при определении понятия смежности. В бинарном изображении $V = \{1\}$, если смежными счита-

ются соседние пиксели с единичным значением яркости. Для полутоновых изображений идея та же, но множество V обычно состоит из большего числа элементов. Например, при определении понятия смежности для пикселей с диапазоном возможных значений яркости от 0 до 255 множество V может быть любым подмножеством этих 256 значений. Мы будем рассматривать три вида смежности:

1) 4-*смежность*. Два пикселя p и q со значениями из множества V являются 4-смежными, если q входит в множество $N_4(p)$;

2) 8-*смежность*. Два пикселя p и q со значениями из множества V являются 8-смежными, если q входит в множество $N_8(p)$;

3) *m-смежность* (*mixed*, смешанная). Два пикселя p и q со значениями из множества V являются m-смежными, если:

а) элемент q входит в множество $N_4(p)$ или

б) элемент q входит в множество $N_D(p)$ и множество $N_4(p) \cap N_4(q)$ не содержит элементов изображения со значением яркости из множества V.

> Мы используем символы $\cap$ и $\cup$ для обозначения соответственно пересечения и объединения множеств. Напомним, что пересечением множеств A и B называется множество, элементы которого входят одновременно в A и B, а объединение этих двух множеств состоит из элементов, принадлежащих или множеству A, или B, или одновременно обоим множествам. Множества и операции над ними подробнее рассматриваются в разделе 2.6.4.

Смешанная смежность представляет собой модификацию 8-смежности с целью исключения неоднозначности, часто возникающей при использовании 8-смежности в чистом виде. Рассмотрим, например, изображенную на рис. 2.25(*а*) конфигурацию пикселей при $V = \{1\}$. Три элемента в верхней части рис. 2.25(*б*) демонстрируют неоднозначную 8-смежность, как указано пунктирными линиями. Эта неоднозначность устраняется при использовании m-смежности, что иллюстрирует рис. 2.25(*в*).

Дискретным путем (или *кривой*) от пикселя p с координатами (x,y) до пикселя q с координатами (s, t) называется неповторяющаяся последовательность пикселей с координатами

$$(x_0,y_0),(x_1,y_1),...,(x_n,y_n),$$

где $(x_0, y_0) = (x,y)$, $(x_n, y_n) = (s, t)$ и пиксели (x_i, y_i) и (x_{i-1}, y_{i-1}) являются смежными при $1 \le i \le n$. В этом случае n называется *длиной* пути. Если $(x_0, y_0) = (x_n, y_n)$, то путь называется *замкнутым*. Можно определить 4-, 8- или *m-путь* в соответствии с заданным типом смежности. Например, на рис. 2.25(*б*) изображено два 8-пути между правым верхним и правым нижним элементами, а на рис. 2.25(*в*) показан m-путь.

Пусть S — некоторое подмножество элементов изображения. Два его элемента p и q называются *связными в S*, если между ними существует путь, целиком состоящий из элементов подмножества S. Для любого пикселя p из S множество всех пикселей, связных с ним в S, называется *связной компонентой* (или *компонентой связности*) S. Если множество S содержит только одну компоненту связности, оно называется *связным множеством*.

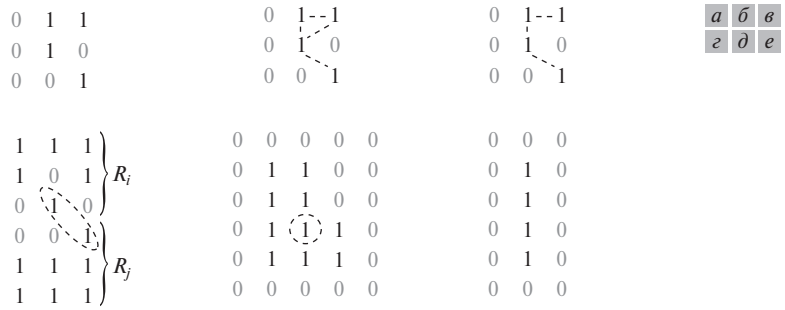


Рис. 2.25. (*а*) Конфигурация пикселей. (*б*) Элементы, являющиеся 8-смежными между собой (показано пунктиром; видна неоднозначность). (*в*) Отношения *m*-смежности. (*г*) Две области единичных пикселей, являющиеся смежными, если используется 8-смежность. (*д*) Элемент в кружке является частью границы области единичных пикселей, если только между областью и фоном рассматривается 8-смежность. (*е*) Внутренняя граница области с единичными значениями не образует замкнутого пути, а ее внешняя граница — образует

Пусть R — подмножество элементов изображения. Будем называть его *областью*, если R — связное множество. Две области R_i и R_j называются *смежными*, если их объединение является связным множеством. Области, не являющиеся смежными, называются *раздельными (несмежными)*. При рассмотрении областей подразумевают либо 4-, либо 8-смежность. Чтобы такие определения были содержательными, необходимо указать используемый тип смежности. Например, две области единичных пикселей на рис. 2.25(*г*) являются смежными только в смысле 8-смежности (в соответствии с данным выше определением, не существует 4-пути между этими областями, и потому их объединение не есть связное множество).

Предположим, что изображение содержит K несмежных областей R_k, $k = 1, 2, ..., K$, ни одна из которых не примыкает к краю изображения[4]. Обозначим R_u объединение этих K областей, а $(R_u)^c$ — его дополнение (напомним, что *дополнением* множества S называется множество точек, не принадлежащих S). Назовем совокупность точек R_u *интересующей областью* изображения, а совокупность точек $(R_u)^c$ — его *фоновой областью (фоном)*.

Границей области R (также называемой *замкнутым контуром* или *краем*) назовем множество точек этой области, которые являются смежными к точкам дополнения R. Иначе говоря, граница области — это множество пикселей этой области, у которых один или более соседей принадлежат фону. Опять-таки здесь необходимо еще указать вид связности, используемый для определения смежности. Например, точка, обведенная кружком на рис. 2.25(*д*), не является частью границы области единичных пикселей, если при рассмотрении смежности интересующей области и фона используется 4-связность. Во избежание

[4] Такое предположение делается, чтобы избежать рассмотрения особых случаев. Потери общности при этом не происходит, так как если одна или более областей примыкают к краям изображения, мы всегда можем дополнить изображение рамкой шириной в один пиксель с фоновым значением яркости.

подобных ситуаций, как правило, смежность точек области и фона определяют в терминах 8-связности.

Предыдущее определение иногда называют определением *внутренней границы* области, в отличие от *внешней границы*, определяемой аналогичным образом для фона. Такое различение важно при разработке алгоритмов, в которых прослеживаются границы. Обычно такие алгоритмы строятся для обхода внешней границы, чтобы получить в результате замкнутый путь. Например, внутренняя граница области пикселей с единичным значением на рис. 2.25(*e*) совпадает с самой областью. Такая граница не удовлетворяет данному выше определению замкнутого пути. С другой стороны, внешняя граница этой области образует замкнутый путь вокруг нее.

Если подмножество R есть все изображение (которое, напомним, является прямоугольной матрицей элементов), то его граница по определению состоит из элементов первого столбца и первой строки, а также элементов последнего столбца и последней строки. Такое доопределение необходимо, поскольку у этих элементов нет соседей за пределами изображения. Обычно, говоря об области, имеют в виду подмножество всего изображения, а элементы на границе области, которые совпадают с краем изображения, безусловно включаются в состав границы этой области.

При обсуждении областей и границ часто возникает понятие *контура*. Между контуром (в общем случае незамкнутым) и границей существует принципиальная разница. Граница конечной области всегда образует замкнутый путь и поэтому является «глобальным» понятием. Контуры же, как подробно обсуждается в главе 10, состоят из пикселей, на которых значения производной яркости превышают заранее заданный порог. Поэтому по самой своей идее контур является «локальным» понятием, основанным на мере непрерывности уровня яркости в некоторой точке. Контурные точки могут соединяться, образуя сегменты контуров, и эти сегменты иногда соединяются подобно границам, но такой случай имеет место не всегда. Единственным исключением, когда контуры и границы соответствуют друг другу, являются бинарные изображения. В зависимости от используемых вида связности и оператора выделения контуров (эти операторы рассматриваются в главе 10), выделение контуров в бинарной области дает результаты, совпадающие с границей этой области. Ограничимся здесь этим изложением на интуитивном уровне и, пока мы не достигли главы 10, будем понимать контуры как разрывы яркости, а границы — как замкнутые пути.

2.5.3. Меры расстояния

Пусть элементы изображения p, q и z имеют координаты (x, y), (s, t) и (v, w) соответственно. Функция D называется *функцией расстояния* или *метрикой*, если:

а) $D(p, q) \geq 0$, причем $D(p, q) = 0$ тогда и только тогда, когда $p = q$,

б) $D(p, q) = D(q, p)$,

в) $D(p, z) \leq D(p, q) + D(q, z)$.

Евклидово расстояние (метрика L_2) между элементами p и q определяется следующим образом:

$$D_e(p,q) = \left[(x-s)^2 + (y-t)^2\right]^{\frac{1}{2}}. \qquad (2.5\text{-}1)$$

При такой метрике пиксели, находящиеся на расстоянии не более r от заданной точки (x, y), образуют круг радиуса r с центром в этой точке.

Расстояние D_4 (метрика L_1) между элементами p и q определяется следующим образом:

$$D_4(p,q) = |x-s| + |y-t|. \tag{2.5-2}$$

В этом случае пиксели, находящиеся на расстоянии D_4, меньшем или равном r, от заданной точки (x, y), образуют повернутый на 45° квадрат с центром в этой точке. Например, пиксели с расстоянием $D_4 \le 2$ от центральной точки образуют следующие замкнутые линии равных расстояний:

$$\begin{array}{ccccc} & & 2 & & \\ & 2 & 1 & 2 & \\ 2 & 1 & 0 & 1 & 2 \\ & 2 & 1 & 2 & \\ & & 2 & & \end{array}$$

Пиксели с расстоянием $D_4 = 1$ являются четверкой соседей для элемента (x, y).

Расстояние D_8 (метрика L_∞) между элементами p и q определяется следующим образом:

$$D_8(p,q) = \max\left(|x-s|, |y-t|\right). \tag{2.5-3}$$

В этом случае пиксели, находящиеся на расстоянии D_8, меньшем или равном r, от заданной точки (x, y), образуют квадрат с центром в этой точке. Например, пиксели с расстоянием $D_8 \le 2$ от центральной точки образуют следующие замкнутые линии равных расстояний:

$$\begin{array}{ccccc} 2 & 2 & 2 & 2 & 2 \\ 2 & 1 & 1 & 1 & 2 \\ 2 & 1 & 0 & 1 & 2 \\ 2 & 1 & 1 & 1 & 2 \\ 2 & 2 & 2 & 2 & 2 \end{array}$$

Пиксели с расстоянием $D_8 = 1$ являются восьмеркой соседей для элемента (x, y).

Заметим, что расстояния D_4 и D_8 между двумя элементами p и q не зависят от каких-либо путей, которые могли существовать между этими пикселями, поскольку в определении этих расстояний участвуют только координаты элементов. Однако, если мы выбираем в качестве меры m-смежность, то расстояние D_m между двумя элементами изображения определяется как длина кратчайшего m-пути между этими элементами. В этом случае расстояние между пикселями будет зависеть от значений всех пикселей вдоль этого пути, равно как и от значений их соседей. Например, рассмотрим следующую конфигурацию пикселей, и пусть элементы p, p_2 и p_4 имеют значение 1, а элементы p_1 и p_3 могут принимать значения 0 или 1:

$$\begin{matrix} & p_3 & p_4 \\ p_1 & p_2 & \\ p & & \end{matrix}$$

Предположим, что рассматривается смежность пикселей со значением 1, т.е. $V = \{1\}$. Если оба элемента p_1 и p_3 имеют значения 0, то длина кратчайшего m-пути (т.е. расстояние D_m) между p и p_4 равна 2. Если значение p_1 равно 1, то элементы p и p_2 больше не являются m-смежными (см. определение отношения m-смежности) и длина кратчайшего m-пути становится равной 3 (этот путь проходит через точки p, p_1, p_2, p_4). Аналогичные рассуждения имеют место в том случае, если значение p_3 равно 1 (а значение p_1 равно 0). В этом случае длина кратчайшего m-пути также равна 3. Наконец, если оба пикселя p_1 и p_3 имеют единичные значения, то длина кратчайшего m-пути между p и p_4 станет равной 4. В таком случае путь проходит через последовательность точек p, p_1, p_2, p_3, p_4.

2.6. Введение в математический аппарат, применяемый в цифровой обработке изображений

Этот раздел преследует две главные цели: (1) познакомить читателя с различными математическими инструментами, которые используются на протяжении всей книги и (2) помочь читателю ощутить, как именно этот аппарат используется, применяя его в разнообразных простых задачах обработки изображений, часть которых будет неоднократно появляться в дальнейших обсуждениях. По мере необходимости область применения этих инструментов будет расширяться в последующих главах.

> Перед тем, как двигаться дальше, читателю может быть полезно загрузить и изучить обзорный материал из раздела «Обучающие материалы» на сайте книги в Интернете. Этот обзор содержит вводный материал о матрицах и векторах, линейных системах, теории множеств и теории вероятности.

2.6.1. Поэлементные и матричные операции

Поэлементные операции, в которых участвуют одно или более изображений, всегда выполняются *попиксельно* над соответственными элементами изображений. Ранее в этой главе упоминалось, что изображения можно также эквивалентно рассматривать как матрицы. И в самом деле, во многих случаях операции над изображениями выполняются по правилам матричной алгебры (см. раздел 2.6.6). Именно по этой причине необходимо четко разграничить поэлементные и матричные операции. Рассмотрим, например, следующие два изображения размерами 2×2:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \text{ и } \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}.$$

Поэлементное произведение этих двух изображений вычисляется так:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} \\ a_{21}b_{21} & a_{22}b_{22} \end{bmatrix}.$$

Напротив, *матричное произведение* изображений определяется выражением

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{bmatrix}.$$

В последующем на протяжении книги мы всюду подразумеваем, что операции выполняются попиксельно, если не оговорено иное. Например, говоря о возведении цифрового изображения в степень, мы имеем в виду, что значение каждого пикселя по отдельности возводится в эту степень; говоря об операции деления одного изображения на другое, мы на самом деле подразумеваем, что деление выполняется для соответственных пикселей двух изображений и т. д.

2.6.2. Линейные и нелинейные преобразования

Одна из важнейших характеристик любого метода обработки изображений — это является ли он линейным или нелинейным. Рассмотрим оператор общего вида H, который строит выходное изображение $g(x, y)$ для данного входного изображения $f(x, y)$:

$$H[f(x,y)] = g(x,y). \tag{2.6-1}$$

Говорят, что оператор *H линейный*, если

$$H[a_i f_i(x,y) + a_j f_j(x,y)] = a_i H[f_i(x,y)] + a_j H[f_j(x,y)] = a_i g_i(x,y) + a_j g_j(x,y), \tag{2.6-2}$$

где $f_i(x, y)$ и $f_j(x, y)$ — любые изображения одинаковых размеров, а a_i и a_j — произвольные константы. Соотношение (2.6-2) показывает, что результат применения линейного оператора к сумме двух входных изображений совпадает с суммой результатов применения такого оператора к этим изображениям по отдельности. А также результат применения линейного оператора к изображению, умноженному на константу, идентичен умножению на эту константу результата применения оператора к исходному изображению. Первое свойство называется *аддитивностью*, а второе — *однородностью.*

Рассмотрим простой пример, где в качестве H используется оператор суммы Σ, функция которого состоит просто в суммировании его входов. Для проверки линейности этого оператора начнем с левой части (2.6-2) и попытаемся доказать, что она равна правой части:

$$\sum[a_i f_i(x,y) + a_j f_j(x,y)] = \sum a_i f_i(x,y) + \sum a_j f_j(x,y) =$$
$$= a_i \sum f_i(x,y) + a_j \sum f_j(x,y) = a_i g_i(x,y) + a_j g_j(x,y),$$

где первый шаг преобразования вытекает из дистрибутивности сложения. Таким образом, левая часть (2.6-2) равна правой, и можно заключить, что оператор суммы линейный.

Здесь всюду используется поэлементное суммирование, а не сумма всех пикселей изображения. Применение оператора суммы к одному изображению дает само это изображение.

Напротив, рассмотрим оператор max, функция которого состоит в нахождении максимального значения пикселя во входном изображении. Для наших целей простейший способ доказать, что этот оператор нелинейный, — найти контрпример, для которого нарушается условие (2.6-2). Рассмотрим следующие два изображения:

$$f_1 = \begin{bmatrix} 0 & 2 \\ 2 & 3 \end{bmatrix} \text{ и } f_2 = \begin{bmatrix} 6 & 5 \\ 4 & 7 \end{bmatrix},$$

и предположим, что константы $a_1 = 1$ и $a_2 = -1$. Проверку линейности опять начнем с левой части (2.6-2):

$$\max\left\{(1)\begin{bmatrix} 0 & 2 \\ 2 & 3 \end{bmatrix} + (-1)\begin{bmatrix} 6 & 5 \\ 4 & 7 \end{bmatrix}\right\} = \max\left\{\begin{bmatrix} -6 & -3 \\ -2 & -4 \end{bmatrix}\right\} = -2.$$

Действуя теперь с правой частью, получим

$$(1)\max\left\{\begin{bmatrix} 0 & 2 \\ 2 & 3 \end{bmatrix}\right\} + (-1)\max\left\{\begin{bmatrix} 6 & 5 \\ 4 & 7 \end{bmatrix}\right\} = 3 + (-1)7 = -4.$$

В данном случае левая и правая части (2.6-2) не равны друг другу, и тем самым доказано, что в общем случае оператор max является нелинейным.

Как мы увидим в трех следующих главах, особенно в главах 4 и 5, линейные операторы исключительно важны для обработки изображений, поскольку они опираются на значительную совокупность хорошо изученных теоретических и практических результатов. Нелинейные операторы исследованы значительно хуже, поэтому область их применения более ограничена. Однако мы познакомимся в последующих главах с несколькими нелинейными операциями обработки изображений, результаты которых значительно превосходят те, которые достигаются с помощью линейных аналогов.

2.6.3. Арифметические операции

Арифметические операции над изображениями являются поэлементными операциями, т. е., как уже говорилось в разделе 2.6.1, они применяются к паре соответственных пикселей двух изображений. Эти четыре арифметические операции обозначаются следующим образом:

$$\begin{aligned} s(x,y) &= f(x,y) + g(x,y), \\ d(x,y) &= f(x,y) - g(x,y), \\ p(x,y) &= f(x,y) \times g(x,y), \\ v(x,y) &= f(x,y) \div g(x,y). \end{aligned} \qquad (2.6\text{-}3)$$

Понятно, что эти операции применяются к соответственным парам элементов изображений f и g для $x = 0, 1, 2, \ldots, M-1$ и $y = 0, 1, 2, \ldots, N-1$, где, как обычно, M и N — число строк и столбцов изображений соответственно. Ясно, что s, d, p и v тоже являются изображениями с размерами $M \times N$. Заметим, что в так опреде-

ленной арифметике участвуют изображения одинаковых размеров. Следующие примеры показывают важную роль, которую играют арифметические операции в цифровой обработке изображений.

Пример 2.5. Сложение (усреднение) зашумленных изображений для уменьшения шума.

■ Рассмотрим зашумленное изображение $g(x,y)$, формируемое прибавлением шума $\eta(x,y)$ к исходному изображению $f(x,y)$, то есть

$$g(x,y) = f(x,y) + \eta(x,y), \tag{2.6-4}$$

где предполагается, что значения шума в каждой точке (x,y) являются некоррелированными[5] и имеют нулевое среднее значение. Целью нижеследующей процедуры является уменьшение уровня шума путем суммирования серии зашумленных изображений $\{g_i(x,y)\}$. Этот метод часто применяется для улучшения изображений.

Если шум удовлетворяет только что сформулированным условиям, то можно показать следующее (задача 2.20). Пусть изображение $\bar{g}(x,y)$ получено усреднением K изображений $g_i(x,y)$, отличающихся лишь шумом $\eta_i(x,y)$,

$$\bar{g}(x,y) = \frac{1}{K}\sum_{i=1}^{K} g_i(x,y), \tag{2.6-5}$$

откуда следует, что

$$E\{\bar{g}(x,y)\} = f(x,y) \tag{2.6-6}$$

и

$$\sigma^2_{\bar{g}(x,y)} = \frac{1}{K}\sigma^2_{\eta(x,y)}, \tag{2.6-7}$$

где $E\{\bar{g}(x,y)\}$ есть математическое ожидание $\bar{g}$, а $\sigma^2_{\bar{g}(x,y)}$ и $\sigma^2_{\bar{\eta}(x,y)}$ — дисперсии $\bar{g}$ и $\bar{\eta}$, все в точке (x,y). Стандартное отклонение в каждой точке усредненного изображения будет

$$\sigma_{\bar{g}(x,y)} = \frac{1}{\sqrt{K}}\sigma_{\eta(x,y)}. \tag{2.6-8}$$

Как следует из уравнений (2.6-7) и (2.6-8), при увеличении K величина отклонения (уровень шума) значения элемента в каждой точке (x,y) уменьшается. Поскольку $E\{\bar{g}(x,y)\} = f(x,y)$, это означает, что $\bar{g}(x,y)$ приближается к $f(x,y)$ с увеличением числа суммируемых зашумленных изображений. На практике изображения $g_i(x,y)$ должны являться *совмещенными*, чтобы уменьшить влияние расфокусировки или других искажений на выходном изображении.

Усреднение изображений находит важное применение в астрономии, где обычным является получение изображений при очень малом уровне освещенности и, как следствие, высоком уровне шума датчика, что зачастую делает отдельные снимки почти бесполезными для анализа. На рис. 2.26(*a*) представле-

---

[5] Вспомним, что дисперсия случайной переменной z со средним m определяется как $E\{(z-m)^2\}$, где $E\{\cdot\}$ есть математическое ожидание аргумента. Ковариация двух случайных переменных z_i и z_j определяется как $E\{(z_i - m_i)(z_j - m_j)\}$. Если переменные *некоррелированы*, их ковариация равна 0.

а *б* *в*
г *д* *е*

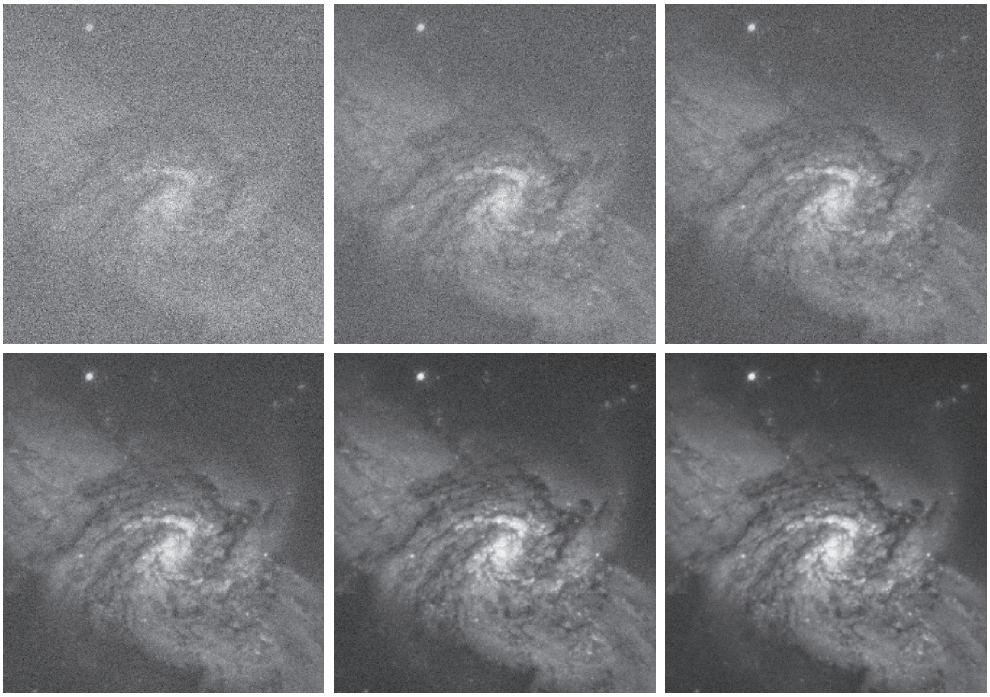


Рис. 2.26. (*а*) Изображение пары галактик NGC 3314, искаженное аддитивным гауссовым шумом. (*б*)—(*е*) Результаты усреднения 5, 10, 20, 50 и 100 зашумленных изображений. (Исходное изображение предоставлено агентством NASA)

но 8-битовое зашумленное изображение, искажения которого имитируются путем прибавления гауссова шума со средним ноль и стандартным отклонением 64 градации яркости. Изображение имеет типичный вид зашумленного снимка при слабой освещенности и выглядит бесполезным для практического использования. На изображениях рис. 2.26(*б*)—(*е*) представлены результаты усреднения 5, 10, 20, 50 и 100 изображений соответственно. Видно, что результат на рис. 2.26(*д*), полученный с $K = 50$, довольно хорошо очищен от шума. Изображение на рис. 2.26(*е*), полученное усреднением 100 зашумленных изображений, лишь немногим лучше по сравнению с рис. 2.26(*д*).

Изображения в этом примере получены при съемке пары галактик NGC 3314 с помощью космического телескопа Хаббл (NASA). Галактика NGC 3314 находится на расстоянии около 140 миллионов световых лет от Земли, в созвездии Гидры, расположенном в южном полушарии. Яркие звезды, образующие форму спирали возле центра передней галактики, сформировались из межзвездного газа и пыли.

Сложение является дискретным аналогом непрерывного интегрирования. В астрономических наблюдениях процесс, эквивалентный только что описан-

ному методу, состоит в использовании интегрирующих свойств ПЗС (приборов с зарядовой связью) или аналогичных сенсоров путем увеличения времени экспозиции. Дополнительное уменьшение уровня шума можно достичь путем охлаждения сенсоров. Тем не менее совместный эффект этих мер аналогичен усреднению набора зашумленных цифровых изображений. ■

Пример 2.6. Вычитание изображений для выделения различий.

■ Операция вычитания изображений часто применяется для выделения имеющихся различий между ними. Например, изображение на рис. 2.27(*б*) получено путем обнуления младшего бита яркости каждого пикселя изображения на рис. 2.27(*а*). Визуально эти изображения неразличимы, однако, как показывает рис. 2.27(*в*), если вычесть одно изображение из другого, различия становятся хорошо заметными. Черные пиксели (с нулевым значением) на разностном изображении указывают те области, где изображения рис. 2.27(*а*) и (*б*) одинаковы.

В качестве другого примера отработанного и успешного применения разностных изображений упомянем область медицинской интроскопии, которая называется *рентгенографией с использованием масок*. Рассмотрим разность двух изображений $f(x, y)$ и $h(x, y)$, выраженную формулой

$$g(x, y) = f(x, y) - h(x, y). \qquad (2.6\text{-}9)$$

> Обнаружение изменений с помощью вычитания изображений также применяется для сегментации изображений; этой теме посвящена глава 10.

В этом случае рентгеновское изображение части тела пациента регистрируется телекамерой, имеющей возможность усиления и размещаемой напротив рентгеновского источника вместо традиционной рентгеновской пленки. Данный метод предполагает введение контрастного вещества в систему кровообращения пациента. Первоначально в качестве *маски* $h(x, y)$ регистрируется серия изображений интересующего участка тела пациента, затем вводится контрастное вещество и регистрируется серия новых изображений — уже в качестве изобра-

а б в

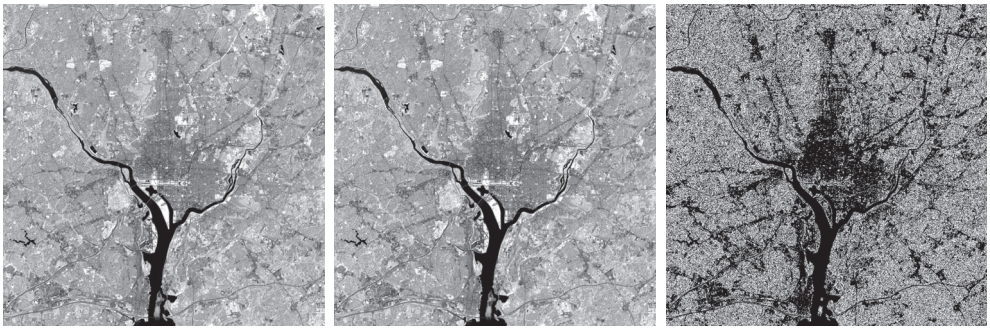


Рис. 2.27. (*а*) Инфракрасное изображение окрестностей г. Вашингтон, округ Колумбия. (*б*) Результат обнуления младшего бита каждого пикселя (*а*). (*в*) Разность изображений (*а*) и (*б*), приведенная для наглядности к диапазону яркостей [0, 255]

а *б*
в *г*

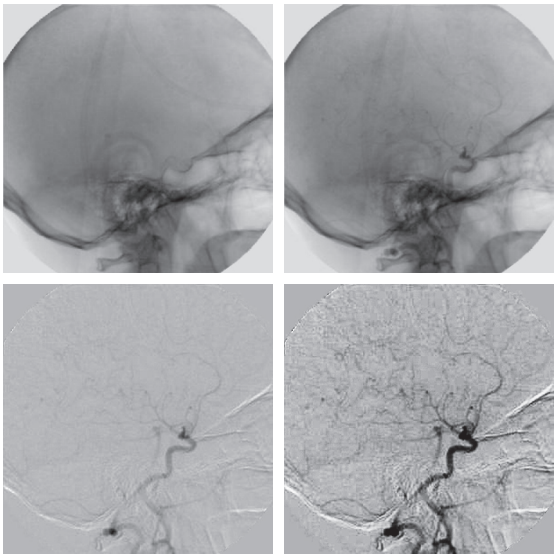


Рис. 2.28. Цифровая субтрактивная ангиография. (*а*) Изображение-маска. (*б*) Изображение, полученное после введения контрастного вещества в систему кровообращения. (*в*) Разность изображений (*а*) и (*б*). (*в*) Улучшенное разностное изображение. (Изображения (*а*) и (*б*) предоставлены Институтом изображений медицинского центра университета г. Утрехт, Нидерланды)

жения $f(x,y)$. Результирующий эффект вычитания маски $h(x,y)$ из изображения $f(x,y)$ заключается в том, что детали, различающиеся на изображениях $f(x,y)$ и $h(x,y)$, на выходном изображении приобретают повышенный контраст. Поскольку изображение может регистрироваться телекамерой, то эта процедура, в сущности, позволяет получить фильм, показывающий, как контрастное вещество распространяется по различным артериям в наблюдаемой области.

На рис. 2.28(*а*) представлена маска — рентгеновское изображение верхней части головы пациента *до* ввода в систему кровообращения контрастного йодистого вещества, а на рис. 2.28(*б*) — пример одного из кадров, полученных через некоторое время *после* введения контрастного вещества в систему кровообращения. Рис. 2.28(*в*) демонстрирует разность изображений (*а*) и (*б*), на которой начинает проявляться мелкая сеть кровеносных сосудов. Эти различия дополнительно усилены на рис. 2.28(*г*), который получен с помощью повышения контраста изображения (*в*) (методика повышения контраста рассматривается в следующей главе). Рис. 2.28(*г*) дает ясную картину распространения контрастного вещества по кровеносным сосудам в мозгу пациента. ■

Пример 2.7. Применение умножения и деления изображений для выравнивания фона.

■ Важной областью применения операций умножения (и деления) изображений является *выравнивание фона*. Предположим, что система регистрации формирует цифровые изображения, которые можно смоделировать как произведение «идеального изображения» $f(x,y)$ и функции неравномерности фона $h(x,y)$,

а б в

Рис. 2.29. Выравнивание фона. (*а*) Изображение нити накаливания и опоры, полученное сканирующим электронным микроскопом при 130-кратном увеличении. (*б*) Характер неравномерности фона. (*в*) Результат деления (*а*) на (*б*). (Исходное изображение предоставил Майкл Шефер, факультет геологических наук университета штата Орегон, г. Юджин)

т. е. $g(x,y) = f(x,y)\ h(x,y)$. Если изображение $h(x,y)$ известно, мы можем получить $f(x,y)$ умножением полученного изображения на обратное к $h(x,y)$, т. е. делением[6] g на h. Если $h(x,y)$ неизвестно, но есть доступ к системе получения изображений, мы можем приближенно найти функцию неравномерности фона, получив изображение поля с постоянной яркостью. Даже если доступ к регистрирующей системе невозможен, часто можно оценить характер неравномерности фона непосредственно по изображению, как мы увидим в разделе 9.6. Пример выравнивания фона показан на рис. 2.29.

Другим типичным примером умножения изображений является *маскирование*, иначе называемое операциями с *областью интереса*. Изображенный на рис. 2.30 процесс состоит просто в умножении данного изображения на изображение-маску, пиксели которого имеют значение 1 в области интереса и значение 0 в прочих местах. Маска может содержать несколько областей интереса, и их форма может быть произвольной, хотя по соображениям простоты реализации часто используются прямоугольные области интереса. ■

Прежде чем завершить данный раздел, уместно сделать несколько комментариев о реализации арифметических операций над изображениями. На практике большинство изображений представляются в виде 8-битового сигнала (даже 24-битовые цветные изображения состоят из трех независимых 8-битовых каналов). Тем самым значения изображений не могут выходить за диапазон [0, 255]. Когда изображения запоминаются в стандартных форматах, таких как TIFF или JPEG, преобразование к указанному диапазону выполняется по правилам, зависящим от используемой программы. Например, значения разности двух 8-битовых изображений могут находиться в интервале от –255 до 255, а зна-

[6] Если $h(x, y)$ и $g(x, y)$ являются логарифмированными значениями яркости объекта (как это чаще всего и бывает), то вместо операции деления следует использовать операцию вычитания. – Прим. ред. перев.

а б в

Рис. 2.30. (*а*) Рентгеновское цифровое изображение зубов. (*б*) Маска области интереса для выделения запломбированных зубов (белое соответствует значению 1, черное — значению 0). (*в*) Произведение (*а*) и (*б*)

чения суммы — от 0 до 510. Многие программные пакеты при преобразовании в 8-битовое представление просто устанавливают в 0 все отрицательные значения и в 255 — все значения, превышающие эту величину.

Подход, который для заданного изображения f, являющегося результатом выполнения любых арифметических операций над изображениями, гарантирует высокую точность и полное покрытие диапазона заданного числа бит, состоит в следующем. Сначала выполняется операция

$$f_m = f - \min(f), \tag{2.6-10}$$

в результате которой формируется изображение с минимальным значением, равным 0. Затем применяется операция

$$f_s = K\left[f_m / \max(f_m)\right], \tag{2.6-11}$$

после чего образуется изображение f_s со значениями, пропорционально преобразованными в диапазон [0, K]. При работе с 8-битовыми изображениями, задавая $K = 255$, получим изображение со значениями яркости, занимающими весь 8-битовый диапазон от 0 до 255. Аналогичные комментарии справедливы для цифровых изображений с разрядностью 16 бит и выше. Этот подход применим для любых арифметических операций. При выполнении деления действует добавочное требование: к пикселям изображения-делителя следует прибавить малое значение во избежание деления на ноль.

2.6.4. Теоретико-множественные и логические операции

В этом разделе мы кратко рассмотрим некоторые важные теоретико-множественные и логические операции. Мы также введем в рассмотрение понятие нечеткого множества.

Базовые операции над множествами

Пусть A — множество, состоящее из *упорядоченных пар* вещественных чисел. Если $a = (a_1, a_2)$ есть *элемент* A, то этот факт обозначается символической записью

$$a \in A. \tag{2.6-12}$$

В противном случае, т. е. когда a не является элементом A, используется запись

$$a \notin A. \tag{2.6-13}$$

Множество, не содержащее ни одного элемента, называется *пустым множеством* и обозначается символом ∅.

Множество задается путем перечисления его содержимого в фигурных скобках: {·}. Например, записывая выражение в форме $C = \{w \mid w = -d,\ d \in D\}$, мы имеем в виду, что множество C состоит из элементов w, которые строятся умножением на –1 каждого элемента множества D. Один из способов использования множеств в обработке изображений состоит в том, чтобы элементами множеств были *координаты* пикселей (упорядоченные пары целых чисел), представляющих объекты или другие интересующие признаки на изображении.

Если все элементы множества A являются также элементами другого множества B, то говорят, что A есть *подмножество* множества B, что символически обозначается

$$A \subseteq B. \tag{2.6-14}$$

Объединение двух множеств A и B, которое обозначается

$$C = A \cup B, \tag{2.6-15}$$

есть по определению множество всех элементов, принадлежащих либо множеству A, либо множеству B, либо обоим множествам одновременно. Аналогично *пересечение* двух множеств A и B, которое обозначается

$$D = A \cap B, \tag{2.6-16}$$

есть по определению множество всех элементов, принадлежащих одновременно обоим множествам A и B. Два множества A и B называются *непересекающимися* или *взаимоисключающими*, если у них нет общих элементов. В этом случае

$$A \cap B = \varnothing. \tag{2.6-17}$$

Универсальное множество (или *генеральная совокупность*) U есть множество всех элементов в данном приложении. По определению, все элементы множеств в данном приложении являются элементами универсального множества, определенного для этого приложения. Например, при работе с множеством вещественных чисел универсальным множеством является числовая прямая, содержащая все вещественные числа. В цифровой обработке изображений в качестве универсального множества обычно определяется прямоугольник, содержащий все пиксели изображения.

Дополнение множества A есть множество элементов, не содержащихся в A:

$$A^c = \{w \mid w \notin A\}. \tag{2.6-18}$$

Разность двух множеств A и B обозначается $A \setminus B$ и определяется следующим образом:

$$A \setminus B = \{w \mid w \in A, w \notin B\} = A \cap B^c. \tag{2.6-19}$$

Видно, что это множество состоит из элементов A, которые не входят в множество B. Можно в качестве примера определить дополнение в терминах универсального множества и разности множеств: $A^c = U \setminus A$.

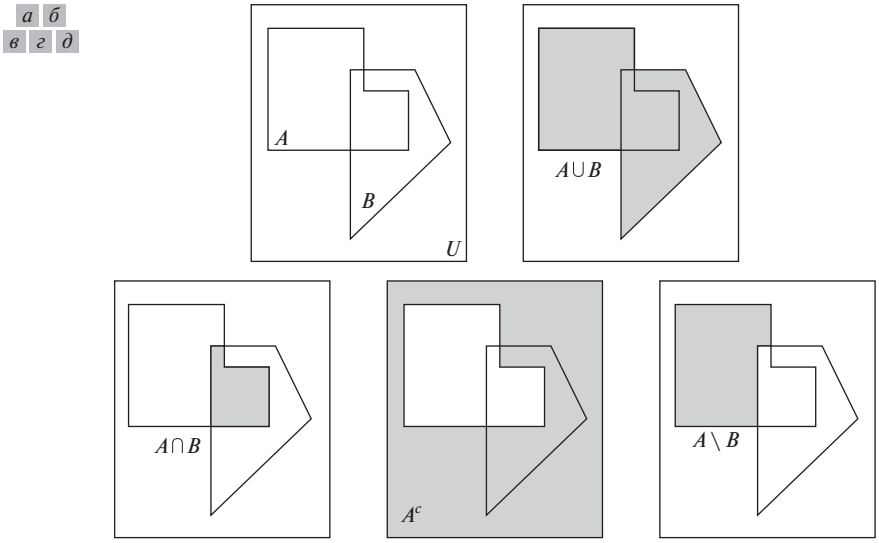


Рис. 2.31. (*а*) Два множества координат на плоскости, *A* и *B*. (*б*) Объединение множеств *A* и *B*. (*в*) Пересечение множеств *A* и *B*. (*г*) Дополнение множества *A*. (*д*) Разность множеств *A* и *B*. Затемненные области на рисунках (*б*)—(*д*) представляют элементы множеств, являющихся результатом указанной операции

Рис. 2.31 иллюстрирует введенные выше понятия, где универсальное множество есть множество координат внутри изображенного прямоугольника, а множества *A* и *B* — множества координат, заключенные внутри показанных границ. Результат выполнения той или иной операции над множествами показан на каждом из рисунков темным цветом.[7]

В обсуждении выше принадлежность множеству основывалась только на координатах элемента. При работе с изображениями делается неявное предположение, что все пиксели множества имеют одно и то же значение, поскольку в определениях теоретико-множественных операций не участвуют значения яркости (например, мы не определили, какую яркость имеют элементы пересечения двух множеств). Единственный вариант, при котором показанные на рис. 2.31 операции имеют смысл, — если изображения, содержащие эти множества, двоичные. В этом случае можно говорить о принадлежности множеству на основании только координат, предполагая, что все элементы множеств имеют одинаковую яркость. Этот вопрос подробнее рассматривается в следующем подразделе.

При работе с полутоновыми изображениями вышеизложенные понятия неприменимы, потому что необходимо указать значения всех пикселей для результата операции над множествами. На самом деле, как мы увидим в разделах 3.8 и 9.6, в случае полутонов операции объединения и пересечения обыч-

[7] Операции, заданные соотношениями (2.6-12)—(2.6-19), являются базовыми для алгебры множеств, которая исходит из ряда свойств, таких как *законы коммутативности*: $A \cup B = B \cup A$ и $A \cap B = B \cap A$, и выводит из них обширную теорию, основанную на операциях над множествами. Изучение алгебры множеств выходит за пределы нашего рассмотрения, но о существовании ее следует знать.

но определяются как соответственно максимум и минимум для пары соответственных пикселей, а дополнение определяется как попарные разности между константой и яркостью каждого пикселя. Тот факт, что мы имеем дело с парами соответственных пикселей, указывает на поэлементный характер операций над полутоновыми множествами, как определялось в разделе 2.6.1. Следующий пример кратко иллюстрирует теоретико-множественные операции над полутоновыми изображениями. Дальнейшее рассмотрение этих понятий проводится в двух упомянутых выше разделах.

Пример 2.8. Операции над множествами с учетом яркости изображений.

■ Пусть пиксели полутонового изображения представляются множеством A, элементами которого являются тройки вида (x, y, z), где x и y — пространственные координаты, а z обозначает яркость в точке с указанными координатами, как уже говорилось в разделе 2.4.2. Можно определить *дополнение* A как множество $A^c = \{(x, y, K - z) | (x, y, z) \in A\}$, которое есть просто множество пикселей A с яркостями, полученными вычитанием их исходных значений из константы K. Эта константа равна $2^k - 1$, где k — число бит, используемых для представления яркости z. Пусть в качестве A выступает 8-битовое полутоновое изображение на рис. 2.32(*а*) и предположим, что мы хотим сформировать негатив A с помощью операций над множествами. Просто построим множество $A_n = A^c = \{(x, y, 255 - z) | (x, y, z) \in A\}$. Заметим, что координаты здесь пробегают всю область, так что размеры изображения A_n те же, что и у A. Результат представлен на рис. 2.32(*б*).

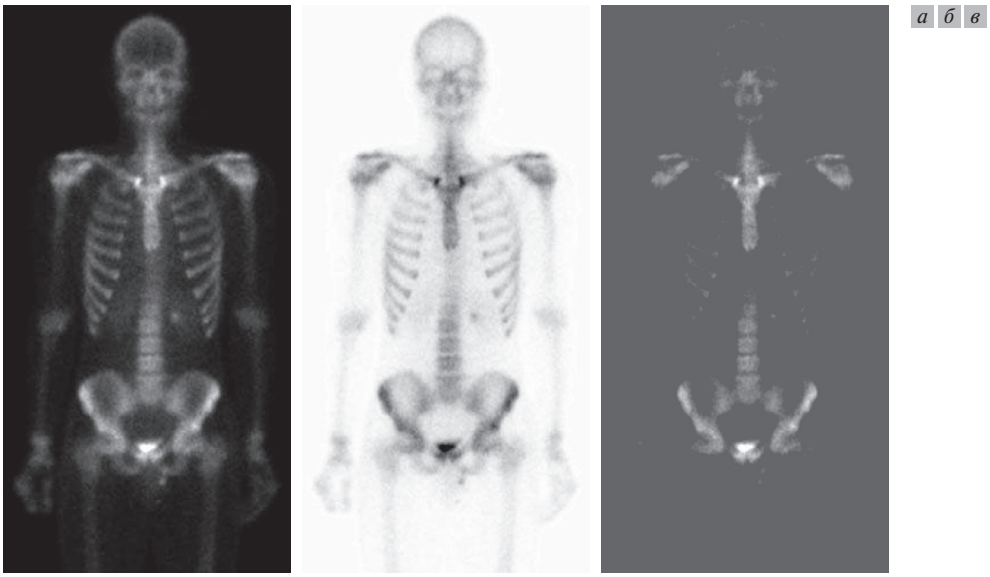


Рис. 2.32. Операции над множествами с учетом яркости изображений. (*а*) Исходное изображение. (*б*) Негативное изображение, полученное как дополнение множества. (*в*) Объединение (*а*) и изображения с постоянной яркостью. (Исходное изображение предоставила компания G. E. Medical Systems)

Объединение двух полутоновых множеств A и B можно определить как множество

$$A \cup B = \left\{ \max_z(a,b) \middle| a \in A, b \in B \right.,$$

т. е. объединение двух полутоновых изображений есть массив, сформированный из максимальных значений яркости каждой пары соответственных пикселей. Снова заметим, что координаты здесь пробегают всю область, так что объединение A и B является изображением с теми же размерами, что и сами изображения A и B. Для иллюстрации предположим, что в качестве A опять используется изображение на рис. 2.32(*а*), а B — прямоугольный массив с такими же размерами, как и A, у которого значения z всех элементов равны утроенной величине средней яркости m пикселей A. Рис. 2.32(*в*) демонстрирует результат объединения этих множеств, в котором из A берутся все элементы со значениями больше $3m$, а остальные пиксели имеют значения $3m$, что соответствует серому цвету. ■

Логические операции

При работе с двоичными изображениями можно говорить о множествах пикселей *фона* (со значениями 0) и *переднего плана* (со значениями 1). Тогда, если определить объекты как области, состоящие из пикселей переднего плана, то показанные на рис. 2.31 операции над множествами превращаются в операции над координатами объектов в двоичном изображении. Имея дело с двоичными изображениями, обычно говорят об объединении, пересечении и дополнении как о *логических* операциях OR (ИЛИ, логическое сложение), AND (И, логическое умножение) и NOT (НЕ, отрицание). Слово «логические» указывает на происхождение из формальной логики, в которой 1 и 0 означают соответственно истинное и ложное значения.

Рассмотрим две области, т. е. множества A и B, состоящие из пикселей переднего плана. Операция OR над этими двумя множествами дает множество элементов (пар координат), принадлежащих либо A, либо B, либо обоим множествам одновременно. Операция AND дает множество элементов, являющихся общими для A и B. Операция NOT в применении к множеству A дает множество элементов, не принадлежащих A. Поскольку речь идет об изображении, где A есть данное множество пикселей переднего плана, множество NOT(A) состоит из всех пикселей изображения, не входящих в A, т. е. всех пикселей фона и, возможно, других пикселей переднего плана. Эту операцию можно рассматривать как установку значений всех элементов A в 0 (черное), а элементов вне A в 1 (белое). Рис. 2.33 иллюстрирует эти операции. Обратите внимание в четвертом ряду, что результат показанной операции есть множество пикселей переднего плана, которые принадлежат A, но не принадлежат B, что совпадает с определением разности множеств (2.6-19). В последнем ряду на рисунке представлена операция XOR (исключающее ИЛИ), результатом которой является множество пикселей переднего плана, принадлежащих A или B, но не обоим множествам одновременно. Заметим, что вышеописанные операции применяются к областям, которые могут иметь произвольную форму и размеры. В этом состоит отличие от обсуждавшихся ранее полутоновых операций, которые являются поэлементными и потому требуют, чтобы множества имели одни и те же про-

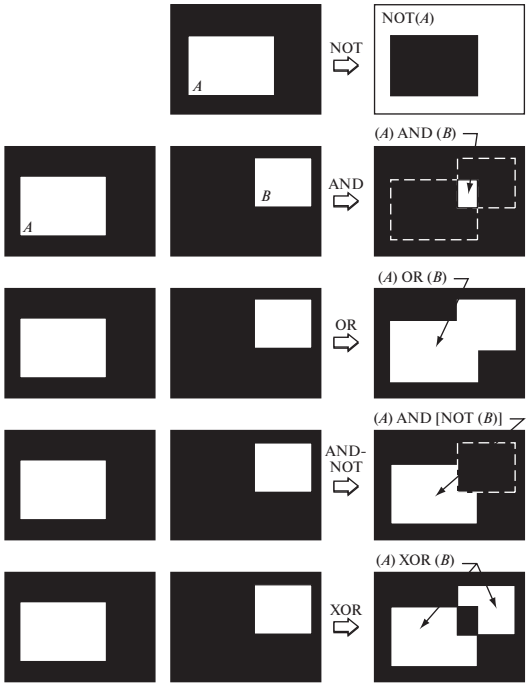


Рис. 2.33. Логические операции над пикселями переднего плана (белыми). Черным цветом показаны пиксели с двоичным значением 0, белым — со значением 1. Пунктирные линии нанесены только для удобства и не являются частью результата

странственные размеры. Иными словами, в операциях над полутоновыми множествами участвуют полные изображения, а не их области.

Теоретически необходимо реализовать только три логические операции AND, OR и NOT, поскольку они образуют *функционально полный* класс в том смысле, что любая другая логическая операция может быть получена путем комбинирования только этих основных операций, как показано в четвертом ряду на рис. 2.33, где операция разности множеств реализована с помощью AND и NOT. Логические операции широко используются при морфологической обработке изображений, которая рассматривается в главе 9.

Нечеткие множества

Изложенные выше теоретико-множественные и логические результаты оперируют *четкими* понятиями в том смысле, что элемент либо принадлежит, либо не принадлежит множеству. В некоторых приложениях это сильно ограничивает возможности. Рассмотрим простой пример. Предположим, что мы хотим поделить всех людей на две категории — молодых и немолодых. Применяя четкие множества, обозначим U множество всех людей и пусть A — подмножество U, которое назовем множеством молодых людей. Чтобы задать множество A, необходима *функция принадлежности*, которая сопоставляет каждому человеку из U значение 0 или 1. Если элементу U сопоставлено значение 1, то этот элемент принадлежит A, и наоборот. Поскольку мы оперируем двузначной логикой, функ-

ция принадлежности просто задает порог для возраста, при котором (и меньших значениях) человек считается молодым, а выше которого — немолодым. Предположим, что мы определили как молодых всех людей, имеющих возраст 20 лет и меньше. Сразу появляется трудность: человек, чей возраст составляет 20 лет и 1 секунду, не должен входить в множество молодых людей. Такое ограничение возникает вне зависимости от значения порога, принятого для классификации людей как молодых. Хотелось бы иметь более гибкое определение того, что подразумевается под «молодостью», т. е. иметь *постепенный* переход из молодых в немолодые. Теория *нечетких множеств* реализует эту идею с помощью функций принадлежности, которые постепенно меняются в интервале между граничными значениями 1 (определенно молодой) и 0 (определенно немолодой). С помощью нечетких множеств можно делать утверждение, что некоторый человек на 50 % молодой (находится в середине перехода от молодого к немолодому). Иначе говоря, возраст представляет собой неточное понятие, и нечеткая логика дает возможность работы с такими понятиями. Нечеткие множества будут подробнее рассматриваться в разделе 3.8.

2.6.5. Пространственные операции

Пространственные операции выполняются непосредственно над значениями пикселей данного изображения. Эти операции можно разделить на три широких класса: (1) поэлементные операции (поточечная обработка), (2) операции над окрестностью (локальная обработка) и (3) геометрические преобразования (глобальная обработка).

Поэлементные операции

Простейшая операция над цифровым изображением состоит в изменении значений отдельных пикселей исходя из их яркости. Такой способ обработки может быть выражен *функцией градационного преобразования* (также называемой *функцией преобразования интенсивностей* или *функцией отображения*) вида

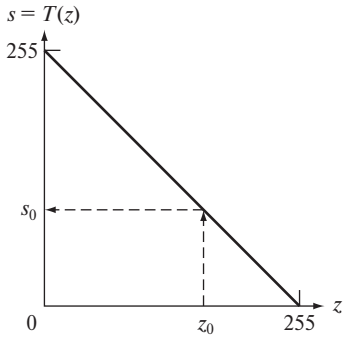


Рис. 2.34. Функция градационного преобразования, используемая для получения негатива 8-битового изображения. Пунктирные стрелки показывают, как произвольное значение исходной яркости z_0 преобразуется в значение яркости s_0 соответственного пикселя на результирующем изображении

$$s = T(z), \tag{2.6-20}$$

где z — яркость пикселя исходного изображения, а s — полученная после преобразования яркость соответственного пикселя на обработанном изображении. Например, на рис. 2.34 изображена функция преобразования, используемая для получения негатива 8-битового изображения, такого как на рис. 2.32(*б*), который был получен с помощью операций над множествами. В главе 3 рассматривается ряд методов задания функций градационного преобразования.

Операции над окрестностью

Обозначим S_{xy} множество координат окрестности произвольной точки (x,y) изображения f. При обработке окрестности генерируется соответственный (имеющий те же координаты) пиксель в выходном изображении g таким образом, что его значение определяется с помощью заданной операции над пикселями исходного изображения с координатами из S_{xy}. Например, пусть эта операция состоит в вычислении среднего значения яркости пикселей прямоугольной окрестности с размерами $m \times n$ и центром в точке (x,y). Координаты пикселей в этой области составляют множество S_{xy}. Рис. 2.35(*а*) и (*б*) иллюстрируют такой процесс. Эту операцию можно выразить уравнением

$$g(x,y) = \frac{1}{mn} \sum_{(r,c) \in S_{xy}} f(r,c), \tag{2.6-21}$$

где r и c суть координаты строки и столбца для тех пикселей, координаты которых входят в множество S_{xy}. Изображение g формируется путем варьирования координат (x,y) так, что центр окрестности передвигается по изображению f от пикселя к пикселю и в каждом новом месте повторяется та же самая операция над окрестностью. Например, изображение на рис. 2.35(*г*) получено таким способом с помощью окрестности размерами 41×41. Достигаемый эффект выражается в локальной расфокусировке исходного изображения. Такой способ обработки применяется, например, для устранения мелких деталей и тем самым выявления «пятен», соответствующих самым крупным областям изображения. Локальная обработка рассматривается в главах 3 и 5, а также в других местах книги.

Геометрические преобразования и совмещение изображений

Геометрические преобразования изменяют пространственные взаимосвязи между пикселями на изображении. Геометрические преобразования часто называют *преобразованиями резинового листа*, поскольку их можно представить себе как процесс распечатки изображения на листе из резины и дальнейшее растягивание этого листа в соответствии с определенными правилами. С точки зрения цифровой обработки изображений геометрические преобразования состоят из следующих двух основных операций: (1) пространственное преобразование координат и (2) *интерполяция значений яркости*, при которой происходит присвоение значений яркости точкам изображения, подвергнутого пространственному преобразованию.

Преобразование координат может быть выражено в форме уравнения

$$(x,y) = T\{(v,w)\}, \tag{2.6-22}$$

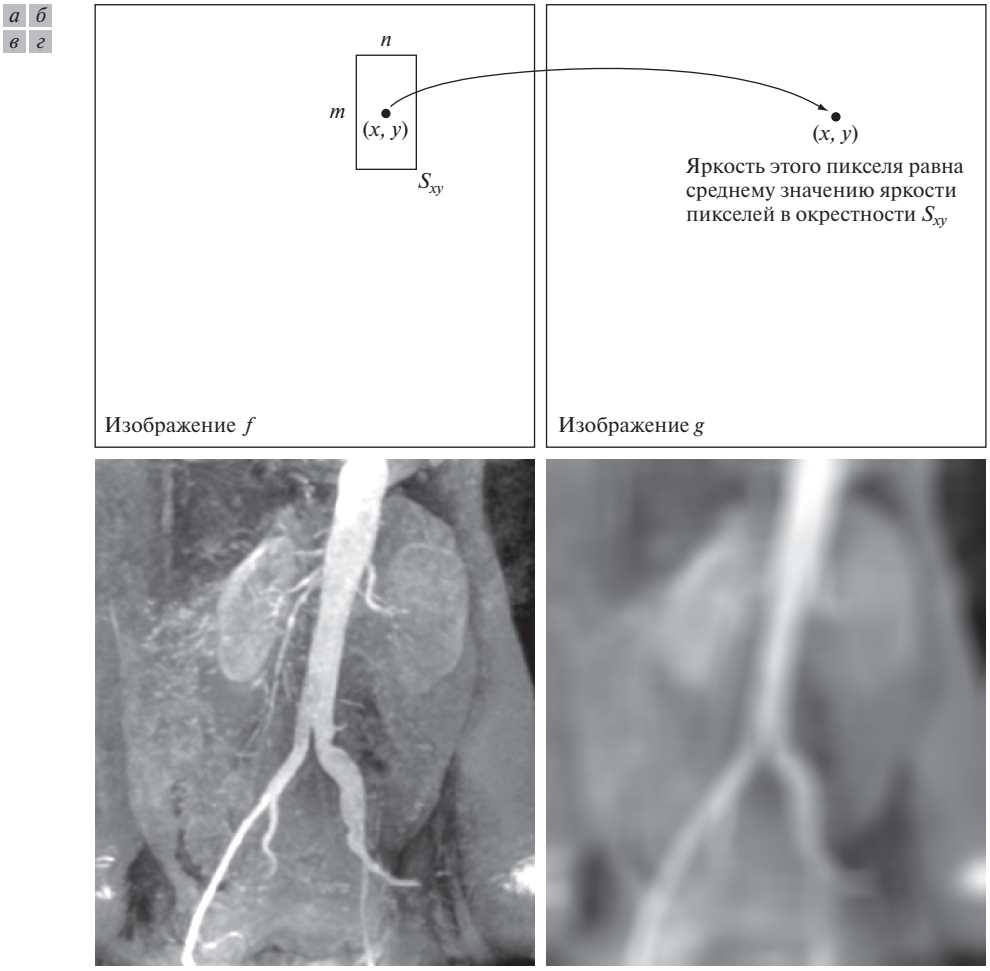
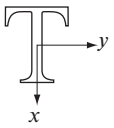
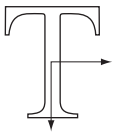
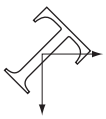
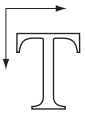
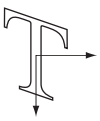
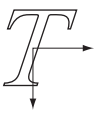


Рис. 2.35. Локальное усреднение с помощью обработки окрестности. Процедура для прямоугольной окрестности показана на рисунках (*а*) и (*б*). (*в*) Ангиограмма аорты из раздела 1.3.2. (*г*) Результат использования уравнения (2.6-21) с $m = n = 41$. Размеры изображений 790×686 пикселей

где (v, w) — координаты пикселя на исходном изображении, а (x, y) — координаты соответственного пикселя на преобразованном изображении. Например, преобразование $(x, y) = T\{(v, w)\} = (v/2, w/2)$ сжимает исходное изображение до половины первоначального размера в каждом направлении. Одним из наиболее часто применяемых преобразований координат является *аффинное преобразование* [Wolberg, 1990], которое имеет общую форму

$$[x \;\; y \;\; 1] = [v \;\; w \;\; 1]\mathbf{T} = [v \;\; w \;\; 1]\begin{bmatrix} t_{11} & t_{12} & 0 \\ t_{21} & t_{22} & 0 \\ t_{31} & t_{32} & 1 \end{bmatrix}. \qquad (2.6\text{-}23)$$

Таблица 2.2. Аффинные преобразования на основе уравнения (2.6-23)

Название преобразования	Аффинная матрица **T**	Преобразование координат	Пример
Тождественное преобразование	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = w$	
Изменение масштаба	$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = c_x v$ $y = c_y w$	
Поворот	$\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v\cos\theta - w\sin\theta$ $y = v\sin\theta + w\cos\theta$	
Параллельный перенос (сдвиг)	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$	$x = v + t_x$ $y = w + t_y$	
Вертикальный скос	$\begin{bmatrix} 1 & 0 & 0 \\ s_v & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v + s_v w$ $y = w$	
Горизонтальный скос	$\begin{bmatrix} 1 & s_h & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = s_h v + w$	

В зависимости от выбранных значений элементов матрицы **T**, это преобразование позволяет осуществлять изменение масштаба, поворот, сдвиг или скос множества координатных точек. В табл. 2.2 указаны значения элементов матрицы, используемые для реализации этих преобразований. Реальная сила матричного представления (2.6-23) состоит в том, что оно предлагает единый каркас для выполнения последовательности различных операций. Например, чтобы изменить размеры изображения, повернуть его и передвинуть результат в другое место, достаточно просто сформировать матрицу 3×3, равную произведению матриц масштабирования, поворота и параллельного переноса из табл. 2.2.

Вышеописанные преобразования перемещают пиксели изображения в новые места. Для завершения процесса необходимо присвоить им значения яркости. Эта задача решается с помощью интерполяции яркостей, которая уже рассматривалась в разделе 2.4.4. Там мы начали с примера, связанного с увеличением изображения, и рассмотрели вопрос о присвоении значений яркости пикселей на новых местах. Увеличение — это просто масштабное преобразование, указанное во второй строке табл. 2.2, и для других преобразований в табл. 2.2 применимы аналогичные способы присвоения значений яркости перемещен-

ным пикселям, как и те, что рассматривались для увеличения. Как и в разделе 2.4.4, при выполнении этих преобразований можно использовать интерполяцию по ближайшему соседу, билинейную и бикубическую интерполяции.

На практике уравнение (2.6-23) можно использовать двумя способами. Первый, который называется *прямое отображение*, состоит в сканировании пикселей исходного изображения с вычислением в каждой координатной точке (v, w) пространственных координат (x, y) соответственного пикселя на выходном изображении, прямо применяя уравнение (2.6-23). С подходом, основанным на прямом отображении, связана та проблема, что два и более пикселей исходного изображения могут преобразоваться в одно и то же место на выходном изображении, так что встает вопрос о том, как совместить несколько полученных значений в одном выходном пикселе. Кроме того, может оказаться, что некоторым пикселям выходного изображения не будет присвоено вообще никакого значения. При втором подходе, называемом *обратным отображением*, сканируются координаты пикселей выходного изображения и для каждой точки (x, y) вычисляется соответственная позиция в исходном изображении с помощью соотношения $(u, v) = T^{-1}(x, y)$. Затем в найденной точке выполняется интерполяция одним из способов, описанных в разделе 2.4.4, с использованием значений соседних пикселей исходного изображения, в результате чего находится яркость очередного пикселя на выходном изображении. Обратное отображение реализуется более эффективно, чем прямое, и используется в многочисленных коммерческих реализациях пространственных преобразований (например, этот подход используется в программном пакете MATLAB).

Пример 2.9. Поворот изображения с интерполяцией яркости.

■ Целью этого примера является проиллюстрировать поворот изображения с помощью аффинного преобразования. На рис. 2.36(*а*) показано исходное изображение с разрешением 300 dpi, а рис. 2.36(*б*)—(*г*) демонстрируют результаты поворота этого изображения на 21° с интерполяцией яркости по ближайшему соседу, билинейной и бикубической интерполяцией соответственно. Поворот

а б в г

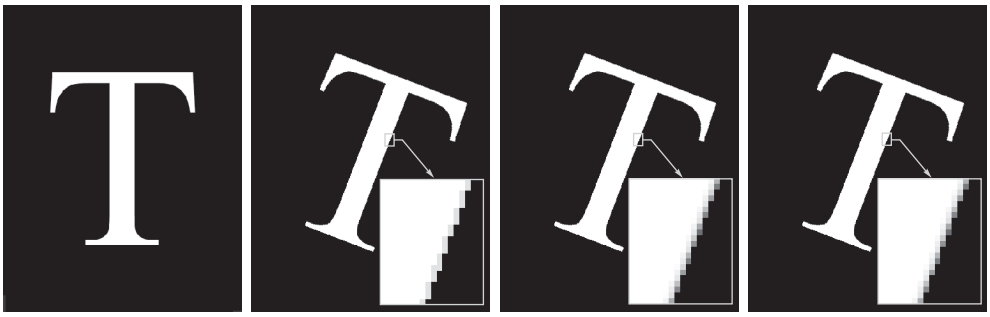


Рис. 2.36. (*а*) Изображение буквы T с разрешением 300 dpi. (*б*) Результат поворота на 21° с использованием интерполяции по ближайшему соседу для присвоения значений яркости преобразованным пикселям. (*в*) То же при использовании билинейной интерполяции. (*г*) То же при использовании бикубической интерполяции. Увеличенные фрагменты показывают детали перепада для трех интерполяционных методов

является одним из наиболее требовательных геометрических преобразований в плане сохранения прямолинейности объектов, присутствующих в исходном изображении. Как видно из рисунка, интерполяция по ближайшему соседу приводит к появлению наиболее ступенчатых краев, а билинейная интерполяция существенно улучшает результат, как отмечалось в разделе 2.4.4. Как и прежде, использование бикубической интерполяции немного повышает резкость. В самом деле, при сравнении увеличенных фрагментов на рис. 2.36(*в*) и (*г*) видно, что в середине фрагмента число вертикальных серых «блоков» в области перехода яркости от светлого к темному на рис. 2.36(*в*) больше, чем в соответственной области (*г*), что говорит о более высокой резкости перепада на последнем изображении. Сходные результаты можно получить и при других пространственных преобразованиях из табл. 2.2, которые требуют интерполяции (тождественное преобразование не требует, как и параллельный перенос, если величина сдвига по каждой координатной оси выражается целым числом пикселей). В этом примере использовалось обратное отображение, рассмотренное в предыдущем абзаце. ■

Важной прикладной задачей цифровой обработки изображений является *совмещение изображений*, т. е. взаимное выравнивание двух или более изображений одной и той же сцены. В рассмотрении выше мы считали известным вид функции, необходимой для получения желаемого результата преобразования исходного изображения. При совмещении изображений у нас помимо исходного есть еще преобразованное изображение, но в общем случае неизвестно конкретное преобразование, которое привело к такому результату. Задача состоит в том, чтобы по двум изображениям оценить функцию преобразования и затем воспользоваться ей для совмещения этих изображений. Для уточнения терминологии будем называть исходным то изображение, которое мы хотим преобразовать, а *эталонным* изображением назовем то, с которым хотим совместить исходное изображение.

Например, может представлять интерес совмещение двух или более изображений, полученных приблизительно в одно и то же время с помощью двух разных систем медицинской интроскопии, например ЯМР (томография на основе ядерного магнитного резонанса) и ПЭТ (позитронная эмиссионная томография) или изображения, полученные одним и тем же прибором в разное время, например спутниковые изображения интересующего участка местности, снятые с интервалом в несколько дней, месяцев или даже лет. В любом случае для комбинирования изображений или проведения их сравнения и количественного анализа необходимо компенсировать геометрические искажения, возникшие из-за различий угла наблюдения, расстояния и ориентации регистрирующей системы, разрешения сенсора, изменения положения объекта и прочих факторов.

Один из главных подходов к решению изложенной задачи состоит в использовании *точек привязки*, которые представляют собой такое подмножество соответственных пикселей, положение которых на исходном и эталонном изображениях точно известно. Для выбора системы точек привязки существуют различные методики, ориентированные на конкретные приложения, от интерактивного выбора точек до использования алгоритмов автоматического обнаружения таких точек. Так, некоторые системы формирования изображений

обладают характерными особенностями, имеющими физическую природу, такими как специфические элементы конструкции самого сенсора (например небольшие металлические метки на нем). В результате прямо на изображении в процессе его формирования возникает множество точек с известными координатами (эти точки называют *опорными* или точками *ризо-маркировки*), которые служат основой системы точек привязки.

Задача оценивания функции преобразования — это задача моделирования. Например, предположим, что на исходном и эталонном изображениях имеется по четыре точки привязки. Простая модель преобразования, основанная на билинейной аппроксимации, задается выражениями:

$$x = c_1 v + c_2 w + c_3 vw + c_4 \qquad (2.6\text{-}24)$$

и

$$y = c_5 v + c_6 w + c_7 vw + c_8, \qquad (2.6\text{-}25)$$

где на этапе оценивания (v, w) и (x, y) — координаты точек привязки в исходном и эталонном изображениях соответственно. Поскольку имеется четыре пары соответственных точек привязки в двух изображениях, можно выписать восемь уравнений (2.6-24) и (2.6-25), а затем решить полученную систему уравнений относительно коэффициентов c_i, $i = 1, 2, ..., 8$. Эти коэффициенты задают модель геометрических преобразований всех пикселей одного изображения в такие координаты пикселей на другом изображении, при которых обеспечивается совмещение изображений.

Как только коэффициенты получены, уравнения (2.6-24) и (2.6-25) можно использовать для преобразования всех пикселей исходного изображения в пиксели нового изображения, которое должно совместиться с эталонным при условии правильного выбора точек привязки[8]. В ситуациях, когда для удовлетворительного совмещения изображений четырех точек недостаточно, часто применяемый метод состоит в выборе большего числа точек привязки и использовании четырехугольников, образованных четверками соседних точек, в качестве более мелких изображений. Эти субизображения совмещаются, как описано выше, так что все пиксели внутри четырехугольника преобразуются с помощью коэффициентов, найденных на основании четырех точек привязки, являющихся его вершинами. Затем та же процедура повторяется для следующей четверки точек привязки и так далее, пока не будут обработаны все четырехугольники. Разумеется, могут применяться области более сложной формы, чем четырехугольные, и более сложные модели функции преобразования, например полиномиальные с нахождением приближения методом наименьших квадратов. В общем случае число требуемых точек привязки и сложность модели, которые необходимы для удовлетворительного решения задачи, зависят от степени серьезности геометрического искажения. Наконец, необходимо помнить, что преобразования вида (2.6-24) и (2.6-25) или любые другие модели для той же цели задают только отображение пространственных координат пикселей на исходном изображении. Необходимо еще провести интерполяцию яркости пикселей на результирующем изображении с помощью любого из рассмотренных выше методов.

---

[8] И, что не менее важно, при правильно выбранной модели искажения. — *Прим. перев.*

Пример 2.10. Совмещение изображений.

■ На рис. 2.37(*а*) показано эталонное изображение, а рис. 2.37 демонстрирует то же изображение после геометрического искажения в форме скоса по вертикали и по горизонтали. Наша цель — выбрать точки привязки на эталонном (и искаженном) изображениях и по ним совместить изображения. Выбранные (вручную) точки привязки показаны в виде маленьких белых квадратиков вблизи углов изображений (в данном случае нам достаточно всего четырех точек привязки, поскольку искажающий оператор есть линейный скос в обоих направлениях). На рис. 2.37(*в*) показан результат привязки исходного изображения по этим точкам с помощью процедуры, изложенной в предыдущем

а б
в г

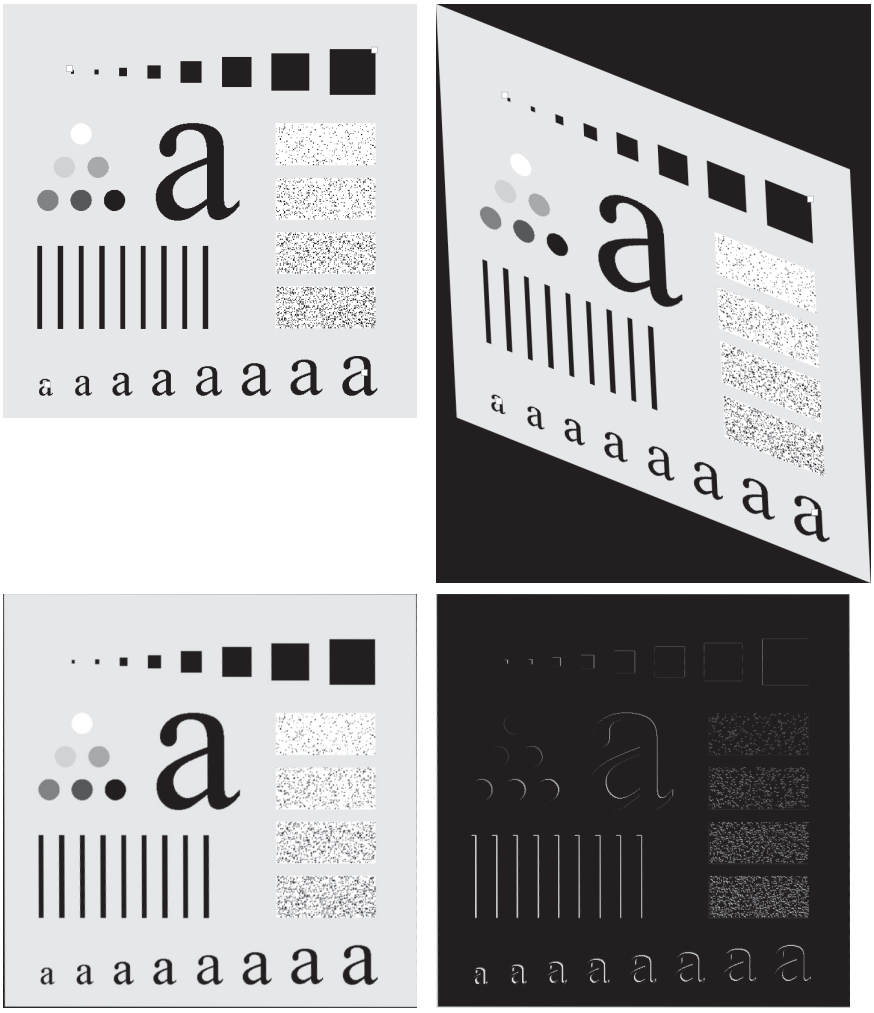


Рис. 2.37. Совмещение изображений. (*а*) Эталонное изображение. (*б*) Исходное (геометрически искаженное) изображение. Соответственные точки привязки указаны маленькими белыми квадратиками вблизи углов. (*в*) Изображение после совмещения (видны дефекты на краях). (*г*) Разность изображений (*а*) и (*в*) показывает другие ошибки совмещения

абзаце. Видно, что достигнутое совмещение не идеально, о чем свидетельствуют черные края на рисунке. Разностное изображение на рис. 2.37(*г*) более ясно демонстрирует легкие несовпадения между эталонным и откорректированным изображениями. Причиной рассогласования является неточность при ручном выборе точек привязки. Трудно достичь идеального совмещения, когда искажения столь значительны. ■

2.6.6. Векторные и матричные операции

Обработка многоканальных изображений является типичной областью использования векторных и матричных операций над изображениями. Например, из главы 6 мы знаем, что цветные цифровые изображения формируются в цветовом пространстве RGB с помощью трех изображений красной, зеленой и синей цветовых компонент, как показано на рис. 2.38. Как мы видим, здесь *каждый* пиксель цветного изображения имеет три компоненты, которые можно записать в форме *вектор-столбца*

$$\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}, \tag{2.6-26}$$

где z_1 есть яркость пикселя на красном изображении, а другие два элемента суть яркости соответственных пикселей на зеленом и синем изображении. Таким образом, цветное RGB-изображение размерами $M \times N$ можно представить тремя изображениями-компонентами тех же размеров или совокупностью MN трехмерных векторов. В общем случае многоканальное изображение, состоящее из n компонент (см., например, рис. 1.10), приводит к n-мерным векторам. Такое векторное представление будет местами использоваться в главах 6, 10, 11 и 12.

В разделе «Обучающие материалы» на сайте книги в Интернете имеется краткое введение в теорию векторов и матриц.

Представив пиксели в форме векторов, мы получаем в распоряжение аппарат линейной алгебры и аналитической геометрии. Например, *евклидово рас-*

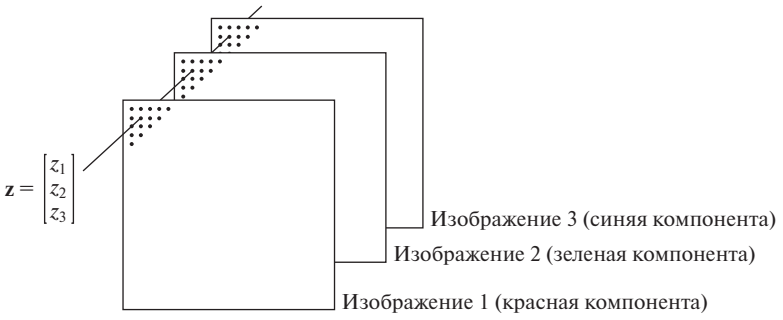


Рис. 2.38. Построение вектора из значений соответственных пикселей трех компонент RGB изображения

стояние D между векторным пикселем **z** и произвольной точкой **a** в *n*-мерном пространстве определяется как скалярное произведение векторов:

$$D(\mathbf{z},\mathbf{a}) = \left[(\mathbf{z}-\mathbf{a})^T(\mathbf{z}-\mathbf{a})\right]^{\frac{1}{2}} = \left[(z_1-a_1)^2+(z_2-a_2)^2+\ldots+(z_n-a_n)^2\right]^{\frac{1}{2}}, \quad (2.6\text{-}27)$$

что является обобщением евклидова расстояния на плоскости, определенного в (2.5-1). Правую часть в (2.6-27) иногда называют *нормой вектора* и обозначают $\|\mathbf{z}-\mathbf{a}\|$. В дальнейшем мы неоднократно будем вычислять это расстояние.

Другое важное преимущество векторного представления пикселей возникает в линейных преобразованиях, которые приобретают простой вид

$$\mathbf{w} = \mathbf{A}(\mathbf{z}-\mathbf{a}), \quad (2.6\text{-}28)$$

где **A** есть матрица порядка $m \times n$, а **z** и **a** суть вектор-столбцы размерности $n \times 1$. Как мы увидим в дальнейшем, преобразования такого типа в ряде случаев полезны при обработке изображений.

Как отмечалось в соотношении (2.4-2), изображение целиком может рассматриваться как матрица (или эквивалентно как вектор), что имеет важные последствия для решения многих задач обработки изображений. Например, можно представить изображение с размерами $M \times N$ в виде вектора размерности $MN \times 1$, используя первую строку изображения в качестве первых N элементов этого вектора, вторую строку — как следующие N элементов и т. д. Если изображение представлено таким способом, то широкий круг линейных преобразований, применяемых к изображению, можно описать в единой форме

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \mathbf{n}, \quad (2.6\text{-}29)$$

где вектор **f** размерности $MN \times 1$ представляет исходное изображение, вектор **n** размерности $MN \times 1$ представляет шумовую составляющую размерами $M \times N$, вектор **g** размерности $MN \times 1$ представляет обработанное изображение, а **H** — матрица порядка $MN \times MN$, которая задает линейное преобразование, применяемое к исходному изображению (см. раздел 2.6.2 о линейных преобразованиях). Например, начав с (2.6-29), можно разработать всю совокупность обобщенных методов восстановления изображений, как мы увидим в разделе 5.9. Мы коснемся темы использования матричного представления в следующем разделе и продемонстрируем другие применения матриц при цифровой обработке изображений в главах 5, 8, 11 и 12.

2.6.7. Преобразования изображений

Все до сих пор обсуждавшиеся методы обработки цифровых изображений оперировали непосредственно пикселями исходного изображения, т. е. работали прямо в *пространственной области*. В некоторых случаях задачи обработки изображений легче формулируются после преобразования исходного изображения, которое переводит задачу в *область преобразования*, а для возвращения обратно в пространственную область используется обратное преобразование. По мере освоения книги мы встретимся с разнообразными преобразованиями. Особенно важный класс двумерных линейных преобразований, которые будем обозначать $T(u, v)$, в общем виде описывается выражением

Рис. 2.39. Общий подход к обработке в области линейных преобразований

$$T(u,v)=\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)r(x,y,u,v), \tag{2.6-30}$$

где $f(x,y)$ — исходное изображение, $r(x,y,u,v)$ называется *ядром прямого преобразования* и выражение (2.6-30) вычисляется для $u=0,1,2,...,M-1$ и $v=0,1,2,...,N-1$. Как и прежде, x и y — пространственные переменные, а M и N — количества строк и столбцов в f. $T(u,v)$ называется *прямым преобразованием* изображения $f(x,y)$, а переменные u и v — *переменными преобразования*. Имея $T(u,v)$, можно восстановить $f(x,y)$ применением к $T(u,v)$ *обратного преобразования*

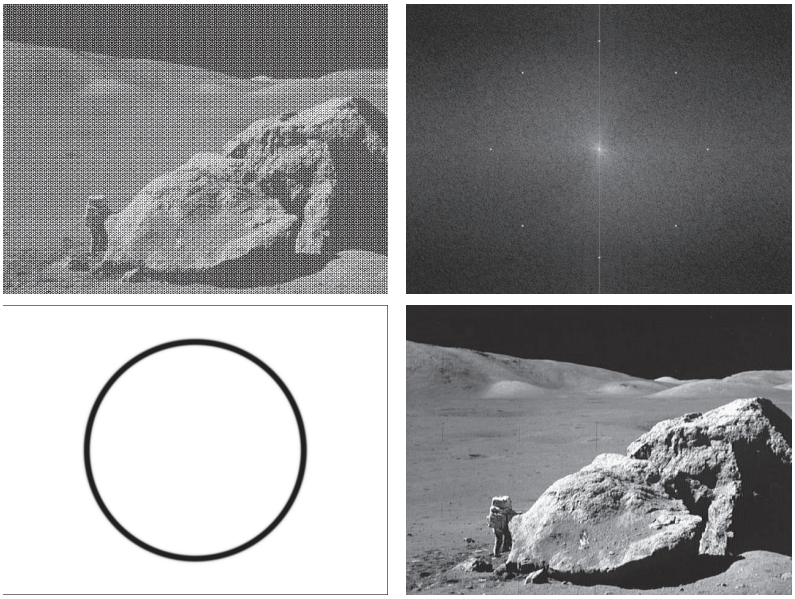
$$f(x,y)=\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}T(u,v)s(x,y,u,v), \tag{2.6-31}$$

для $x=0,1,2,...,M-1$ и $y=0,1,2,...,N-1$, где $s(x,y,u,v)$ называется *ядром обратного преобразования*. Вместе (2.6-30) и (2.6-31) называются *парой преобразований*.

На рис. 2.39 показаны основные шаги обработки изображения в области линейных преобразований. Сначала исходное изображение преобразуется, затем результат преобразования модифицируется с помощью заранее заданной операции, и в конце с помощью обратного преобразования вычисляется результирующее изображение. Таким образом, процесс из пространственной области переходит в область преобразования и затем возвращается обратно в пространственную область.

Пример 2.11. Обработка изображения в пространстве преобразования.

■ Рис. 2.40 демонстрирует пример выполнения шагов, показанных на рис. 2.39. В данном случае в качестве преобразования используется преобразование Фурье, коротко упоминаемое ниже в этом разделе и подробно рассмотренное в главе 4. На рис. 2.40(*а*) приведено изображение, искаженное синусоидальными помехами, а на рис. 2.40(*б*) представлена амплитуда его Фурье-преобразования, что составляет результат первого шага на рис. 2.39. Как мы узнаем в главе 4, синусоидальные помехи в пространственной области проявляются в форме ярких точек в преобразованном пространстве (частотной области). В данном случае эти точки расположены по кругу, как видно из рис. 2.40(*б*). На рис. 2.40(*в*) показано изображение-маска (называемое фильтром), в котором белое соответствует значению 1, а черное 0. В этом примере операция во втором блоке на рис. 2.39 состоит в умножении преобразованного изображения на маску, в результате чего яркие точки, ответственные за помехи, удаляются. Окончательный результат, полученный с помощью вычисления обратного преобразования ранее преобразованного и затем измененного изображения, представлен на рис. 2.40(*г*). Теперь помехи отсутствуют, и хорошо видны детали изображения. Можно даже различить координатные метки (тусклые крестики), используемые для совмещения изображений. ■



а б
в г

Рис. 2.40. (*а*) Изображение, искаженное синусоидальными помехами. (*б*) Амплитуда Фурье-преобразования (спектр) с заметными яркими точками, которые ответственны за помехи. (*в*) Маска для подавления энергии ярких точек. (*г*) Результат вычисления обратного преобразования Фурье от измененного Фурье-образа. (Исходное изображение предоставлено NASA)

Говорят, что ядро прямого преобразования *разделимо* (*сепарабельно*), если

$$r(x,y,u,v) = r_1(x,u)r_2(y,v). \tag{2.6-32}$$

Если, кроме того, $r_1(x,y)$ и $r_2(x,y)$ суть одна и та же функция, так что

$$r(x,y,u,v) = r_1(x,u)r_1(y,v), \tag{2.6-33}$$

то ядро называется *симметричным*. То же самое относится к ядру обратного преобразования, если в двух последних уравнениях заменить r на s.

Рассмотренное в примере 2.11 двумерное преобразование Фурье имеет прямое и обратное ядра соответственно

$$r(x,y,u,v) = e^{-i2\pi(ux/M+vy/N)} \tag{2.6-34}$$

и

$$s(x,y,u,v) = \frac{1}{MN}e^{i2\pi(ux/M+vy/N)}, \tag{2.6-35}$$

где $i = \sqrt{-1}$, так что эти ядра комплексные. Подставляя эти ядра в общий вид преобразований (2.6-30) и (2.6-31), получаем *пару дискретных преобразований Фурье*:

$$T(u,v) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-i2\pi(ux/M+vy/N)} \tag{2.6-36}$$

и $$f(x,y)=\frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}T(u,v)e^{i2\pi(ux/M+vy/N)}. \qquad (2.6\text{-}37)$$

Эти соотношения являются исключительно важными для цифровой обработки изображений, и бóльшую часть главы 4 мы посвятим сначала их выводу из некоторых базовых принципов, а затем использованию для широкого круга прикладных задач.

Нетрудно показать, что ядра преобразований Фурье разделимы и симметричны (задача 2.25) и что двумерное преобразование с разделимым и симметричным ядром можно вычислить с помощью одномерного преобразования (задача 2.26). Если прямое и обратное ядра пары преобразований удовлетворяют указанным двум условиям и $f(x, y)$ — квадратное изображение размерами $M \times M$, тогда (2.6-30) и (2.6-31) можно записать в матричной форме:

$$\mathbf{T} = \mathbf{AFA}, \qquad (2.6\text{-}38)$$

где $\mathbf{F}$ — квадратная матрица порядка M, составленная из элементов $f(x, y)$ (см. соотношение (2.4-2)), матрица $\mathbf{A}$ такого же размера с элементами $a_{ij} = r_1(i, j)$ и результат преобразования $\mathbf{T}$ — также квадратная матрица порядка M, элементы которой суть значения $T(u, v)$ для $u, v = 0, 1, 2,..., M - 1$.

Для вычисления обратного преобразования нужно умножить обе части (2.6-38) справа и слева на матрицу обратного преобразования $\mathbf{B}$:

$$\mathbf{BTB} = \mathbf{BAFAB}. \qquad (2.6\text{-}39)$$

Если $\mathbf{B} = \mathbf{A}^{-1}$, получаем соотношение

$$\mathbf{F} = \mathbf{BTB}, \qquad (2.6\text{-}40)$$

показывающее, что матрица $\mathbf{F}$ (т. е. изображение $f(x, y)$) полностью восстанавливается из результата прямого преобразования). В противном случае восстанавливается приближение

$$\hat{\mathbf{F}} = \mathbf{BAFAB}. \qquad (2.6\text{-}41)$$

Помимо Фурье-преобразования, в форме соотношений (2.6-30)—(2.6-31) или, что то же самое, (2.6-38)—(2.6-40) можно выразить множество других важных преобразований, в том числе преобразования Уолша—Адамара, Хаара, дискретное косинусное и S-преобразование. Мы рассмотрим некоторые из них, а также другие преобразования в последующих главах.

2.6.8. Вероятностные методы

При цифровой обработке изображений многие объекты могут быть описаны с точки зрения вероятностного подхода. Самый простой из них — когда значения яркости трактуются как реализации случайной величины. Например, обозначим через z_i, $i = 0, 1, 2,..., L - 1$, всевозможные значения яркостей пикселей на цифровом изображении размерами $M \times N$. Вероятность $p(z_k)$ встретить уровень яркости z_k в данном изображении оценивается величиной

В разделе «Обучающие материалы» на сайте книги в Интернете дается краткий обзор теории вероятности.

$$p(z_k) = \frac{n_k}{MN}, \tag{2.6-42}$$

где n_k — число пикселей с яркостью z_k в этом изображении, а MN — общее число пикселей. Ясно, что

$$\sum_{k=0}^{L-1} p(z_k) = 1. \tag{2.6-43}$$

Зная $p(z_k)$, можно определить ряд важных характеристик изображения. Например, математическое ожидание (среднее значение) яркости задается выражением

$$m = \sum_{k=0}^{L-1} z_k\, p(z_k). \tag{2.6-44}$$

Аналогично дисперсия яркости равна

$$\sigma^2 = \sum_{k=0}^{L-1} (z_k - m)^2\, p(z_k). \tag{2.6-45}$$

Дисперсия характеризует рассеяние значений z вблизи математического ожидания и поэтому является удобной мерой контраста изображения. В общем случае центральный момент n-го порядка случайной величины z определяется выражением

$$\mu_n(z) = \sum_{k=0}^{L-1} (z_k - m)^n\, p(z_k). \tag{2.6-46}$$

Как видим, $\mu_0(z) = 1$, $\mu_1(z) = 0$ и $\mu_2(z) = \sigma^2$. В то время как математическое ожидание и дисперсия имеют непосредственную и очевидную связь с визуальными свойствами изображения, влияние моментов более высокого порядка намного менее заметно. Например, положительная величина третьего момента показывает, что яркости смещены в бóльшую сторону от среднего значения, при отрицательном третьем моменте картина противоположная, а нулевая величина третьего момента говорит о том, что значения яркости распределены приблизительно поровну от среднего. Эти характеристики могут быть полезны при вычислениях, но они мало говорят о картине изображения в целом.

Размерность дисперсии равна квадрату значений яркости. При сравнении величины контраста вместо дисперсии обычно используют стандартное отклонение σ (квадратный корень из дисперсии), поскольку оно имеет ту же размерность, что и яркость.

Пример 2.12. Величина стандартного отклонения как мера контраста изображения.

■ На рис. 2.41 приведены три 8-битовых изображения соответственно с низким, средним и высоким контрастом. Стандартные отклонения яркости пикселей

а б в

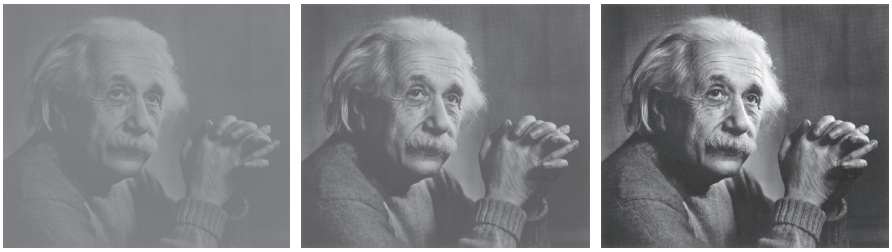


Рис. 2.41. Изображения с (*а*) низким, (*б*) средним и (*в*) высоким контрастом

для этих изображений равны соответственно 14,3, 31,6 и 49,2 уровней яркости, а дисперсии равны соответственно 204,3 , 997,8 и 2424,9. Значения обеих характеристик говорят об одном и том же, но коль скоро диапазон возможных яркостей этих изображений составляет [0, 255], связь величины стандартного отклонения с этим интервалом намного понятнее, чем в случае дисперсии. ■

Как мы увидим дальше в этой книге, вероятностные концепции играют центральную роль в развитии алгоритмов цифровой обработки изображений. Например, в главе 3 мы используем оценку вероятности (2.6-42) при построении алгоритмов градационного преобразования. В главе 5 вероятностная постановка и матричная запись используются при изложении алгоритмов восстановления изображений. В главе 10 вероятностный подход применяется для сегментации изображений, а в главе 11 он используется для описания текстуры. Наконец, в главе 12 оптимальные методы распознавания объектов строятся на вероятностной постановке.

До сих пор рассматривался вероятностный подход с использованием одной случайной величины (яркости), распределенной по одиночному двумерному изображению. Если рассматривать последовательность изображений, то можно интерпретировать время как третью переменную. Для оперирования этой добавочной сложностью необходим инструментарий *стохастических* методов обработки изображений (слово «стохастический» происходит от греческого слова *stohasis* («догадка»), что подразумевает наличие в результатах процесса случайной составляющей). Можно сделать еще шаг и рассматривать *целое* изображение (а не одну его точку) как случайное пространственное событие. Чтобы решать задачи в такой постановке, необходим аппарат *случайных полей*. В разделе 5.8 дается пример полных изображений как случайных событий, но дальнейшее обсуждение стохастических процессов и случайных полей выходит за рамки этой книги. Исходной точкой для изучения этих тем являются ссылки на литературу в конце данной главы.

Заключение

Изложенный в этой главе материал в первую очередь является подготовительным для последующих обсуждений. Рассмотрение зрительной системы человека дает, хотя и краткое, начальное представление о способностях глаза воспринимать изобразительную информацию. Рассмотрение электромагнитного

спектра и света служит фундаментом для понимания происхождения многих изображений, приведенных в книге. Модель изображения, предложенная в разделе 2.3.4, используется далее в главе 4 как основа для метода улучшения изображения, называемого *гомоморфной фильтрацией*.

Принципы дискретизации и интерполяции, представленные в разделе 2.4, объясняют природу многих явлений, возникающих при оцифровке изображений и встречающихся на практике. Мы вернемся к теме дискретизации и сопутствующим вопросам в главе 4 после освоения Фурье-преобразования и частотной области.

Введенные в разделе 2.5 понятия являются фундаментом для построения методов обработки изображений на основе локальной окрестности пикселей. Например, как показано в следующей главе и в главе 5, методы локальной обработки находятся в центре многих процедур улучшения и восстановления изображений. В главе 9 локальные операции используются при морфологической обработке изображений; в главе 10 — для сегментации изображений, а в главе 11 — для описания изображений. Насколько это возможно, в коммерческих системах обработки изображений предпочтение отдается методам обработки в локальной окрестности, прежде всего благодаря высокой скорости выполнения операций и простоты аппаратной реализации.

Материал раздела 2.6 станет хорошим подспорьем на протяжении всей книги. Хотя обсуждение носит сугубо вводный характер, оно помогает лучше представить, что означает тот или иной метод применительно к обработке цифровых изображений. Как отмечалось в этом разделе, введенный в рассмотрение аппарат по мере необходимости расширяется в последующих главах. Вместо того, чтобы посвятить отдельную главу или приложение исчерпывающему изложению математических концепций, мы считаем более осмысленным осваивать необходимые расширения математического аппарата из раздела 2.6 в дальнейших главах в контексте его применения для решения конкретных задач обработки изображений.

Ссылки и литература для дальнейшего изучения

Дополнительный материал к разделу 2.1 по строению человеческого глаза можно найти в книгах [Atchison, Smith, 2000] и [Oyster, 1999], а дальнейшие сведения о зрительном восприятии почерпнуть в монографиях [Regan, 2000] и [Gordon, 1997]. Интерес также представляют книги [Hubel, 1988] и ставшая уже классикой [Cornsweet, 1970]. Книга [Born, Wolf 1999] может служить основным справочным руководством, в котором свет рассматривается с позиций электромагнитной теории. Подробности распространения энергии электромагнитных волн обсуждаются в [Felsen, Marcuvitz, 1994].

Тема регистрации изображений необычайно широка и очень быстро развивается. Прекрасным источником информации об оптических сенсорах и других чувствительных элементах для регистрации изображений являются публикации Общества по оптической технике (SPIE). Примерами типичных изданий SPIE по этой тематике могут служить книги [Blouke et al., 2001], [Hoover, Doty, 1996] и [Freeman, 1987].

Представленная в разделе 2.3.4 модель изображения взята из статьи [Oppenheim, Schafer, Stockham, 1968]. Приведенные в этом разделе значения освещенности и коэффициента отражения взяты из справочника [*IESNA Lighting Handbook*, 2000]. Дополнительный материал по дискретизации изображений и некоторым связанным с ней эффектам можно найти в [Bracewell, 1995]. Эта тема более подробно рассматривается в главе 4. Упомянутые в разделе 2.4.3 ранние эксперименты по изучению зависимости воспринимаемого качества изображения от дискретизации и квантования описаны в [Huang, 1965]. Вопрос об уменьшении частоты дискретизации и числа уровней квантования при минимальном ухудшении качества изображения по-прежнему представляет интерес, как показывает статья [Papamarkos, Atsalakis, 2000]. Дополнительные сведения об уменьшении и увеличении изображений можно найти в работах [Sid-Ahmed, 1995], [Unser et al., 1995], [Umbaugh, 2005] и [Lehmann et al., 1999]. В качестве дополнительного материала по теме раздела 2.5 рекомендуются книги [Rosenfeld, Kak, 1982], [Marchand-Maillet, Sharaiha, 2000] и [Ritter, Wilson, 2001].

Дальнейшие сведения о линейных системах применительно к обработке изображений (раздел 2.6.2) можно найти в монографии [Castleman, 1996]. Метод уменьшения шума с помощью усреднения изображения (раздел 2.6.3) впервые был предложен в работе [Kohler, Howell, 1963]. См. также [Peebles, 1993] об ожидаемых значениях среднего и дисперсии суммы случайных величин. Вычитание изображений (раздел 2.6.3) является типовым методом обработки, который широко применяется для обнаружения изменений. Чтобы имело смысл его использовать, либо изображения должны быть совмещены, либо должны быть идентифицированы все артефакты, обусловленные движением. Две статьи [Meijering et al., 1999, 2000] демонстрируют методы, применяемые для достижения этих целей.

Основным справочником по материалу раздела 2.6.4 является книга [Cameron, 2005], а более глубокое рассмотрение этой темы проводится в книге [Tourlakis, 2003]. Введение в теорию нечетких множеств дается в разделе 3.8 и в относящихся к нему работах, указанных в ссылках к главе 3. Дальнейшие подробности поэлементной и локальной обработки (раздел 2.6.5) обсуждаются в разделах 3.2—3.4, их также можно найти по ссылкам к главе 3. О геометрических пространственных преобразованиях см. книгу [Wolberg, 1990].

Основным справочником по матричным и векторным операциям (раздел 2.6.6) служит книга [Noble, Daniel, 1988]. Подробному рассмотрению преобразования Фурье (раздел 2.6.7) посвящена глава 4, а в главах 7, 8 и 11 рассматриваются примеры других преобразований, применяемых при цифровой обработке изображений. Основы теории вероятности и случайных величин (раздел 2.6.8) изложены в [Peebles, 1993], а углубленное рассмотрение этой темы проводится в книге [Papoulis, 1991]. Основополагающий материал по использованию стохастических процессов и случайных полей в цифровой обработке изображений приведен в работах [Rosenfeld, Kak, 1982], [Jähne, 2002] и [Won, Gray, 2004].

Подробности программной реализации многих методов, обсуждаемых в этой главе, рассматриваются в книге [Gonzalez, Woods, Eddins, 2004].

Задачи

Подробные решения задач, отмеченных звездочкой, приводятся на посвященном книге сайте в Интернете. Там же можно найти темы предлагаемых проектов, основанных на материале данной главы.

★**2.1** Используя материал раздела 2.1 и рассуждая чисто геометрически, оцените наименьший диаметр напечатанной точки, различаемой глазом, если страница рассматривается с расстояния 0,3 м. Для простоты предполагайте, что зрительная система не воспринимает точку, если размеры ее изображения на центральной ямке меньше диаметра одного рецептора (колбочки) в этой области сетчатки. Считайте, что центральная ямка имеет форму круга диаметром 1,5 мм и колбочки равномерно распределены на этой площади с промежутками, равными размеру рецептора.

2.2 Когда Вы входите в темный кинозал с улицы, проходит некоторое время, прежде чем Вы станете видеть достаточно хорошо, чтобы найти свободное место. Какой из объяснявшихся в разделе 2.1 процессов происходит в это время в зрительной системе?

★**2.3** Переменный ток является частью электромагнитного спектра, хотя это явно не отражено на рис. 2.10. Частота промышленного тока в США равна 60 Гц. Какова длина волны в метрах для этой составляющей спектра?

2.4 Перед Вами поставлена задача разработать входную часть системы обработки изображений для исследования формы клеток, бактерий, вирусов и белков. В данном случае эта входная часть должна состоять из одного или нескольких источников освещения и отвечающих им устройств регистрации изображений (камер). Каждая из упомянутых категорий характеризуется размером особей 25 мкм, 0,5 мкм, 0,05 мкм и 0,005 мкм соответственно.

(а) Можно ли в этой задаче обеспечить формирование изображений с помощью одного набора из источника и камеры? Если да, то укажите диапазон длин волн источника освещения и вид необходимого устройства регистрации. Под «видом» понимается диапазон электромагнитного спектра, в котором камера обладает наибольшей чувствительностью (например инфракрасный диапазон).

(б) Если нет, то какие виды источников освещения и устройств регистрации Вы бы порекомендовали? (Указывайте виды источников и камер так же, как и в части (а).) Используйте *минимально необходимое* для решения задачи число источников освещения и устройств регистрации.

«Решить задачу» означает способность обнаруживать объекты круглой формы с диаметром соответственно 25, 0,5 , 0,05 и 0,005 мкм.

2.5 ПЗС-камера оснащена матрицей размерами 14×14 мм, состоящей из 2048×2048 элементов. На эту матрицу проецируется изображение плоской квадратной области, находящейся на расстоянии 0,5 м. Сколько пар линий на миллиметр в указанной области способна различить такая камера? Камера оборудована объективом с фокусным расстоянием 35 мм.

(*Подсказка*: используйте геометрическую модель формирования изображения, аналогичную изображенной на Рис 2.3, в которой вместо фокусного расстояния глаза участвовало бы фокусное расстояние объектива).

★**2.6** Для автозавода разрабатывается автоматизированная система, управляющая размещением декоративных элементов на бамперах ограниченной серии спортивных автомобилей. Эти элементы должны сочетаться по цвету с автомобилем, поэтому сборочный робот должен знать цвет конкретного автомобиля для правильного выбора декоративных элементов. Модели выпускаются только четырех цветов: синего, зеленого, красного и белого. Вам поручено предложить решение на основе системы технического зрения. Как бы Вы предложили решить проблему автоматического определения цвета автомобиля, считая *стоимость* наиболее важным соображением при выборе компонентов такой системы?

2.7 Предположим, что плоская область с центром в точке (x_0, y_0) освещена источником света, обеспечивающим распределение освещенности по следующему закону:

$$i(x, y) = Ke^{-\left[(x-x_0)^2+(y-y_0)^2\right]}.$$

Примем для простоты, что коэффициент отражения поверхности постоянен и равен 1, а $K = 255$. Предполагая, что получаемое изображение квантуется с яркостным разрешением k бит, а глаз способен ощущать границу лишь при разности величины яркости соседних пикселей в 4 градации и выше, определите, при каком значении k станут заметны ложные контуры.

2.8 Нарисуйте вид изображения в задаче 2.7 при $k = 4$.

★**2.9** Общепринятой единицей измерения скорости передачи цифровых данных является *бод*; по определению 1 бод соответствует скорости передачи 1 бит/с. В общем случае информация передается пакетами, состоящими из стартового бита, байта информации (8 бит) и стопового бита. Опираясь на эти факты, ответьте:

(а) Сколько минут займет передача изображения размерами 2048×2048 элементов с 256 градациями яркости с помощью модема, обеспечивающего скорость передачи 33,6 Кбод (1 Кбод = 1000 бод)?

(б) Каково будет время при скорости 3000 Кбод, типичной для соединения по телефонной линии в режиме DSL (цифровая абонентская линия)?

2.10 В телевидении высокой четкости (ТВЧ) изображения с вертикальным разрешением 1080 строк генерируются в режиме чересстрочной развертки, т. е. когда на экран с интервалом 1/60 с по очереди выводятся два поля (полукадра), состоящие соответственно из четных и нечетных строк каждого кадра. Отношение ширины изображения к высоте равно 16:9. Тот факт, что изображение формируется из заданного числа отдельных горизонтальных строк, определяет разрешение по вертикали. Предположим, что создано устройство для записи программ ТВЧ в виде последовательности цифровых изображений каждого кадра, и в этом устройстве разрешение по горизонтали (вдоль телевизионной строки) пропорционально вертикальному разрешению (в соответствии с отношением сторон экрана). Каждый пиксель цветного изображения представляется 24 бит яркости — по 8 бит для красного, зеленого и синего изображений,

в совокупности образующих цветное изображение. Сколько бит займет запись 90-минутного фильма ТВЧ с помощью такого устройства?

★**2.11** Рассмотрим на изображении два подмножества S_1 и S_2, показанные на рисунке. Считая V={1}, определите, являются ли эти два подмножества (а) 4-смежными, (б) 8-смежными или (в) m-смежными.

	S_1				S_2				
0	0	0	0	0	0	0	1	1	0
1	0	0	1	0	0	1	0	0	1
1	0	0	1	0	1	1	0	0	0
0	0	1	1	1	0	0	0	0	0
0	0	1	1	1	0	0	1	1	1

★**2.12** Составьте алгоритм для преобразования 8-пути, имеющего толщину в один пиксель, в 4-путь.

2.13 Составьте алгоритм для преобразования m-пути, имеющего толщину в один пиксель, в 4-путь.

2.14 Вернемся к рассмотрению в конце раздела 2.5.2, где мы определили фон как дополнение $(R_u)^c$ к объединению всех областей на изображении. В некоторых прикладных задачах лучше определять фон как подмножество пикселей, не включающее «окон» внутри областей (говоря нестрого, под окном понимается множество фоновых пикселей, окруженное пикселями области переднего плана). Как бы Вы изменили определение, чтобы исключить из $(R_u)^c$ пиксели окон? Ответ в духе «фоном называется подмножество пикселей $(R_u)^c$, которые не принадлежат окнам» не принимается. (*Подсказка:* используйте понятие связности.)

2.15 Для показанного ниже участка изображения:

★ **(а)** Считая V={0, 1, 2}, вычислите длины кратчайших 4-, 8-, и m-путей между элементами p и q. Если какой-то из путей между этими двумя точками не существует, то объясните почему.

(б) Повторите то же для V={2, 3, 4}.

3	4	1	2	0
0	1	0	4	2(q)
2	2	3	1	4
(p)3	0	4	2	1
1	2	0	3	4

2.16 ★ **(а)** Сформулируйте условия, при которых расстояние D_4 между двумя точками p и q равно длине кратчайшего 4-пути между этими точками.

(б) Является ли этот путь единственным?

2.17 Решите задачу 2.16 для расстояния D_8.

★**2.18** В следующей главе мы познакомимся с операторами, функция которых состоит в вычислении суммы значений пикселей в пределах малой подобласти изображения S. Докажите, что такие операторы являются линейными.

2.19 Медианой ζ множества чисел называется такой его элемент, который больше или равен половине элементов этого множества и меньше или равен другой половине элементов. Например, медианой множества чисел {2, 3, 8, 20, 21, 25, 31} является число 20. Докажите, что оператор, вычисляющий медиану значений пикселей в пределах малой подобласти изображения S, является нелинейным.

★**2.20** Докажите справедливость соотношений (2.6-6) и (2.6-7). (*Подсказка:* начните с (2.6-4) и воспользуйтесь тем, что математическое ожидание суммы равно сумме математических ожиданий.)

2.21 Пусть имеются два 8-битовых изображения, у которых значения яркости занимают весь диапазон [0, 255].

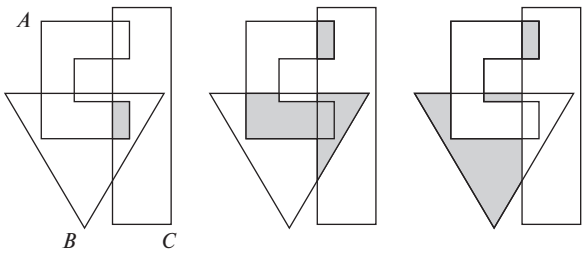
(а) Рассмотрите эффект ограничения, возникающий в результате неоднократного вычитания второго изображения из первого. Считайте, что результат представляется также 8 битами.

(б) Изменится ли результат операции при перемене изображений местами?

★**2.22** Вычитание изображений часто используется в промышленных приложениях для обнаружения отсутствующих сборочных компонентов. Метод опирается на использовании хранящегося эталонного изображения правильно собранного изделия; это изображение вычитается из изображения каждого поступающего готового изделия. В идеале разность должна быть нулевой, если предъявленное изделие собрано правильно. Для изделий с отсутствующими компонентами разностное изображение отлично от нуля в тех областях, где оно отличается от эталонного изображения. Сформулируйте условия, которые должны выполняться на практике, чтобы изложенный метод был работоспособен.

2.23 ★ **(а)** Для изображения на рис. 2.31 изобразите множество $(A^c \setminus B) \cup (B \setminus A)$.

(б) Запишите выражениями относительно множеств A, B и C отмеченные темным цветом множества на следующем рисунке. Темные области на каждом рисунке образуют одно множество, так что для каждого из трех рисунков необходимо дать одно выражение.



2.24 Каковы будут аналоги уравнений (2.6-24) и (2.6-25) при использовании областей треугольной формы вместо четырехугольников?

2.25 Докажите, что ядра преобразований Фурье (2.6-34) и (2.6-35) являются разделимыми и симметричными. Какие преимущества дает использование разделимых преобразований изображений?

★**2.26** Докажите, что двумерное преобразование с разделимым симметричным ядром можно вычислить за два шага: (1) вычислить одномерные преобразования отдельно вдоль каждой строки (столбца) исходного изображения, затем (2) вычислить одномерные преобразования вдоль столбцов (строк) изображения, полученного на шаге (1).

2.27 Завод изготовляет миниатюрные квадратики из полупрозрачного полимера. Жесткие требования качества вынуждают проводить стопроцентный визуальный контроль, требующий больших затрат. Проверка осуществляется в полуавтоматическом режиме: на каждом посту контроля роботизированный механизм помещает каждый квадратик между источником света и оптической системой, проецирующей увеличенное изображение квадратика на экран размерами 80×80 мм, занимая всю его площадь. Дефекты выглядят как круглые темные пятна, и в функции контролера входит, используя нанесенную на экран шкалу, вести отбраковку образцов, в которых встречается хотя бы одно такое темное пятно диаметром 0,8 мм и более. Менеджер убежден, что если удастся найти способ полностью автоматизировать процесс контроля, это приведет к увеличению прибыли на 50% и поможет ему подняться по корпоративной лестнице. После изучения вопроса он решает оснастить каждый пост контроля ПЗС-телекамерой, подключенной к промышленной системе обработки изображений, которая бы обнаруживала пятна, измеряла их диаметр и активировала механизм отбраковки, как это раньше делал контролер. Менеджеру удается найти на рынке систему, способную выполнить все эти действия, если дефект минимального размера будет занимать на цифровом изображении область не менее чем 2×2 пикселя. Вас нанимают в качестве консультанта, чтобы помочь правильно выбрать тип камеры и объектива среди серийно выпускаемых изделий. Доступны объективы с фокусным расстоянием от 25 мм до 200 мм (с шагом 25 мм) и ПЗС-камеры с разрешением 512×512, 1024×1024 или 2048×2048 пикселей, у которых *каждый* сенсор имеет форму квадрата размером 8×8 мкм, а расстояние между соседними сенсорами равно 2 мкм. В этой прикладной задаче камеры обходятся намного дороже объективов, поэтому следует выбрать камеру минимально необходимого разрешения, используя соответствующий объектив. Сформулируйте в письменном виде свои рекомендации, обосновав их с помощью достаточно подробного анализа. Используйте ту же геометрическую модель формирования изображения, что и в задаче 2.5.

ГЛАВА 3

ЯРКОСТНЫЕ ПРЕОБРАЗОВАНИЯ И ПРОСТРАНСТВЕННАЯ ФИЛЬТРАЦИЯ

Вся разница в том, что именно мы хотим увидеть: темное на светлом или светлое на темном.

Дэвид Линдсей

Введение

Термин *пространственная область* относится к плоскости изображения как таковой, и методы обработки изображений, относящиеся к данной категории, основаны на прямых операциях с пикселями изображения. Это составляет противоположность обработке изображений в *трансформационной области*, которая, как введено в разделе 2.6.7 и более подробно обсуждается в главе 4, означает первоначально преобразование изображения в трансформационную область, выполнение обработки в ней и осуществление обратного преобразования для перевода результата снова в пространственную область. Двумя главными классами пространственной обработки являются яркостные преобразования и пространственная фильтрация. Как будет показано в этой главе, яркостные преобразования оперируют отдельными пикселями изображения, главным образом с целью управления контрастом и пороговыми операциями над изображением. Пространственная фильтрация осуществляет выполнение таких операций, как повышение резкости изображения, оперируя над окрестностью каждой точки изображения. В последующих разделах будет рассмотрен ряд «классических» методов яркостного преобразования и пространственной фильтрации. Также будут обсуждены некоторые детали нечетких методов, которые позволяют объединить неточную, взятую из базы знаний информацию в формулировке алгоритмов яркостных преобразований и пространственной фильтрации.

3.1. Предпосылки

3.1.1. Основы яркостных преобразований и пространственной фильтрации

Все методы обработки изображений, обсуждаемые в настоящем разделе, реализованы в *пространственной области*, которая, как мы знаем из раздела 2.4.2, есть просто плоскость, содержащая пиксели изображения. Как отмечено в раз-

деле 2.6.7, пространственные методы оперируют непосредственно пикселями изображения, в противоположность, например, частотным методам (что является предметом обсуждения главы 4), в которых операции выполняются над результатами Фурье-преобразования изображения, а не над самим изображением. Как вы узнаете из дальнейшего прочтения книги, одни задачи обработки изображений легче интерпретируются при реализации в пространственной области, тогда как для других лучше годятся иные подходы. Как правило, пространственные методы в вычислительном отношении являются более эффективными и требуют меньших вычислительных ресурсов при реализации.

Процессы пространственной обработки, рассматриваемые в настоящей главе, описываются уравнением

$$g(x,y) = T[f(x,y)], \tag{3.1-1}$$

где $f(x,y)$ — входное изображение, $g(x,y)$ — обработанное изображение, а T — оператор над f, определенный в некоторой окрестности точки (x,y). Оператор T может быть применим к одному изображению (именно этому уделено основное внимание в данной главе) или же к набору изображений, например для выполнения поэлементного суммирования последовательности изображений с целью подавления шума, как рассматривалось в разделе 2.6.3. На рис. 3.1 показан основной вариант реализации уравнения (3.1-1) при обработке одного изображения. Отмеченная точка (x,y) может находиться в любом месте изображения, а окружающая ее небольшая отмеченная область является окрестностью точки (x,y), как было указано в разделе 2.6.5. Обычно окрестность является прямоугольной, центрированной относительно (x,y) и значительно меньшей в размерах, чем само изображение.

> Иногда используются окрестности другой формы, такие как дискретное приближение к кругу, однако почти всегда превалирует прямоугольная форма, потому что она значительно проще при реализации вычислений.

Рис. 3.1 иллюстрирует процесс обработки, при котором осуществляется переход начала координат окрестности от точки к точке и применения оператора T к пикселям окрестности для получения выходного значения в текущей точке. Таким образом, для любой выбранной точки (x,y) исходного изображения f выходное значение g вычисляется в результате применения T к элементам окрестности точки (x,y). Предположим, например, что окрестность является квадратом размерами 3×3 элементов и оператор T определен как «вычислить среднюю яркость по окрестности». Рассмотрим произвольную точку изображения, например имеющую координаты (100, 150). Полагая, что начало координат окрестности расположено в ее центре, результат $g(100, 150)$ вычисляется как сумма значений самой точки $f(100, 150)$ и 8 ее соседей, деленная на 9 (т. е. средняя яркость точек, попадающих в окрестность). Затем начало координат окрестности передвигается на соседнюю точку и процедура повторяется для получения значения следующего элемента выходного изображения g. Как правило, обработка начинается с верхнего левого угла исходного изображения и продолжается последовательно для всех пикселей строки по горизонтали, по одной строке за проход. Когда начало координат окрестности попадает на границу изображения, часть

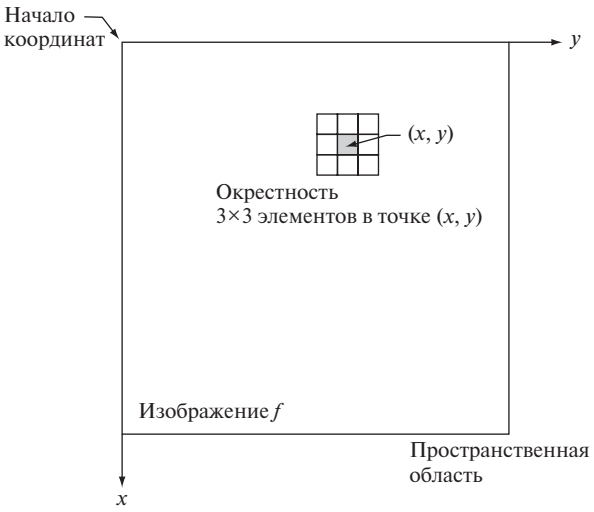


Рис. 3.1. Окрестность 3×3 вокруг точки (x,y) изображения в пространственной области. При формировании выходного изображения окрестность передвигается от точки к точке изображения

окрестности оказывается вне изображения. В таком случае при выполнении вычислений, заданных оператором T, приходится либо игнорировать элементы, выходящие за границы изображения, либо присваивать им нулевые или какие-то другие значения. Ширина такой полосы расширения изображения зависит от размера окрестности. Мы еще вернемся к этому вопросу в разделе 3.4.1.

Только что описанный процесс обработки подробно рассмотрен в разделе 3.4 и называется *пространственной фильтрацией*, в которой окрестность вместе с заданной операцией называется *пространственным фильтром* (также используются названия *маска*, *ядро* или *окно*)[1].

Наименьшая возможная окрестность имеет размеры 1×1. В этом случае g зависит только от значения f в точке (x,y), и T в уравнении (3.1-2) становится *функцией градационного преобразования*, также называемой *функцией преобразования яркости* или *функцией отображения*, имеющей вид

$$s = T(r), \tag{3.1-2}$$

где для простоты обозначения r и s суть переменные, обозначающие соответственно значения яркостей изображений f и g в каждой точке (x,y). Например, если $T(r)$ имеет вид, показанный на рис. 3.2(*a*), то эффект от применения этого преобразования к каждому пикселю изображения f для получения соответствующего пикселя изображения g выразится в формировании изображения более высокого контраста по сравнению с оригиналом посредством затемнения пикселей со значениями, меньшими k, и повышении яркостей пикселей со значениями большими k. В этом методе, иногда называемом *усилением контраста* (см. раздел 3.2.4), значения r, меньшие k, при приближении к уровню черного сжимаются с помощью функции преобразования во все более узкий диапазон s.

[1] Понятие «*окно*» обычно означает лишь пространственную совокупность элементов исходного изображения без учета оператора T. — *Прим. перев.*

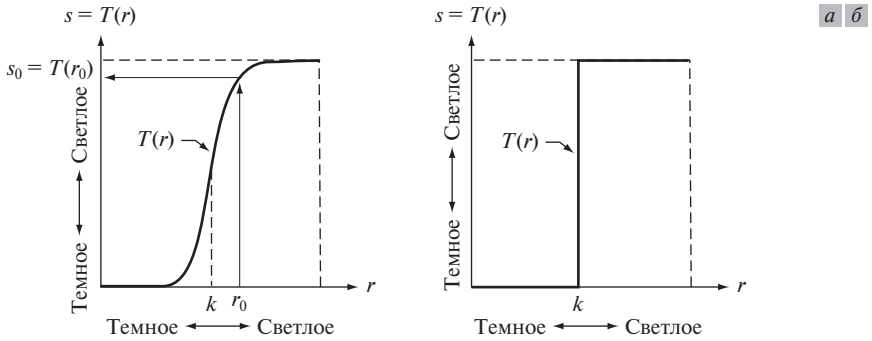


Рис. 3.2. Функция градационного преобразования. (*а*) Функция повышения контраста. (*б*) Пороговая функция

Обратный эффект имеет место для значений r, больших k. Обратим внимание, как значение яркости r_0 отображается в соответствующее значение s_0. В предельном случае, показанном на рис. 3.2(*б*), $T(r)$ дает в результате двухградационное (бинарное) изображение. Отображение такой формы называют *пороговой* функцией. С помощью градационных преобразований могут быть построены некоторые довольно простые, но действенные методы обработки изображений. В данной главе мы используем яркостные преобразования главным образом для улучшения изображений. В главе 10 они будут использованы для сегментации изображений. Подходы, в которых результирующее значение элемента зависит лишь от яркости соответствующего исходного элемента, называются методами *поэлементной обработки* в противоположность методам обработки по окрестности, обсуждаемым далее в данном разделе.

3.1.2. О примерах, приводимых в данной главе

Хотя яркостные преобразования и пространственная фильтрация охватывают весьма широкий диапазон приложений, большинство примеров в данной главе связаны с их применениями в области улучшения изображений. *Улучшение* означает такую процедуру преобразования изображения, что результат оказывается более применимым для конкретного приложения, чем оригинал. Слово *конкретный* является здесь важным, поскольку с самого начала постулирует, что методы улучшения являются проблемно-ориентированными. Так, например, метод, весьма полезный для улучшения рентгеновских изображений, может не оказаться наилучшим для улучшения изображений со спутника, полученных в инфракрасном участке спектра. Не существует общей «теории» улучшения изображений. Когда изображение обрабатывается с целью визуальной интерпретации, то конечным судьей, решающим, насколько хорошо работает тот или иной метод, является человек. Дать количественную оценку конкретной процедуре обработки проще в случае, когда мы имеем дело с машинным восприятием. Например, в системе автоматического распознавания символов наиболее подходящим методом улучшения является тот, который приводит к наилучшему уровню распознавания, оставляя в стороне другие критерии, такие как вычислительные требования того или иного метода.

Независимо от приложений или методов, улучшение изображений является одной из наиболее привлекательных в визуальном отношении областей обработки изображений. Как правило, новички в области обработки изображений находят приложения, в которых используется улучшение, интересными и сравнительно простыми для понимания. Таким образом, использование примеров из области улучшения изображений для иллюстрации методов пространственной обработки, рассматриваемых в настоящей главе, позволяет не только устранить в книге дополнительную главу, касающуюся улучшения изображений, но, что более важно, является эффективным способом ознакомления новичков с деталями методов обработки изображений в пространственной области. Как вы увидите при дальнейшем прочтении книги, базовый материал, рассматриваемый в данной главе, применим в значительно более широких рамках, чем просто улучшение изображений.

3.2. Некоторые основные градационные преобразования

Яркостные преобразования относятся к числу простейших из всех методов улучшения изображений. Значения пикселей до и после обработки будут обозначаться символами r и s соответственно. Как указано в предыдущем разделе, эти величины связаны выражением вида $s = T(r)$, где T является преобразованием, отображающим значение пикселя r в значение пикселя s. Поскольку мы имеем дело с дискретным (квантованным) представлением, значения функции преобразования, как правило, хранятся в одномерном массиве, и отображение из r в s осуществляется по таблице. В случае 8-битового представления таблица преобразования, содержащая значения T, будет состоять из 256 элементов.

В качестве введения в яркостные преобразования рассмотрим рис. 3.3, на котором показаны три основных типа преобразований, часто используемых для улучшения изображений: линейное (негатив и тождественное преобразование), логарифмическое (логарифм и обратный логарифм) и степенное (n-я степень и корень n-й степени). Тождественное преобразование является тривиальным случаем, при котором яркости на выходе идентичны яркостям на входе. Оно приведено на графике только для полноты рассмотрения.

3.2.1. Преобразование изображения в негатив

Преобразование изображения в негатив с яркостями в диапазоне $[0, L - 1]$ осуществляется с использованием негативного преобразования, показанного на рис. 3.3 и определяемого выражением

$$s = L - 1 - r. \tag{3.2-1}$$

Подобный переворот уровней яркости изображения создает эквивалент фотографического негатива. Этот тип обработки особенно подходит для усиления белых или серых деталей на фоне темных областей изображения, особенно когда темные области имеют преобладающие размеры. Пример показан на рис. 3.4.

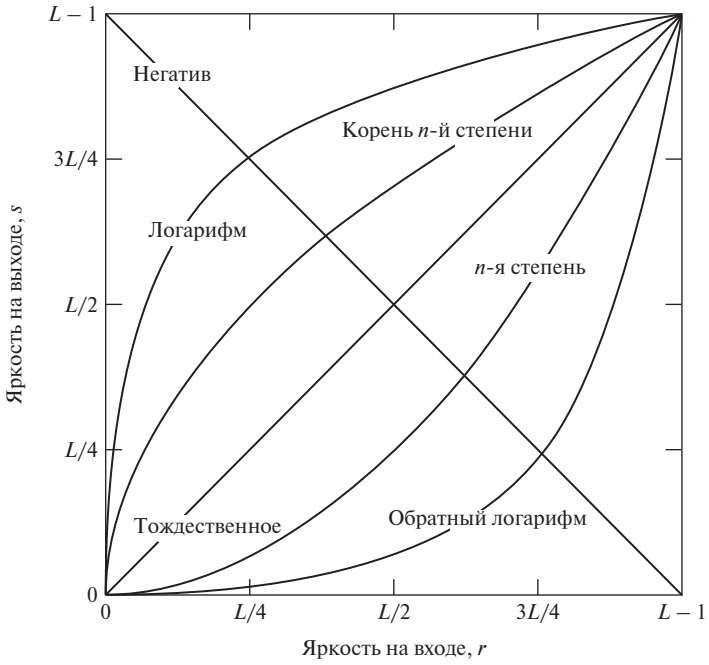


Рис. 3.3. Некоторые основные функции яркостных преобразований. Все кривые масштабированы, чтобы полностью использовать показанный диапазон

На исходном изображении представлена цифровая маммограмма (рентгенограмма молочной железы), демонстрирующая небольшое поражение. Несмотря на тот факт, что визуальное содержание на обоих изображениях является одним и тем же, заметим, насколько проще в данном случае анализировать молочную железу на негативном изображении.

3.2.2. Логарифмическое преобразование

Общий вид логарифмического преобразования, показанного на рис. 3.3, выражается формулой

$$s = c\lg(1+r), \tag{3.2-2}$$

где c — константа и предполагается, что $r \geq 0$. Форма логарифмической кривой на рис. 3.3 показывает, что данное преобразование отображает узкий диапазон малых значений яркостей на исходном изображении в более широкий диапазон выходных значений. Для больших значений входного сигнала верно противоположное утверждение. Мы используем этот тип преобразования для растяжения диапазона значений темных пикселей на изображении с одновременным сжатием диапазона значений ярких пикселей. Наоборот, при использовании обратного логарифмического преобразования происходит растяжение диапазона ярких пикселей и сжатие диапазона темных пикселей.

Любая кривая, имеющая общий вид, близкий к показанной на рис. 3.3 логарифмической функции, будет осуществлять такое растяжение/сжатие диа-

а *б*

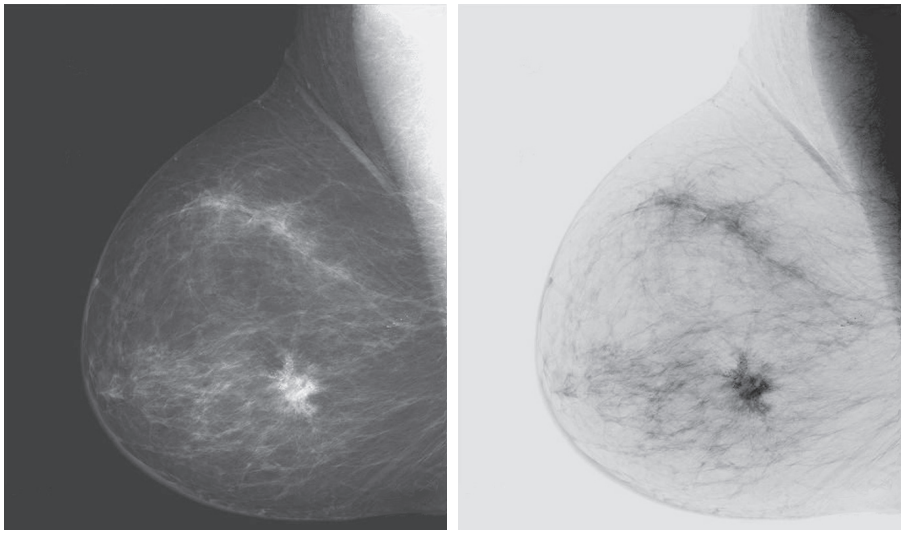
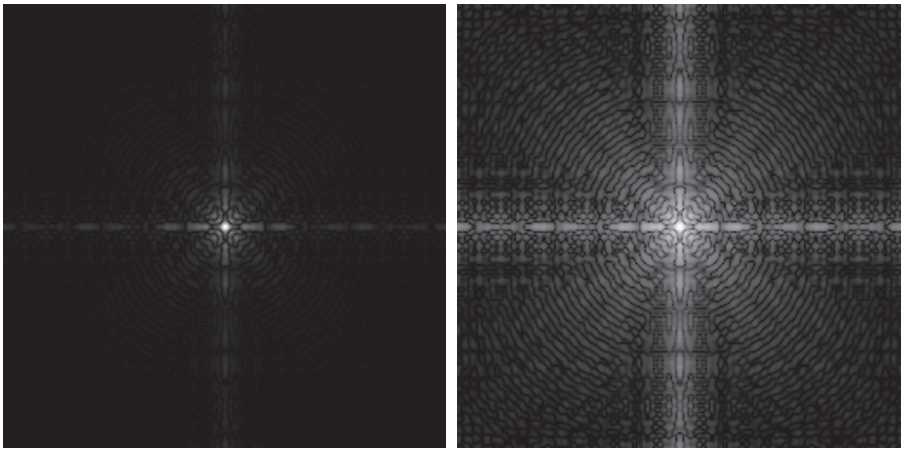


Рис. 3.4. (*а*) Исходный вид цифровой маммограммы. (*б*) Негативное изображение, полученное применением негативного преобразования по формуле (3.2-1). (Предоставлено компанией G. E. Medical Systems)

пазонов яркости на изображении; однако для этих целей значительно более универсальными, чем логарифмические, оказываются степенные преобразования, обсуждаемые в следующем разделе. Логарифмическая функция имеет важную особенность — она позволяет сжимать динамический диапазон изображений, имеющих большие вариации в значениях пикселей. Классическим примером, в котором значения пикселей имеют большой динамический диапазон, является спектр Фурье, обсуждаемый в главе 4. В данный момент нас интересуют лишь свойства спектра как изображения. Спектр, значения которого изменяются в диапазоне от 0 до 10^6 или более, не является чем-то необычным. Если обработка подобной совокупности значений не представляет проблемы для компьютера, то система воспроизведения изображений обычно неспособна правильно отобразить столь большой диапазон значений интенсивности. Результирующий эффект таков, что при обычном воспроизведении спектра Фурье значительное число ярких деталей может быть потеряно.

В качестве иллюстрации логарифмического преобразования на рис. 3.5(*а*) приведено изображение спектра Фурье, имеющего значения в диапазоне от 0 до $1{,}5 \cdot 10^6$. Если масштабировать эти значения линейно для отображения в 8-битовой системе воспроизведения, то наиболее яркие пиксели будут доминировать над слабыми (и зачастую важными) значениями спектра. Эффект такого доминирования ярко иллюстрирует рис. 3.5(*а*), на котором только весьма малая область изображения не воспринимается как черная. Если же вместо подобного способа воспроизведения мы сначала применим к значениям спектра преобразование по формуле (3.2-2) (с $c = 1$ в данном случае), тогда диапазон значений результата будет от 0 до 6,2, что намного удобнее. На рис. 3.5(*б*) показан результат линейного масштабирования нового диапазона и отображения спектра на том же самом 8-битовом устройстве воспроизведения. Из этих иллюстраций становится очевидным богатство видимых деталей на втором изображении



а б

Рис. 3.5. (*а*) Спектр Фурье. (*б*) Результат применения логарифмического преобразования по формуле (3.2-2) с $c = 1$

по сравнению с немодифицированным воспроизведением. Большинство спектров Фурье, демонстрируемых в публикациях по обработке изображений, масштабируются именно таким способом.

3.2.3. Степенные преобразования (гамма-коррекция)

Степенные преобразования имеют вид

$$s = cr^{\gamma}, \tag{3.2-3}$$

где c и γ являются положительными константами. Иногда уравнение (3.2-3) записывается в виде $s = c(r + \varepsilon)^{\gamma}$ для того, чтобы ввести смещение, т. е. измеримый (ненулевой) выход, когда на входе ноль. Впрочем, смещения возникают при калибровке устройстве воспроизведения, и поэтому в уравнении (3.2-3) они обычно игнорируются. Графики зависимостей s от r при различных значениях γ показаны на рис. 3.6. Так же как в случае логарифмического преобразования, кривые степенных зависимостей при малых γ отображают узкий диапазон малых входных значений в широкий диапазон выходных значений, при этом для больших входных значений верно обратное утверждение. Однако в отличие от логарифмических функций здесь возникает целое семейство кривых возможного преобразования, получаемых простым изменением параметра γ. Как и следовало ожидать, на рис. 3.6 видно, что кривые, полученные со значениями $\gamma > 1$ дают прямо противоположный эффект по сравнению с теми, которые получены при $\gamma < 1$. Наконец отметим, что уравнение (3.2-3) приводится к тождественному преобразованию при $c = \gamma = 1$.

Амплитудная характеристика многих устройств, используемых для ввода, печати или визуализации изображений, соответствует степенному закону. По традиции показатель степени в уравнении степенного преобразования называют *гамма*, и именно поэтому символ γ использован в уравнении (3.2-3). Процедура, используемая для коррекции таких степенных характеристик, называется *гамма-коррекцией*. Например, устройства с электронно-лучевой трубкой (ЭЛТ)

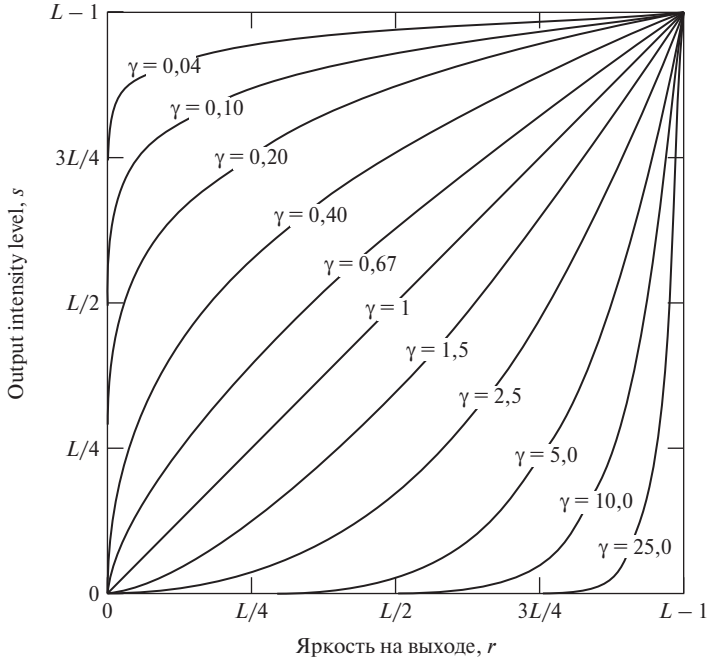


Рис. 3.6. Графики уравнения $s = cr^{\gamma}$, для различных значений γ ($c = 1$ во всех случаях). Все кривые были масштабированы, чтобы полностью использовать показанный диапазон

имеют степенную зависимость яркости от напряжения с показателем степени в диапазоне от 1,8 до 2,5. Обращая внимание на кривую для $\gamma = 2{,}5$ на рис. 3.6, можно видеть, что подобная система отображения будет иметь тенденцию к воспроизведению изображений темнее, чем они есть на самом деле. Этот эффект иллюстрируется на рис. 3.7. На рис. 3.7(*а*) показан простой полутоновой линейный клин, подающийся на вход монитора. Как и ожидалось, изображение на экране реального монитора оказывается темнее, чем должно быть на экране идеального монитора, что и видно на рис. 3.7(*б*). Необходимость применения гамма-коррекции очевидна. Все, что требуется для компенсации, — это произвести предобработку визуализируемого изображения с помощью преобразования $s = r^{1/2{,}5} = r^{0{,}4}$ прежде, чем оно поступит на вход монитора. Результат показан на рис. 3.7(*в*). При воспроизведении на том же мониторе такая гамма-коррекция обеспечивает вывод, визуально близкий к оригинальному изображению, как и показано на рис. 3.7(*г*). Аналогичные исследования должны быть применены и по отношению к другим устройствам для работы с изображениями, таким как сканеры и принтеры. Единственным различием между ними должно быть значение гамма, зависящее от конкретного устройства [Poynton, 1996].

Гамма-коррекция необходима, если требуется точное воспроизведение изображения на экране компьютера. Изображения, которые не откорректированы правильно, могут выглядеть или как выбеленные, или, что более вероятно, как слишком темные. Правильное воспроизведение цветов также требует некоторых знаний о гамма-коррекции, поскольку подобное преобразование меняет не только яркость, но также соотношения между красным, зеленым и синим

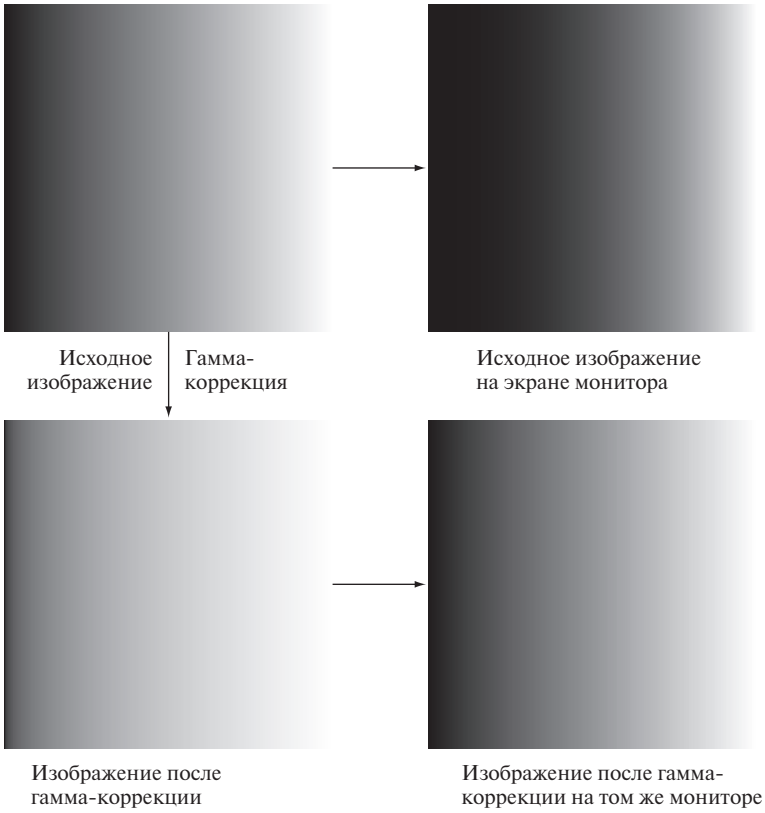


Рис. 3.7. (*а*) Полутоновое изображение с линейным клином. (*б*) Отклик монитора на линейный клин. (*в*) Клин, подвергнутый гамма-коррекции. (*г*) Результат на экране монитора

на цветном изображении. В последние годы гамма-коррекция становится более важной, поскольку увеличивается коммерческое использование цифровых изображений в Интернете. Зачастую изображения, размещенные на популярных сайтах в Интернете, рассматриваются миллионами людей, большинство из которых имеет различные мониторы или их настройки. Некоторые компьютерные системы даже включают в себя встроенную частичную гамма-коррекцию. К тому же используемые в настоящее время стандарты изображений не содержат исходного значения гамма, с которым изображение формировалось, усложняя тем самым получение правильного результата. При подобных ограничениях разумным подходом при хранении изображений на сайте в Интернете является их предобработка со значением гамма, отражающим «средние» параметры мониторов и компьютерных систем.

Пример 3.1. Улучшение контрастов с помощью степенных преобразований.

■ В дополнение к возможностям гамма-коррекции степенные преобразования полезны для универсального управления контрастом. На рис. 3.8(*а*) показан снимок, полученный с помощью ЯМР-томографа (основанного на эффекте

ядерного магнитного резонанса). На нем изображена грудная часть позвоночника человека, имеющего перелом со смещением и поражением позвоночного столба. Перелом виден вблизи центра позвоночника на расстоянии одной четверти от верхнего края снимка. Поскольку изображение преимущественно темное, желательно осуществить растяжение уровней яркости. Это может быть достигнуто с помощью степенного преобразования с дробным (меньшим единицы) показателем степени. Оставшиеся три изображения получены путем обработки изображения рис. 3.8(*а*) степенным преобразованием по формуле (3.2-3). Значения гамма для изображений (*б*), (*в*) и (*г*) равны соответственно 0,6, 0,4 и 0,3 (значение *c* равно 1 во всех случаях). Можно заметить, что с уменьше-

а *б*
в *г*

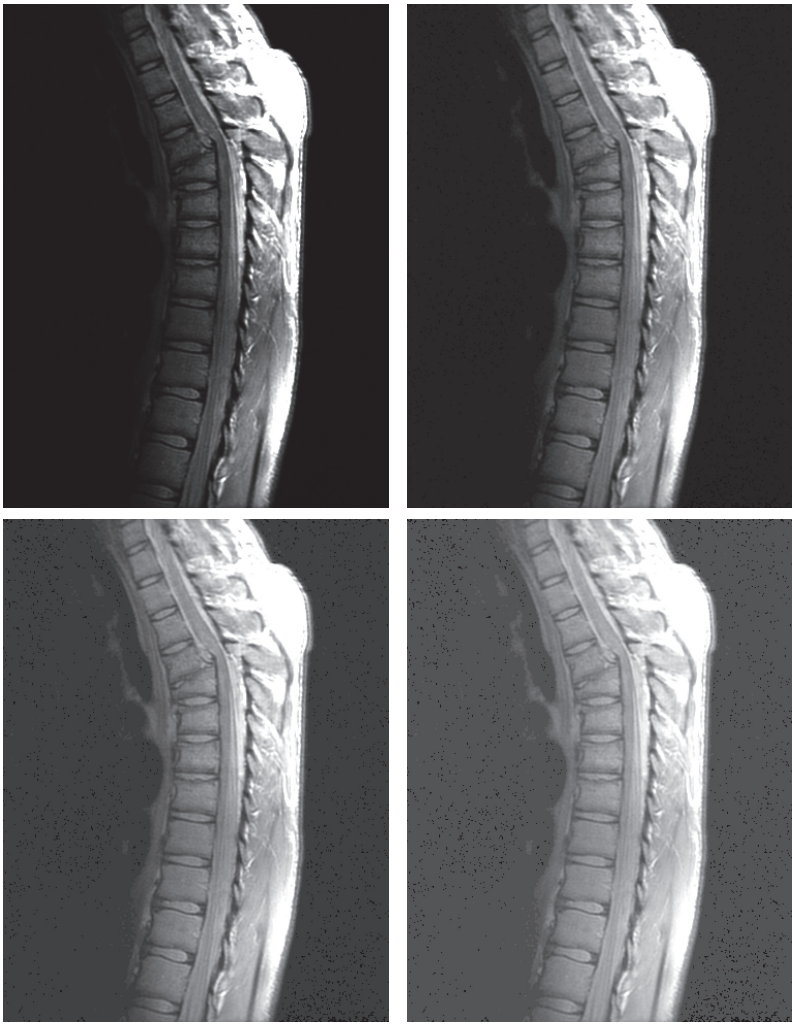


Рис. 3.8. (*а*) Снимок позвоночника человека с переломом; изображение получено с помощью ЯМР-томографа. (*б*)—(*г*) Результаты преобразований по формуле (3.2-3) с $c = 1$ и $\gamma = 0{,}6$, 0,4 и 0,3 соответственно. (Исходное изображение предоставил д-р Дэвид Пикенс, отделение радиологии и рентгенологии медицинского центра университета Вандербильта)

нием значения гамма от 0,6 до 0,4 становится видимым все большее количество деталей. Дальнейшее уменьшение гамма до 0,3 несколько усиливает детали фона, но снижает контраст до уровня, когда изображение приобретает вид «вылинявшего», что особенно заметно на фоновых участках. Сравнивая все изображения, можно видеть, что наилучший результат в смысле оценки контраста и различимости деталей достигается при $\gamma = 0{,}4$. Значение $\gamma = 0{,}3$ можно считать пределом, ниже которого контраст данного конкретного изображения уменьшается до неприемлемого уровня. ■

Пример 3.2. Другая иллюстрация степенного преобразования.
■ Изображение на рис. 3.9(*а*) демонстрирует пример задачи, противоположной той, которая решалась для изображения на рис. 3.8(*а*). В данном случае требуется обработать изображение, которое выглядит «вылинявшим», что указывает на необходимость понижения яркости. Это может быть достигнуто с помощью преобразования по формуле (3.2-3) со значениями γ больше 1. Результаты об-

а б
в г

Рис. 3.9. (*а*) Аэрофотоснимок. (*б*)—(*г*) Результаты преобразования по формуле (3.2-3) с $c = 1$ и $\gamma = 3{,}0$, 4,0 и 5,0 соответственно. (Исходное изображение предоставлено агентством NASA)

работки изображения рис. 3.9(*а*) при $\gamma = 3{,}0$, 4,0 и 5,0 показаны на рис. 3.9(*б*—*г*). Подходящие результаты достигаются при значениях гамма 3,0 и 4,0, причем последний вариант выглядит более предпочтительным, поскольку имеет больший контраст. На изображении, полученном при $\gamma = 5{,}0$, имеются слишком темные области, на которых часть деталей утеряна. Такие темные области можно наблюдать в левом верхнем квадранте, слева от основной дороги. ■

3.2.4. Кусочно-линейные функции преобразований

Подходом, дополняющим методы, рассмотренные в предыдущих трех разделах, является использование кусочно-линейных функций. Главное преимущество кусочно-линейных функций по сравнению с вышерассмотренными состоит в том, что их форма может быть сколь угодно сложной. На самом деле, как будет скоро показано, практическая реализация некоторых важных преобразований может быть осуществлена только с помощью кусочно-линейных функций. Основной недостаток кусочно-линейных функций заключается в том, что для их описания необходимо задавать значительно большее количество параметров.

Усиление контраста

Одним из простейших случаев использования кусочно-линейных функций является преобразование, усиливающее контрасты. Низкий контраст изображений может быть следствием плохого освещения, излишне большого динамического диапазона сенсора или даже неверно установленной диафрагмы объектива при съемке. *Усиление контраста* — операция, расширяющая диапазон яркостей изображения так, что он занимает весь динамический диапазон устройства записи или отображения.

На рис. 3.10(*а*) показано типичное преобразование, используемое для усиления контрастов. Положения точек (r_1, s_1) и (r_2, s_2) задают вид функции преобразования. Если $r_1 = s_1$ и $r_2 = s_2$, преобразование становится тождественным, не вносящим изменения в значения яркостей. Если $r_1 = r_2$, $s_1 = 0$ и $s_2 = L - 1$, преобразование превращается в *пороговую функцию*, которая в результате дает двухградационное изображение, как это показано на рис. 3.2(*б*). Промежуточные значения (r_1, s_1) и (r_2, s_2) обеспечивают различные степени растяжения уровней яркости на результирующем изображении, меняя тем самым его контраст. Вообще говоря, условия $r_1 \le r_1$ и $s_1 \le s_1$ означают, что функция является однозначной и монотонно возрастающей[2]. Это условие обеспечивает сохранение правильной последовательности уровней яркости, предотвращая тем самым появление ложных деталей на обработанном изображении.

На рис. 3.10(*б*) представлено исходное малоконтрастное 8-битовое изображение. На рис. 3.10(*в*) показан результат усиления контраста, полученный при $(r_1, s_1) = (r_{\min}, 0)$ и $(r_2, s_2) = (r_{\max}, L - 1)$, где $r_{\min}$ и $r_{\max}$ означают соответственно минимальную и максимальную яркости на изображении. Таким образом, функция преобразования линейно растягивает исходный диапазон яркостей в пол-

[2] На самом деле для этого требуется соблюдение строгих неравенств: $r_1 < r_1$ и $s_1 < s_1$. — *Прим. перев.*

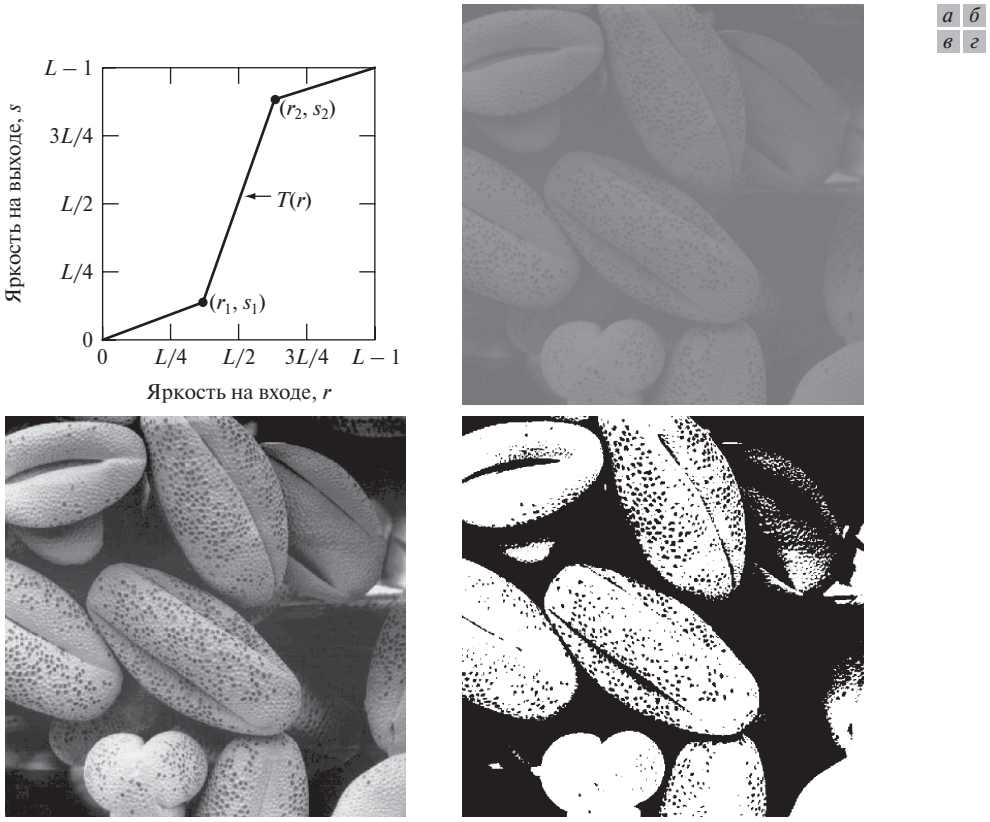


Рис. 3.10. Усиление контраста. (*а*) Вид функции преобразования. (*б*) Исходное малоконтрастное изображение. (*в*) Результат усиления контраста. (*г*) Результат порогового преобразования. (Исходное изображение предоставил д-р Роджер Хиди, факультет биологических исследований Австралийского национального университета, Канберра, Австралия)

ный диапазон [0, $L - 1$]. Наконец, на рис. 3.10(*г*) показан результат порогового преобразования с $r_1 = r_2 = m$, где m — среднее значение яркостей на изображении. В качестве исходного изображения в этом примере использован снимок цветочной пыльцы, полученный сканирующим электронным микроскопом с увеличением в 700 раз.

Вырезание диапазона яркостей

Зачастую интересно выделить какой-то конкретный диапазон яркостей на изображении. В практических применениях может потребоваться улучшение контраста отдельных деталей, таких как участки воды на спутниковых изображениях или дефекты изделий на рентгеновских снимках. Процедуры, называемые *вырезанием диапазона яркостей*, могут быть использованы по-разному, однако большинство из них являются вариациями двух следующих подходов. Первый подход состоит в отображении всех тех уровней, которые представляют интерес, некоторым одним значением (скажем, белым), а всех остальных уровней — другим (скажем, черным). Такое преобразование, показанное функцией

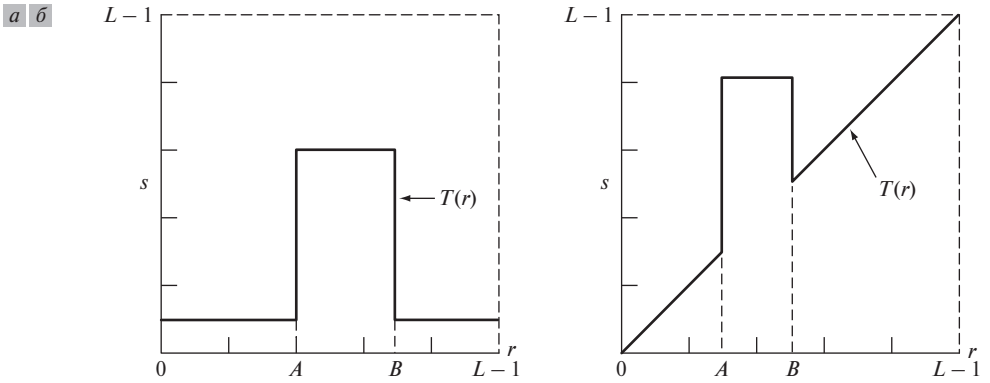


Рис. 3.11. (*а*) Данное преобразование выделяет диапазон яркостей [*A*, *B*] и приводит остальные значения к уровню константы. (*б*) Данное преобразование повышает яркость диапазона [*A*, *B*] и сохраняет все остальные уровни

на рис. 3.11(*а*), дает в результате двухградационное изображение. Второй подход, основанный на преобразовании с функцией на рис. 3.11(*б*), повышает (или понижает) яркость точек из выбранного диапазона, но сохраняет яркости фона и остальных точек изображения.

Пример 3.3. Вырезание диапазона яркостей.

■ На рис. 3.12(*а*) показана ангиограмма аорты в области почки (см. раздел 1.3.2 для более подробного пояснения данного изображения). Цель данного примера состоит в использовании вырезания диапазона яркостей, чтобы высветить основные кровеносные сосуды, которые выглядят несколько более яркими

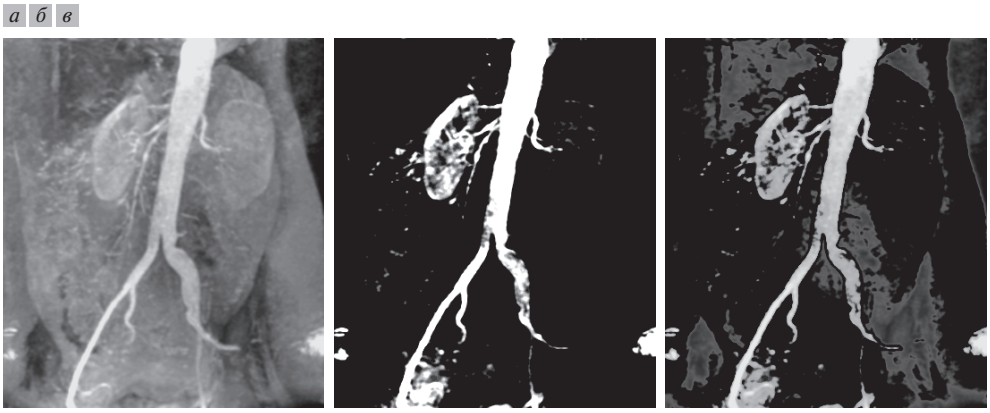


Рис. 3.12. (*а*) Ангиограмма аорты. (*б*) Результат применения преобразования вырезания диапазона яркостей по типу, показанному на рис. 3.11(*а*) с диапазоном интересующих яркостей в верхней части шкалы серого. (*в*) Результат преобразования с формой на рис. 3.11(*б*) и подавлением значений в выбранном диапазоне до черного, так что полутоновые значения в диапазоне кровеносных сосудов и почек сохранены. (Исходное изображение предоставил д-р Томас Р. Гест, медицинский факультет университета штата Мичиган)

за счет введения контрастного вещества. На рис. 3.12(*б*) представлен результат использования преобразования, отвечающего показанной на рис. 3.11(*а*) форме с выбранным диапазоном вблизи верхней части шкалы яркостей, поскольку интересующие детали ярче окружающего фона. Конечным результатом такого преобразования является то, что кровеносные сосуды и части почек выглядят на изображении белыми, тогда как все остальные детали оказываются черными. Преобразование этого типа дает в результате бинарное изображение, которое полезно для изучения картины потока контрастного вещества (например для обнаружения закупорки сосудов).

С другой стороны, если важно сохранить реальные яркости деталей в интересующем диапазоне, можно использовать преобразование по форме на рис. 3.11(*б*). На рис. 3.12(*в*) показан результат такого преобразования, при котором интервалу яркостей вблизи середины диапазона было присвоено значение черного, оставив остальные яркости не измененными. Видно, что яркостная тональность основных кровеносных сосудов и части областей почек осталась прежней. Такой результат может быть полезен, когда представляет интерес измерение фактического потока контрастного вещества как функции времени по серии изображений. ■

Вырезание битовых плоскостей

Цифровые значения пикселей состоят из битов. Например, яркость каждого пикселя полутонового монохромного 256-градационного изображения задается 8 битами (т. е. одним байтом). Вместо выделения диапазонов яркостей может оказаться полезным выделение информации о вкладе тех или иных битов в общее изображение. Как показано на рис. 3.13, 8-битовое изображение может быть представлено как композиция восьми однобитовых плоскостей, где плоскость 1 содержит младшие биты, а плоскость 8 — старшие биты.

На рис. 3.14(*а*) показано 8-битовое полутоновое изображение, а на рис. 3.14(*б*)—(*и*) — восемь его однобитовых плоскостей, причем рис. 3.14(*б*) соответствует младшему биту. Можно заметить, что старшие битовые плоскости (главным образом первые четыре) содержат основную часть визуально значимых данных. Младшие битовые плоскости дают вклад в детали изображения более тонких яркостных различий. Исходное изображение имеет серую границу, яркость которой равна 194. Заметим, что на некоторых битовых плоскостях

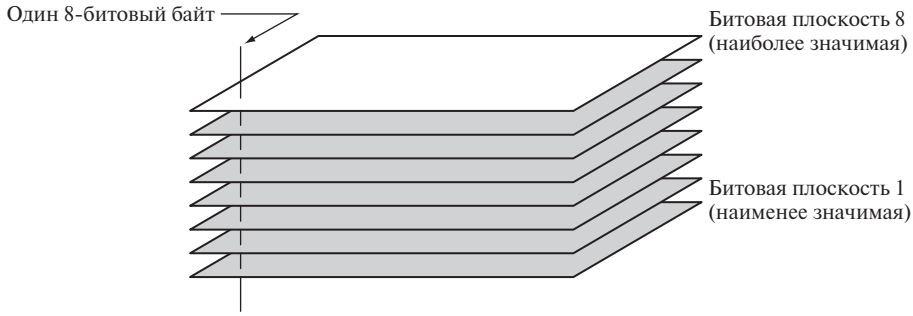


Рис. 3.13. Представление 8-битового изображения в виде набора битовых плоскостей

а *б* *в*
г *д* *е*
ж *з* *и*

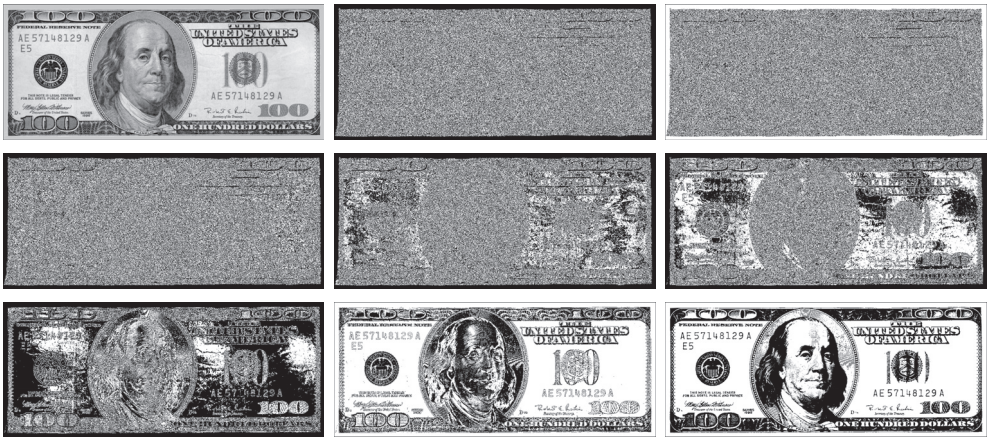


Рис. 3.14. (*а*) 8-битовое полутоновое изображение размерами 500×1192 пикселей. (*б*)—(*и*) Битовые плоскости от 1 до 8; битовая плоскость 1 соответствует младшим битам. Каждая битовая плоскость есть бинарное изображение

соответствующие границы оказываются черными (0), а на других — белыми (1). Чтобы понять, почему, рассмотрим, скажем, пиксель в центре нижней части границы изображения на рис. 3.14(*а*). Соответствующие пиксели в битовых плоскостях, начиная с плоскости старшего порядка, имеют значения 1 1 0 0 0 0 1 0, что является двоичным представлением числа 194. Значение любого пикселя исходного изображения может быть подобным образом восстановлено по значениям соответствующих двоичных пикселей в битовых плоскостях.

На основе функции градационного преобразования нетрудно показать, что (бинарное) изображение битовой плоскости 8 для 8-битового изображения может быть получено обработкой исходного изображения пороговым градационным преобразованием, которое отображает все уровни изображения от 0 до 127 в 0 и отображает все уровни от 128 до 255 в 1. Бинарное изображение на рис. 3.14(*и*) было получено именно таким образом. Формирование функций градационного преобразования для получения остальных битовых плоскостей оставим в качестве упражнения (задача 3.4).

Декомпозиция изображения на битовые плоскости полезна для анализа относительной информативности, которую несет каждый бит изображения, что позволяет оценить необходимое число битов, требуемое для квантования изображения. Также такая декомпозиция полезна при сжатии изображений (тема главы 8), при котором для восстановления изображения используются не все плоскости. Так, например, на рис. 3.15(*а*) показано изображение, восстановленное только из плоскостей 8 и 7. Восстановление осуществлялось умножением значения пикселя n-й плоскости на константу 2^{n-1}. Это есть не что иное, как перевод n-го значащего бита в десятичное значение. Значения каждой используемой плоскости умножаются на соответствующую константу и суммируются для получения полутонового изображения. Таким образом, чтобы получить

а б в

Рис. 3.15. Изображения, восстановленные при использовании (*а*) битовых плоскостей 8 и 7; (*б*) битовых плоскостей 8, 7 и 6; (*в*) битовых плоскостей 8, 7, 6 и 5. Сравните (*в*) с рис. 3.14(*а*)

изображение на рис. 3.15(*а*), необходимо умножить битовую плоскость 8 на 128, битовую плоскость 7 на 64 и сложить значения вместе. Хотя главные детали исходного изображения и восстановлены, однако полученное изображение выглядит плоским, особенно в области фона. Это не удивительно, поскольку две плоскости могут дать в результате лишь четыре различных уровня яркости. По рис. 3.15(*б*) видно, что добавление плоскости 6 в восстановление улучшает ситуацию. Отметим, что в фоновой части изображения появились ложные контуры. Этот эффект заметно уменьшается добавлением в восстановление плоскости 5, что видно на рис. 3.15(*в*). Использование в восстановлении большего числа плоскостей уже не даст заметного вклада в вид этого изображения. Отсюда мы делаем вывод, что сохранение четырех битовых плоскостей старших порядков позволит нам восстановить исходное изображение в приемлемых подробностях. Запоминание этих четырех плоскостей вместо оригинального изображения (отвлекаясь от вопросов архитектуры памяти) требует на 50 % меньше объема базы данных.

3.3. Видоизменение гистограммы

Гистограммой цифрового изображения с уровнями яркости в диапазоне $[0, L-1]$ называется дискретная функция $h(r_k) = n_k$, где r_k есть k-й уровень яркости, а n_k — число пикселей на изображении, имеющих яркость r_k. Общей практикой является нормировка гистограммы путем деления каждого из ее значений на общее число пикселей в изображении, обозначаемое произведением MN, где, как обычно, M и N суть число строк и столбцов изображения, то есть его размеры. Тем самым значения нормированной гистограммы будут $p(r_k) = n_k/MN$ для $k = 0, 1, ..., L-1$. Вообще говоря, $p(r_k)$ есть оценка вероятности появления пикселя[3] со значением яркости r_k. Заметим, что сумма всех значений нормированной гистограммы равна единице.

По вопросам основ теории вероятности обратитесь к интернет-сайту книги.

Гистограммы являются основой для многочисленных методов пространственной обработки. Как показано в настоящем разделе, видоизменение гистограммы (гистограммная обработка) может быть использовано для улучшения изображений. В последующих разделах мы увидим, что, кроме получения по-

[3] При условии некоррелированности значений яркостей пикселей. — *Прим. перев.*

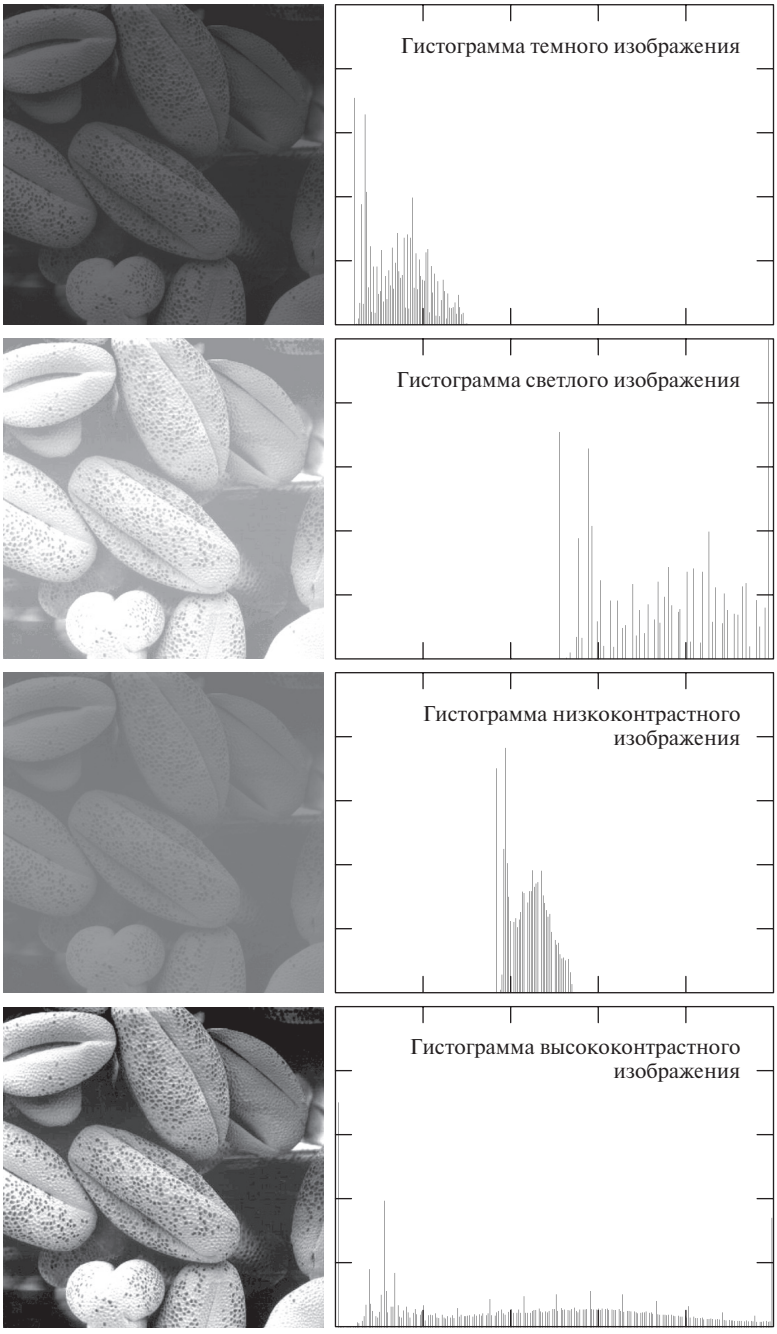


Рис. 3.16. Четыре основных типа изображения: темное, светлое, низкоконтрастное, высококонтрастное и соответствующие им гистограммы

лезной статистики об изображении, содержащаяся в гистограмме информация также весьма полезна и в других задачах, таких как сжатие и сегментация изображений. Гистограммы достаточно просты как для программного вычисле-

ния, так и для аппаратной реализации, что делает их удобным инструментом для обработки изображений в реальном времени.

В качестве ознакомления с видоизменением гистограммы для градационных преобразований рассмотрим рис. 3.16, на котором приведен тот же снимок пыльцы, что и на рис. 3.10, но показанный здесь в четырех вариантах яркостных характеристик: темном, светлом, низкоконтрастном и высококонтрастном. На правой части рисунка приведены гистограммы, соответствующие этим изображениям. По горизонтальной оси каждого графика отложены значения уровней яркости r_k. По вертикальной оси — значения гистограммы $h(r_k) = n_k$ или $p(r_k) = n_k/MN$, если они нормированы. Тем самым гистограммы могут быть представлены просто как графики зависимостей $h(r_k) = n_k$ от r_k или $p(r_k) = n_k/MN$ от r_k.

Легко видеть, что на гистограмме темного изображения ненулевые уровни сконцентрированы в области низких (темных) значений диапазона яркостей. Аналогично значимые уровни гистограммы светлого изображения смещены к верхней части диапазона. Изображение с низким контрастом имеет узкую гистограмму, расположенную, как правило, вблизи центра диапазона яркостей. Для одноцветного изображения это означает вялый, «вылинявший» серый вид. Наконец видно, что ненулевые уровни гистограммы высококонтрастного изображения покрывают широкую часть диапазона яркостей, а также что распределение значений пикселей не слишком отличается от равномерного, за исключением небольшого числа пиков, возвышающихся над остальными значениями. Интуитивно можно сделать вывод, что изображение, у которого распределение значений элементов близко к равномерному и занимает весь диапазон возможных значений яркостей, будет выглядеть высококонтрастным и будет содержать большое количество полутонов. Вскоре будет показано, что, основываясь только на информации, содержащейся в гистограмме исходного изображения, можно построить функцию преобразования, которая позволит автоматически добиваться такого эффекта.

3.3.1. Эквализация гистограммы

Для простоты сначала рассмотрим непрерывные значения яркостей, и пусть переменная r означает яркости элементов обрабатываемого изображения. Как обычно, при этом мы полагаем, что r распределена в диапазоне $[0, L - 1]$, причем значение $r = 0$ соответствует черному, а $r = L - 1$ — белому. Для любого r, удовлетворяющего вышеуказанным условиям, рассматривается преобразование (отображение яркости) вида

$$s = T(r) \quad 0 \le r \le L - 1, \tag{3.3-1}$$

которое для любого пикселя, имеющего значение r, дает значение s. Мы предполагаем, что функция преобразования $T(r)$ удовлетворяет следующим условиям: **(а)** $T(r)$ является монотонно неубывающей[4] функцией на интервале $0 \le r \le L - 1$; а также

[4] Напомним, что функция $T(r)$ является монотонно неубывающей, если $T(r_2) \ge T(r_1)$ для $r_2 > r_1$. $T(r)$ является *монотонно возрастающей*, если $T(r_2) > T(r_1)$ для $r_2 > r_1$. Аналогичные определения применимы к монотонно убывающим функциям.

(б) $0 \le T(r) \le L-1$ при $0 \le r \le L-1$.

В некоторых формулировках, обсуждаемых позже, используется обратное преобразование

$$r = T^{-1}(s) \quad 0 \le s \le L-1, \tag{3.3-2}$$

в случае которого условие **(а)** изменяется на

(а') $T(r)$ является монотонно возрастающей функцией на интервале $0 \le r \le L-1$.

В условии **(а)** требование монотонного возрастания функции $T(r)$ гарантирует, что возрастание яркости на входе не может привести к убыванию яркости на выходе, предотвращая тем самым артефакты, вызываемые инвертированием изменения яркости. Условие **(б)** означает, что допустимый диапазон выходных значений сигнала совпадает с диапазоном входных значений. Наконец, условие **(а')** гарантирует, что обратное отображение из s в r будет взаимно однозначным, предотвращая неопределенности. На рис. 3.17(*а*) показан пример функции преобразования, которая удовлетворяет условиям **(а)** и **(б)**. Как видно, здесь допускается отображение многих значений в одно, но функция все еще будет удовлетворять этим двум условиям. Это означает, что не строго монотонная функция допускает как отображения «один в один», так и «много в один». Это замечательно, если происходит отображение из r в s. Однако с зависимостью на рис. 3.17(*а*) возникают проблемы, если мы хотим по полученным значениям однозначно восстановить значение r (обратное отображение может быть представлено диагональным переворотом графика и переменой мест осей r и s). Обратное отображение возможно для точек вида s_k на рис. 3.17(*а*), но для точки s_q оно приводит к *диапазону* значений, что, вообще говоря, не позволяет восстановить исходное значение r по s_q. Как видно на рис. 3.17(*б*), требование, чтобы $T(r)$ являлась строго монотонной, гарантирует, что обратное отображение будет *однозначным* (т. е. отображение является взаимно однозначным). Это теоретически необходимые условия, которые позволяют реализовать некоторые важные методы видоизменения гистограммы, рассматриваемые позже в настоящей

а б

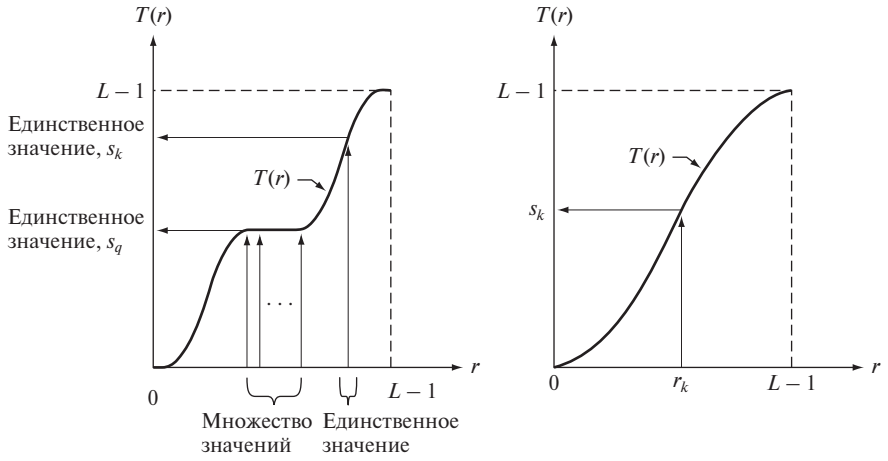


Рис. 3.17. (*а*) Монотонно неубывающая функция, показывающая, что много значений могут отображаться в одно. (*б*) Монотонно возрастающая функция. Отображение является взаимно однозначным

главе. Поскольку на практике приходится иметь дело не с действительными, а с целыми значениями яркости, мы вынуждены округлять все результаты до ближайшего целого. Таким образом, когда условие строгой монотонности не удовлетворяется, мы обходим проблему неоднозначности обратного преобразования отысканием ближайшего целого отображения. Пример 3.8 дает этому иллюстрацию.

Уровни яркости на изображении могут рассматриваться как значения случайной величины в интервале [0, $L-1$]. Важнейшей характеристикой случайной величины является плотность распределения вероятностей (ПРВ). Пусть $p_r(r)$ и $p_s(s)$ означают ПРВ случайных переменных r и s соответственно, где индекс при p означает, что $p_r(r)$ и $p_s(s)$ являются, вообще говоря, разными функциями. Из элементарной теории вероятностей следует, что если $p_r(r)$ и $T(r)$ известны и $T(r)$ является непрерывной и дифференцируемой на множестве интересующих значений, то ПРВ результата преобразования (отображения) — переменной s — может быть получена с помощью следующей простой формулы:

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right|. \tag{3.3-3}$$

Таким образом, ПРВ значений преобразованного сигнала s задается через ПРВ значений яркостей входного изображения и выбранную функцию преобразования (напомним, что r и s связаны функцией $T(r)$).

В обработке изображений особую важность имеет следующая функция:

$$s = T(r) = (L-1)\int_0^r p_r(w)dw, \tag{3.3-4}$$

где w — переменная интегрирования. Правая часть данного уравнения есть не что иное, как функция распределения (ФР) случайной переменной r. Поскольку ФР всегда положительна[5], а интеграл функции равен площади под графиком функции, следовательно, функция преобразования в уравнении (3.3-4) удовлетворяет условию **(а)**, т. к. площадь под графиком функции не может уменьшаться при увеличении r. Когда достигается верхний предел значений $r = L-1$, интеграл становится равным 1 (площадь под кривой ПРВ всегда равна 1), а максимальное значение s становится равным $(L-1)$; значит условие **(б)** также выполняется.

Зная функцию преобразования $T(r)$, ПРВ $p_s(s)$ можно найти из уравнения (3.3-3). Из дифференциального исчисления известно, что производная определенного интеграла по его верхнему пределу равна подынтегральному выражению в точке верхнего предела (правило Лейбница). Другими словами,

$$\frac{ds}{dr} = \frac{dT(r)}{dr} = (L-1)\frac{d}{dr}\left[\int_0^r p_r(w)dw\right] = (L-1)p_r(r). \tag{3.3-5}$$

Подставляя этот результат для dr/ds в уравнение (3.3-3) и предполагая, что все значения плотности вероятностей больше нуля, получаем в результате

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right| = p_r(r)\left|\frac{1}{(L-1)p_r(r)}\right| = \frac{1}{(L-1)} \qquad 0 \le s \le L-1. \tag{3.3-6}$$

[5] Вообще говоря, *неотрицательна*, но поскольку далее авторы вводят условие, что *все значения плотности вероятностей больше нуля*, то это утверждение допустимо. — *Прим. перев.*

а б

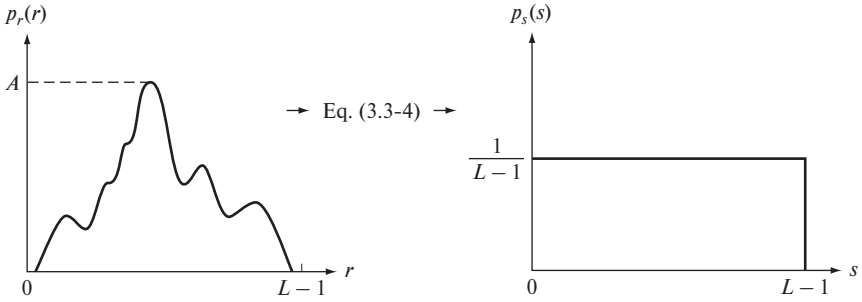


Рис. 3.18. (*а*) Произвольная ПРВ. (*б*) Результат преобразования согласно уравнению (3.3-4) для всех уровней яркости *r*. Результирующие яркости *s* имеют равномерную ПРВ независимо от формы ПРВ значений *r*

Таким образом, мы получили, что $p_s(s)$ есть *равномерная* плотность распределения вероятностей. Попросту говоря, было продемонстрировано, что выполнение градационного преобразования согласно функции, заданной уравнением (3.3-4), приводит к получению некоторой случайной величины *s*, характеризующейся равномерной ПРВ. Здесь важно заметить, что хотя $T(r)$, как это следует из (3.3-4), зависит от $p_r(r)$, результирующая ПРВ $p_s(s)$, как следует из (3.3-6), *всегда* является равномерной, *независимо* от формы $p_r(r)$. Рис. 3.18 иллюстрирует эту концепцию.

Пример 3.4. Иллюстрация применения уравнений (3.3-4) и (3.3-6).

■ Рассмотрим следующий простой пример. Предположим, что непрерывные значения яркости изображения имеют ПРВ следующего вида:

$$p_r(r)=\begin{cases}\dfrac{2r}{(L-1)^2} & \text{для } 0\le r\le L-1;\\ 0 & \text{в остальных случаях.}\end{cases}$$

Согласно уравнению (3.3-4),

$$s=T(r)=(L-1)\int_0^r p_r(w)dw=\frac{2}{L-1}\int_0^r w dw=\frac{r^2}{L-1}.$$

Предположим, что мы формируем новое изображение с яркостями *s*, полученными по этой формуле; т. е. значение *s* получается как квадрат значения яркости элемента исходного изображения, деленный на $(L-1)$. Например, пусть для некоторого изображения $L=10$ и пиксель в некоторой точке (x,y) имеет значение $r=3$. Тогда значение пикселя нового изображения в этой точке будет $s=T(r)=r^2/9=1$. Проверить, что ПРВ яркостей нового изображения будет равномерной, можно, просто подставив $p_r(r)$ в уравнение (3.3-6) и используя тот факт, что $s=r^2/(L-1)$; т. е.

$$p_s(s)=p_r(r)\left|\frac{dr}{ds}\right|=\frac{2r}{(L-1)^2}\left|\left[\frac{ds}{dr}\right]^{-1}\right|=\frac{2r}{(L-1)^2}\left|\left[\frac{d}{dr}\frac{r^2}{L-1}\right]^{-1}\right|=\frac{2r}{(L-1)^2}\left|\frac{L-1}{2r}\right|=\frac{1}{L-1},$$

где последнее действие следует из того факта, что r является неотрицательным[6] и что $L > 1$. Как и ожидалось, результатом является равномерная ПРВ. ■

В случае дискретных значений вместо плотностей распределения вероятностей и интегралов мы имеем дело с вероятностями (значениями гистограммы) и суммами. Как упоминалось выше, вероятность появления на цифровом изображении пикселя со значением яркости r_k приближенно равна

$$p_r(r_k) = \frac{n_k}{MN} \qquad k = 0,1,2,...,L-1, \tag{3.3-7}$$

где MN есть общее число пикселей на изображении, n_k — число точек яркости r_k, а L — максимально допустимое число уровней яркости на изображении (т. е. 256 для 8-битового изображения). Как отмечалось в начале данного раздела, зависимость $p_r(r_k)$ обычно называют *гистограммой*.

Дискретным аналогом функции преобразования, задаваемой уравнением (3.3-4), будет

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j) = \frac{(L-1)}{MN}\sum_{j=0}^{k} n_j \qquad k = 0,1,2,...,L-1. \tag{3.3-8}$$

Таким образом, обработанное (выходное) изображение получается отображением каждого пикселя входного изображения, имеющего яркость r_k, в соответствующий элемент выходного изображения со значением s_k, согласно уравнению (3.3-8). Преобразование (отображение), задаваемое уравнением (3.3-8), называется *эквализацией* или *линеаризацией гистограммы*. Нетрудно показать (задача 3.10), что данное преобразование удовлетворяет условиям (а) и (б), которые были ранее сформулированы в настоящем разделе.

Пример 3.5. Простая иллюстрация эквализации гистограммы.

■ Прежде чем продолжить, было бы полезно проработать простой пример. Предположим, что трехбитовое изображение ($L = 8$) размерами 64×64 пикселя ($MN = 4096$) имеет распределение яркостей, представленное в табл. 3.1; его уровни яркостей являются целыми в диапазоне $[0, L-1] = [0, 7]$.

Гистограмма нашего гипотетического изображения схематично изображена на рис. 3.19(*а*). Значения функции преобразования для эквализации гистограммы получены при помощи уравнения (3.3-8). Например,

$$s_0 = T(r_0) = 7\sum_{j=0}^{0} p_r(r_j) = 7\,p_r(r_0) = 1{,}33.$$

Аналогично,

$$s_1 = T(r_1) = 7\sum_{j=0}^{1} p_r(r_j) = 7\,p_r(r_0) + 7\,p_r(r_1) = 3{,}08$$

и $s_2 = 4{,}55$, $s_3 = 5{,}67$, $s_4 = 6{,}23$, $s_5 = 6{,}65$, $s_6 = 6{,}86$, $s_7 = 7{,}00$. Такая функция преобразования имеет ступенчатый вид, показанный на рис. 3.19(*б*).

---

[6] В этих преобразованиях необходимо также условие, что $r \neq 0$. — *Прим. перев.*

Таблица 3.1. Распределение яркостей и значения гистограммы для трехбитового изображения размерами 64×64 пикселя

r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0,19
$r_1 = 1$	1023	0,25
$r_2 = 2$	850	0,21
$r_3 = 3$	656	0,16
$r_4 = 4$	329	0,08
$r_5 = 5$	245	0,06
$r_6 = 6$	122	0,03
$r_7 = 7$	81	0,02

В данный момент значения s все еще являются дробными, поскольку они были получены суммированием значений вероятностей; теперь они округляются до ближайшего целого:

$$s_0 = 1{,}33 \to 1 \qquad s_4 = 6{,}23 \to 6$$
$$s_1 = 3{,}08 \to 3 \qquad s_5 = 6{,}65 \to 7$$
$$s_2 = 4{,}55 \to 5 \qquad s_6 = 6{,}86 \to 7$$
$$s_3 = 5{,}67 \to 6 \qquad s_7 = 7{,}00 \to 7$$

Это значения эквализованной гистограммы. Заметим, что в ней всего пять различающихся уровней яркости. Поскольку в $s_0 = 1$ отображаются лишь $r_0 = 0$, на изображении с эквализованной гистограммой будет 790 пикселей со значением 1 (см. табл. 3.1). Также на этом изображении будет 1023 пикселя со значением $s_1 = 3$ и 850 пикселей со значением $s_2 = 5$. Однако r_3 и r_4 отображаются в одно и то же значение 6, так что на эквализованном изображении будет (656 + 329) = 985 пикселей с этим значением, а также (245 + 122 + 81) = 448 пикселей со значением 7. Делением этих чисел на $MN = 4096$ получаем эквализованную гистограмму на рис. 3.19(*в*).

Поскольку гистограмма есть приближение ПРВ и в процессе преобразования новые уровни яркости не образуются, а могут лишь сливаться, получение в результате эквализации совершенно ровной гистограммы является на практике весьма редким случаем. Таким образом, в отличие от непрерывного случая, в общем случае нельзя доказать, что дискретная эквализация гистограммы

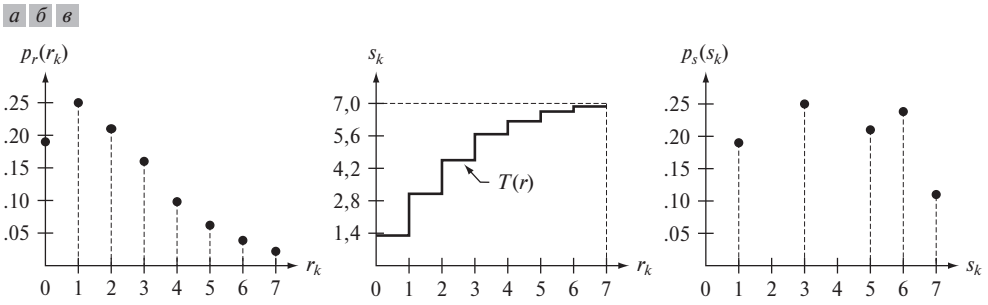


Рис. 3.19. Иллюстрация эквализации гистограммы трехбитового изображения (8 уровней яркости). (*а*) Исходная гистограмма. (*б*) Функция преобразования. (*в*) Эквализованная гистограмма

приводит к равномерной гистограмме. Однако, как скоро будет видно, использование уравнения (3.3-8) имеет тенденцию к растягиванию гистограммы исходного изображения, и уровни яркости эквализованного изображения занимают более широкий диапазон шкалы яркостей. В итоге результатом является повышение контрастов. ■

Ранее в настоящем разделе обсуждались многие преимущества изображения, уровни яркости которого покрывают весь диапазон возможных значений. Рассмотренный только что метод, кроме того, что формирует изображение, близкое к выдвинутому критерию, обладает тем дополнительным преимуществом, что является полностью «автоматическим». Иными словами, получая на вход изображение, процедура эквализации гистограммы сводится к выполнению преобразования по формуле (3.3-8), опираясь лишь на информацию, которая может быть извлечена непосредственно из обрабатываемого изображения без указания каких-либо дополнительных параметров. Стоит также отметить простоту вычислений, которые требуются для реализации этого метода.

Обратное преобразование из s в r определяется следующей формулой:

$$r_k = T^{-1}(s_k) \qquad k = 0, 1, 2, ..., L-1. \tag{3.3-9}$$

Может быть показано (задача 3.10), что это обратное преобразование удовлетворяет условиям **(а')** и **(б)** только в том случае, если не утерян ни один из уровней яркостей r_k, $k = 0, 1, 2, \ldots, L-1$ исходного изображения. Хотя при эквализации гистограммы обратное преобразование и не используется, оно играет центральную роль в схеме приведения гистограмм, рассматриваемой в следующем разделе.

Пример 3.6. Эквализация гистограммы.

■ В левом столбце рис. 3.20 представлены те же четыре изображения, что и на рис. 3.16, а в центральном столбце — результаты выполнения преобразований эквализации гистограммы по каждому из этих изображений. Первые три результата сверху вниз демонстрируют значительное улучшение. Как и ожидалось, на четвертом изображении эквализация гистограммы не приводит к появлению заметной визуальной разницы, поскольку яркости данного изображения изначально занимали весь диапазон значений. Графики функций, по которым производились преобразования изображений на рис. 3.20, представлены на рис. 3.21. Эти функции были получены по формуле (3.3-8). Заметим, что график преобразования (4) близок к линейной форме, опять-таки показывая этим, что уровни яркости на четвертом изображении близки к равномерному распределению.

Гистограммы изображений после эквализации представлены в правом столбце на рис. 3.20. Интересно отметить, что хотя все эти гистограммы и различаются, тем не менее эквализованные изображения выглядят весьма похоже. Это не является неожиданным, поскольку основная разница между изображениями в левой колонке заключается только в их контрастах, а не содержании. Другими словами, поскольку изображения имеют одно и то же содержание, то увеличения контраста, получаемого путем эквализации гистограммы, достаточно, чтобы компенсировать разницы в яркостях и сделать эквализованные

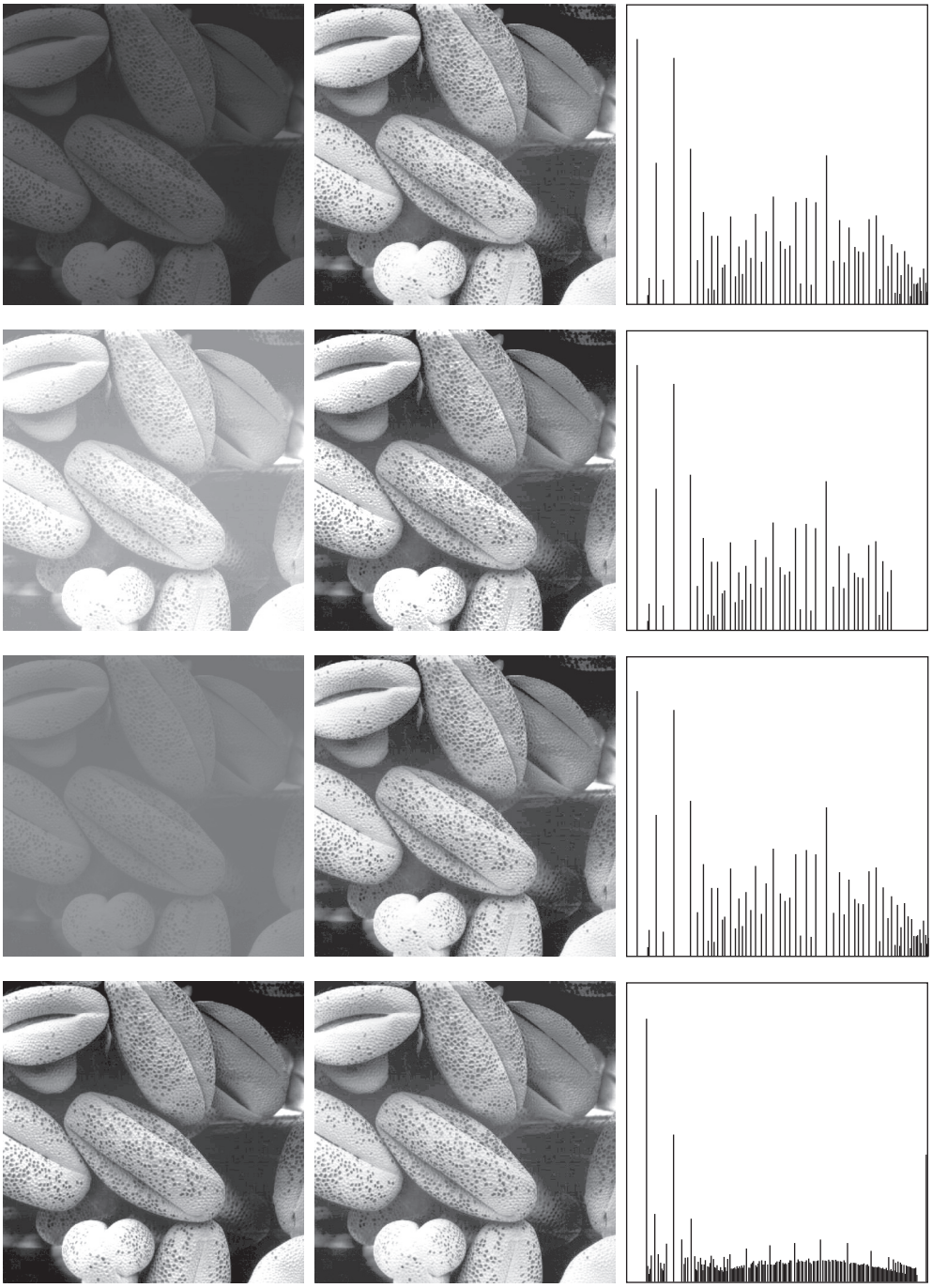


Рис. 3.20. Левый столбец: изображения из рис. 3.16. Центральный столбец: соответствующие результаты эквализации гистограммы. Правый столбец: гистограммы изображений в центральном столбце

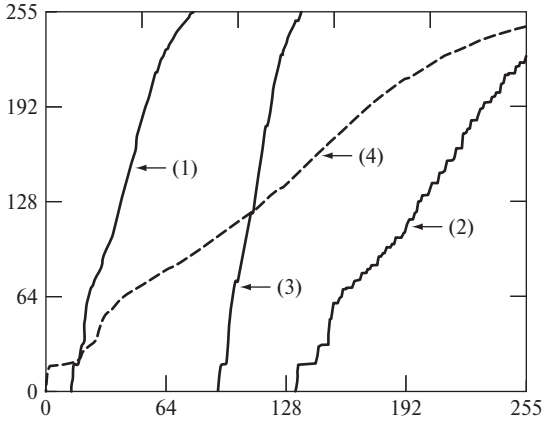


Рис. 3.21. Функции преобразований для эквализации гистограммы. Функции (1)—(4) были получены на основе гистограмм изображений, находящихся в левом столбце (сверху вниз) на рис. 3.20, с помощью уравнения (3.3-8)

изображения визуально неотличимыми. На этом примере с изображениями, изначально имеющими значительные разницы в контрастах, иллюстрируются возможности эквализации гистограммы как адаптивного инструмента улучшения изображений. ■

3.3.2. Приведение гистограммы (задание гистограммы)

Как обсуждалось выше, эквализация гистограммы автоматически находит функцию преобразования, которая стремится сформировать выходное изображение с равномерной гистограммой. В случае необходимости автоматического улучшения это является хорошим подходом, поскольку результаты данного метода предсказуемы и он прост в реализации. В настоящем разделе мы покажем, что в некоторых случаях улучшение, основанное на модели равномерной гистограммы, не является наилучшим подходом. В частности, иногда полезно задать иную желаемую форму гистограммы для обрабатываемого изображения. Метод, позволяющий получить обработанное изображение с задаваемой формой гистограммы, называется методом *приведения гистограммы* или *задания гистограммы*.

Вернемся ненадолго к непрерывному представлению яркостей r и z (рассматриваемые непрерывные случайные переменные), которые обозначают уровни яркостей входного и выходного (обработанного) изображений, и пусть $p_r(r)$ и $p_z(z)$ соответственно означают их непрерывные плотности распределения вероятностей. Значения $p_r(r)$ мы можем оценить по исходному изображению, в то время как $p_z(z)$ является *задаваемой* плотностью распределения вероятностей, которую должно иметь выходное изображение.

Пусть s — случайная переменная со следующими свойствами:

$$s = T(r) = (L-1)\int_0^r p_r(w)dw, \qquad (3.3\text{-}10)$$

где, как и ранее, w — переменная интегрирования. Это выражение есть не что иное, как непрерывная форма эквализации гистограммы — повторение уравнения (3.3-4).

Предположим также, что z — еще одна случайная переменная со свойством

$$G(z) = (L-1)\int_0^z p_z(t)dt = s, \tag{3.3-11}$$

где t — переменная интегрирования. Из этих двух уравнений следует, что $G(z) = T(r)$, а значит, z должно подчиняться следующему условию:

$$z = G^{-1}(s) = G^{-1}[T(r)]. \tag{3.3-12}$$

Преобразование $T(r)$ может быть получено из уравнения (3.3-10) сразу, как только $p_r(r)$ оценена по входному изображению. Подобным образом, используя уравнение (3.3-11), можно получить функцию преобразования $G(z)$, поскольку задана $p_z(z)$.

Из уравнений (3.3-10)—(3.3-12) следует, что изображение с заданной плотностью распределения вероятностей может быть получено из исходного изображения с помощью следующей процедуры.

1. Получить $p_r(r)$ исходного изображения и использовать уравнение (3.3-10) для получения s.

2. Используя выбранную ПРВ $p_z(z)$, с помощью уравнения (3.3-11) получить функцию преобразования $G(z)$.

3. Вычислить функцию обратного преобразования $z = G^{-1}(z)$; поскольку z получена из s, эта процедура есть отображение s в z, что и является искомым значением.

4. Получить выходное изображение путем первоначальной эквализации исходного по формуле (3.3-10); значения пикселей данного изображения являются значениями s. Для каждого пикселя s эквализованного изображения выполнить обратное отображение $z = G^{-1}(s)$, чтобы получить значение соответствующего пикселя выходного изображения. В результате обработки всех пикселей с помощью такой процедуры получится изображение, ПРВ которого будет совпадать с выбранной ПРВ.

Пример 3.7. Приведение гистограммы.

■ Полагая значения яркостей величинами непрерывными, предположим, что изображение имеет ПРВ яркости, равную $p_r(r) = 2r/(L-1)^2$ для $0 \le r \le (L-1)$ и $p_r(r) = 0$ для остальных значений r. Найдем функцию преобразования, которая даст в результате изображение с ПРВ яркости, равной $p_z(z) = 3z^2/(L-1)3$ для $0 \le z \le (L-1)$ и $p_z(z) = 0$ для остальных значений z.

Во-первых, найдем преобразование эквализации гистограммы на интервале $[0, L-1]$:

$$s = T(r) = (L-1)\int_0^r p_r(w)dw = \frac{2}{(L-1)}\int_0^r w\,dw = \frac{r^2}{(L-1)}.$$

По определению, это выражение равно 0 для значений вне отрезка $[0, L-1]$. Таким образом, возводя значения входных яркостей в квадрат и деля их на $(L-1)$, получаем изображение, значение яркостей пикселей s которого имеет равно-

мерно распределенную ПРВ, поскольку это есть преобразование, эквализующее гистограмму, как было обсуждено ранее.

Нас интересует получение изображения с заданной гистограммой, поэтому находим функцию $G(z)$, равную

$$G(z) = (L-1)\int_0^z p_z(w)dw = \frac{3}{(L-1)^2}\int_0^z w^2\,dw = \frac{z^3}{(L-1)^2}$$

на интервале [0, $L-1$] и 0 вне этого интервала. Наконец, мы требуем, чтобы $G(z) = s$. Но $G(z) = z^3/(L-1)^2$, а значит, $z^3/(L-1)^2 = s$, и мы получаем:

$$z = \left[(L-1)^2 s\right]^{1/3}.$$

Тем самым, если мы умножим значение каждого элемента изображения с эквализованной гистограммой на $(L-1)^2$ и возведем в степень 1/3, то в результате получим изображение, значения яркостей s которого будут иметь ПРВ, равную $p_z(z) = 3z^2/(L-1)^3$ на интервале [0, $L-1$], что и требовалось.

Поскольку $s = r^2/(L-1)$, можно получить значения z напрямую из значений яркостей r исходного изображения:

$$z = \left[(L-1)^2 s\right]^{1/3} = \left[(L-1)^2\frac{r^2}{(L-1)}\right]^{1/3} = \left[(L-1)r^2\right]^{1/3}.$$

Таким образом, возводя в квадрат значение каждого пикселя исходного изображения, умножая результат на $(L-1)$ и возводя произведение в степень 1/3, получим изображение, яркости элементов которого имеют заданную ПРВ. Видно, что промежуточный шаг эквализации исходного изображения может быть опущен; все, что требуется, — это получить функцию преобразования $T(r)$, которая отображает r в s. Затем два шага могут быть объединены в одно преобразование из r в z. ■

Как видно из примера, процедура приведения гистограммы в принципе является достаточно прозрачной. На практике основная трудность заключается в отыскании содержательного аналитического выражения для $T(r)$ и G^{-1}. К счастью, в случае дискретных величин эта задача значительно упрощается. Издержки остаются теми же, что и в случае эквализации гистограммы, когда достижимым является только некоторое приближение к желаемой гистограмме. Несмотря на это, однако, некоторые весьма полезные результаты могут быть получены даже с достаточно грубыми приближениями.

Дискретная формулировка уравнения (3.3-10) задана преобразованием эквализации гистограммы согласно уравнению (3.3-8), которое мы здесь повторяем для удобства:

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j) = \frac{(L-1)}{MN}\sum_{j=0}^{k} n_j \qquad k = 0,1,2,\ldots,L-1, \tag{3.3-13}$$

где, как и ранее, MN есть общее число пикселей в изображении, n_j — число точек яркости r_j, а L — общее число возможных уровней яркости на изображении. Подобным образом при задании желаемых значений s_k дискретная формулировка уравнения (3.3-11) приводит к вычислению функции преобразования вида

$$G(z_q) = (L-1)\sum_{i=0}^{q} p_z(z_i) \tag{3.3-14}$$

для значений q таких, что

$$G(z_q) = s_k, \tag{3.3-15}$$

где $p_z(z_i)$ — i-тое значение заданной гистограммы. Как и ранее, дискретное значение z_i находится выполнением обратного преобразования:

$$z_q = G^{-1}(s_k). \tag{3.3-16}$$

Другими словами, данная операция дает значение z для каждого исходного значения s; тем самым происходит *отображение* из s в z.

На практике нет необходимости вычислять обратное преобразование для каждого элемента из G. Поскольку мы имеем дело с уровнями яркостей, являющимися целыми значениями (например от 0 до 255 для 8-битового изображения), с помощью уравнения (3.3-14) несложно вычислить все возможные значения G для $q = 0, 1, 2, \ldots, L-1$. Эти значения масштабируются и округляются до ближайшего целого, занимая весь диапазон $[0, L-1]$. Полученные значения запоминаются в таблице. Затем, получив конкретное значение s_k, мы отыскиваем ближайшее значение в таблице. Если, например, ближайшим к s_k является значение в 64-й ячейке таблицы, тогда $q = 63$ (напомним, что отсчет начинается с 0) и z_{63} является наилучшим решением уравнения (3.3-15). Таким образом, данное значение s_k будет ассоциировано с z_{63} (т. е. конкретное значение s_k будет *отображаться* на z_{63}). Поскольку значения z суть яркости, используемые как аргументы для задания гистограммы $p_z(z)$, то $z_0 = 0$, $z_1 = 1, \ldots, z_{L-1} = L-1$, а z_{63} будет иметь значение яркости 63. Повторяя эту процедуру, мы найдем для каждого из значений s_k отображение в z_q, являющееся ближайшим к решению уравнения (3.3-15). Полученные отображения и являются решением задачи приведения гистограммы.

Вспомнив, что s_k являются значениями изображения с эквализованной гистограммой, можно кратко изложить процедуру приведения гистограммы в следующем виде:

1. Получить гистограмму $p_r(r)$ исходного изображения и использовать ее для отыскания преобразования эквализации гистограммы по формуле (3.3-13). Округлить результирующие значения s_k до целых в диапазоне $[0, L-1]$.
2. С помощью уравнения (3.3-14) вычислить значения функции преобразования G для $q = 0, 1, 2, \ldots, L-1$, где $p_z(z_i)$ суть значения задаваемой гистограммы. Округлить значения G до целых в диапазоне $[0, L-1]$. Запомнить значения G в таблице.
3. Используя полученные на шаге 2 значения G, найти для каждого значения s_k, $k = 0, 1, 2, \ldots, L-1$ соответствующее значение z_q такое, что $G(z_q)$ является ближайшим к s_k, и сохранить эти отображения из s в z. Если какому-то значению z_q соответствует более одного значения s_k (т. е. отображение не является единственным), взять для определенности наименьшее значение.
4. Сформировать изображение с приведенной гистограммой путем (1) эквализации гистограммы исходного изображения и (2) отображения

каждого эквализованного значения пикселя s_k этого изображения в соответствующее значение z_q изображения с приведенной гистограммой, используя отображение, найденное на шаге 3. Как и в непрерывном случае, промежуточный шаг эквализации исходного изображения является умозрительным. Он может быть опущен объединением двух функций преобразования T и G^{-1}, как это показано в примере 3.8

Как указывалось ранее, чтобы преобразование G^{-1} удовлетворяло условиям **(а')** и **(б)**, G должна быть строго монотонной функцией, а это в соответствии с уравнением (3.3-14) означает, что ни одно из значений $p_z(z_i)$ задаваемой гистограммы не может быть нулевым (проблема 3.10). Как видно из вышеописанного шага 3, при работе с дискретными значениями тот факт, что это условие может не удовлетворяться, не создает серьезных препятствий. Следующий пример иллюстрирует этот факт численно.

Пример 3.8. Простой пример приведения гистограммы.

■ Рассмотрим опять гипотетическое изображение размерами 64×64 из примера 3.5, гистограмма которого повторена на рис. 3.22(*а*). Желательно преобразовать эту гистограмму так, чтобы она имела значения, указанные во втором столбце табл. 3.2. На рис. 3.22(*б*) показан схематичный вариант такой гистограммы.

Первым шагом процедуры является определение масштабированных значений эквализованной гистограммы, которые мы получили в примере 3.5:

$$s_0 = 1 \quad s_2 = 5 \quad s_4 = 6 \quad s_6 = 7$$
$$s_1 = 3 \quad s_3 = 6 \quad s_5 = 7 \quad s_7 = 7$$

На следующем шаге, используя уравнение (3.3-14), вычисляются все значения функции преобразования G:

$$G(z_0) = 7\sum_{i=0}^{0} p_z(z_i) = 0{,}00.$$

Аналогично

$$G(z_1) = 7\sum_{i=0}^{1} p_z(z_i) = 7\left[p(z_0) + p(z_1)\right] = 0{,}00$$

и

$$G(z_2) = 0{,}00 \quad G(z_4) = 2{,}45 \quad G(6_6) = 5{,}95,$$

$$G(z_3) = 1{,}05 \quad G(z_5) = 4{,}55 \quad G(z_7) = 7{,}00.$$

Так же как и в примере 3.5, эти дробные значения преобразуются в целые в диапазоне [0, 7]. В результате будет:

$$G(z_0) = 0{,}00 \to 0 \qquad G(z_4) = 2{,}45 \to 2$$
$$G(z_1) = 0{,}00 \to 0 \qquad G(z_5) = 2{,}45 \to 5$$
$$G(z_2) = 0{,}00 \to 0 \qquad G(z_6) = 2{,}45 \to 6$$
$$G(z_3) = 0{,}00 \to 1 \qquad G(z_7) = 2{,}45 \to 7$$

Эти результаты приведены в табл. 3.3, и функция преобразования схематично показана на рис. 3.22(*в*). Видно, что G не является строго монотонной, так что условие **(а')** не выполняется. Поэтому, чтобы исправить ситуацию, воспользуемся приемом, указанным на шаге 3 алгоритма.

На третьем шаге процедуры ищутся наименьшие значения z_q такие, что значение $G(z_q)$ является ближайшим к s_k. Чтобы создать требуемое отображение из s в z, эта операция выполняется для всех s_k. Например, $s_0 = 1$ и $G(z_3) = 1$, что яв-

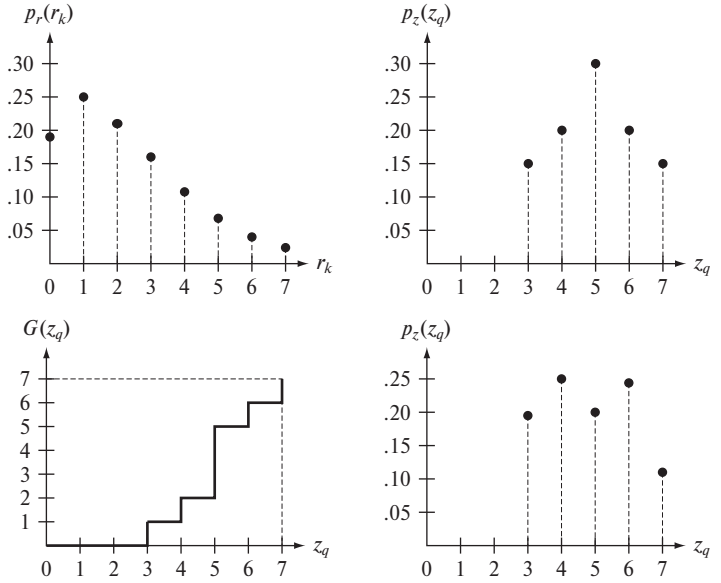


Рис. 3.22. (*а*) Гистограмма трехбитового изображения. (*б*) Заданная гистограмма. (*в*) Функция преобразования, полученная по заданной гистограмме. (*г*) Результат применения процедуры приведения гистограммы. Сравните (*б*) и (*г*)

ляется в данном случае прекрасным совпадением, таким образом, имеется соответствие $s_0 \rightarrow z_3$. То есть каждый пиксель, имеющий значение 1 на изображении с эквализованной гистограммой, будет отображен в пиксель со значением 3 (в соответствующей точке) на изображении с приведенной гистограммой. Повторяя эту процедуру для остальных значений, мы получим все отображения, показанные в табл. 3.4.

На заключительном шаге процедуры используются отображения, представленные в табл. 3.4, чтобы преобразовать каждый пиксель изображения с эквализованной гистограммой в соответственный пиксель формируемого изображения с приведенной гистограммой. Значения результирующей гистограммы приведены в третьем столбце табл. 3.2, а сама гистограмма показана на рис. 3.22(*г*). Значения $p_z(z_k)$ были получены с использованием той же процедуры, что и в примере 3.5. Например, как видно из табл. 3.4, $s = 1$ отображается в $z = 3$, но на изображении с эквализованной гистограммой 790 пикселей имеют значение 1. Таким образом, $p_z(z_3) = 790/4096 = 0{,}19$.

Хотя окончательный результат, представленный на рис. 3.22(*г*), и не совпадает абсолютно с заданной гистограммой, общая тенденция к сдвигу яркостей в сторону верхнего конца диапазона была достигнута. Как уже отмечалось ранее, формирование изображения с эквализованной гистограммой в качестве промежуточного шага полезно для объяснения процедуры, но не является необходимым. Вместо этого можно было бы перечислить отображения из значений r в s и из s в z в виде трехстолбцовой таблицы. Затем можно было бы использовать эти отображения для прямого преобразования исходных значений пикселей в значения пикселей изображения с приведенной гистограммой. ■

Таблица 3.2. Заданная и реальная гистограммы (значения в третьем столбце получены в процессе вычислений в примере 3.8).

	Заданная	Реальная
z_q	n_k	$p_z(z_k)$
$r_0 = 0$	0,00	0,00
$r_1 = 1$	0,00	0,00
$r_2 = 2$	0,00	0,00
$r_3 = 3$	0,15	0,19
$r_4 = 4$	0,20	0,25
$r_5 = 5$	0,30	0,21
$r_6 = 6$	0,20	0,24
$r_7 = 7$	0,15	0,11

Таблица 3.3. Все возможные значения функции преобразования G, масштабированные, округленные и упорядоченные по z

z_q	$G(z_q)$
$r_0 = 0$	0
$r_1 = 1$	0
$r_2 = 2$	0
$r_3 = 3$	1
$r_4 = 4$	2
$r_5 = 5$	5
$r_6 = 6$	6
$r_7 = 7$	7

Таблица 3.4. Отображения всех значений s_k в соответствующие значения z_q

s_k	→	z_q
$r_1 = 1$	→	3
$r_3 = 3$	→	4
$r_5 = 5$	→	5
$r_6 = 6$	→	6
$r_7 = 7$	→	7

Пример 3.9. Сравнение эквализации гистограммы и приведения гистограммы.

■ На рис. 3.23(*а*) показан снимок спутника Марса — Фобоса, сделанный космической станцией Mars Global Surveyor (NASA), а на рис. 3.23(*б*) представлена его гистограмма. На изображении преобладают большие темные области, вследствие чего точки на изображении концентрируются вблизи темного края шка-

а *б*

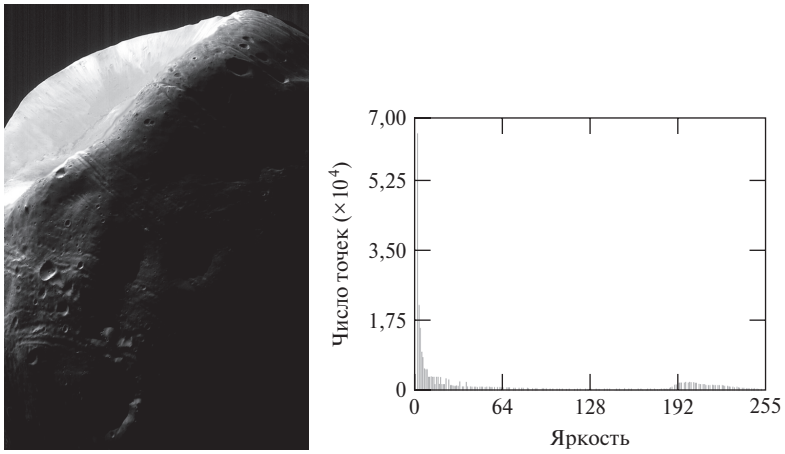


Рис. 3.23. (*а*) Снимок спутника Марса — Фобоса, сделанный космической станцией *Mars Global Surveyor* (NASA). (*б*) Его гистограмма. (Исходный снимок предоставлен агентством NASA)

лы яркостей и на гистограмме появляется пик вблизи нуля. На первый взгляд кажется, что эквализация гистограммы будет хорошим способом улучшения изображения, таким, что детали в темных областях станут более видимыми. Далее будет продемонстрировано, что это не так.

На рис. 3.24(*а*) и (*б*) представлен результат преобразования эквализации гистограммы (уравнения (3.3-8) или (3.3-13)), полученный по гистограмме на рис. 3.23(*б*). Наиболее характерной деталью данной функции преобразования является то, насколько возрос уровень черного — от 0 до почти 190. Причина заключается в том, что большинство значений пикселей входного изображения сконцентрировано вблизи нулевых уровней гистограммы. Эффект применения данного преобразования к входному изображению с целью эквализации гистограммы сводится к тому, что очень узкий интервал темных пикселей отображается в верхнюю часть яркостного диапазона выходного изображения. Поскольку большая часть пикселей входного изображения имеет значения как раз в этом интервале, следовало бы ожидать, что результатом будет светлое изображение, выглядящее как «вылинявшее». Как видно по изображению на рис. 3.24(*б*), именно так и происходит. Гистограмма обработанного изображения представлена на рис. 3.24(*в*). Заметим, насколько все уровни яркости смещены в верхнюю половину диапазона.

Поскольку проблемы с функцией преобразования на рис. 3.24(*а*) возникли из-за высокой концентрации значений пикселей исходного изображения вблизи нуля, разумным подходом было бы модифицировать гистограмму таким образом, чтобы избежать этого обстоятельства. Так, на рис. 3.22(*а*) показана искусственно заданная функция, которая сохраняет основную форму исходной гистограммы, но имеет сглаженный переход уровней в темной области шкалы яркостей. Равномерная дискретизация этой функции на 256 значений и будет необходимой заданной гистограммой. Функция преобразования $G(z)$, полученная на основе этой гистограммы с помощью уравнения (3.3-14), показана графиком (1) на рис. 3.22(*б*). Аналогично обратное преобразование $G^{-1}(s)$ из уравнения

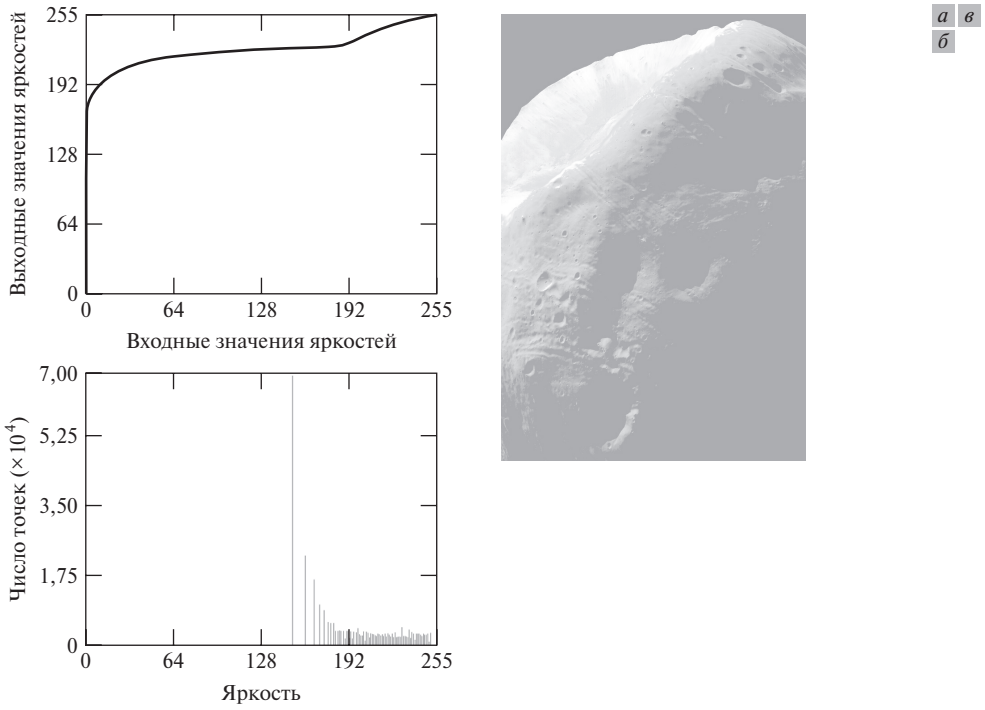


Рис. 3.24. (*а*) Функция преобразования эквализации гистограммы. (*б*) Эквализованное изображение (хорошо виден эффект осветления). (*в*) Гистограмма изображения (*б*)

(3.3-16), полученное итеративным путем, обсуждавшимся в связи с неравенством (3.3-17), показано графиком (2) на рис. 3.22(*б*). Изображение на рис. 3.22(*в*) было получено применением преобразования (2) к значениям пикселей эквализованного изображения на рис. 3.21(*б*). При сравнении этих двух изображений становится очевидным преимущество улучшения изображения по методу задания гистограммы по сравнению с эквализацией гистограммы. Интересно отметить, что сравнительно небольших изменений исходной гистограммы было достаточно, чтобы значительно улучшить результаты. Гистограмма изображения на рис. 3.22(*в*) показана на рис. 3.22(*г*). Наиболее характерной особенностью данной гистограммы (по сравнению с исходной) является то, что ее начальный участок сдвинулся вправо — к более светлым значениям шкалы яркостей (но не чрезмерно), что и являлось целью. ■

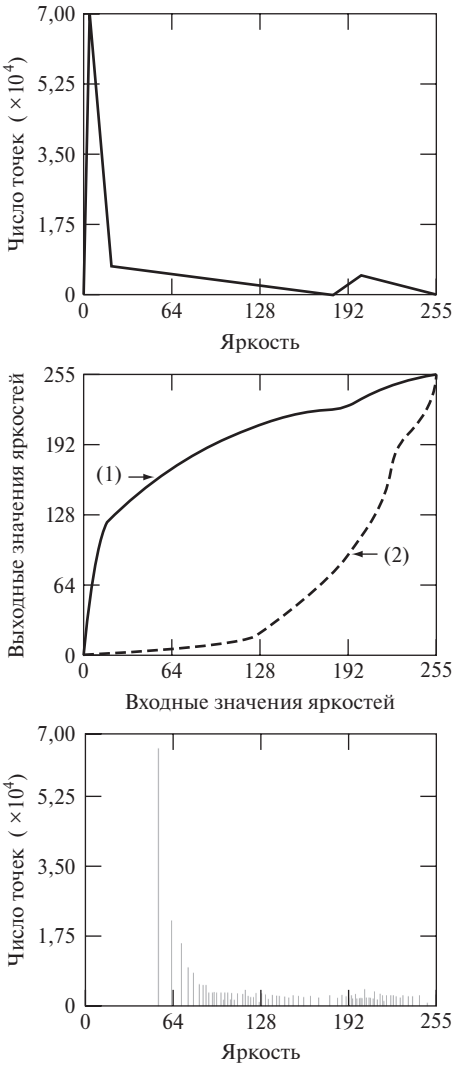
Хотя к настоящему моменту это уже должно быть очевидно, в заключение данного раздела имеет смысл еще раз подчеркнуть, что метод задания гистограммы в большинстве случаев является процессом проб и ошибок. Можно использовать руководства, составленные на основе встречавшихся проблем, как это было сделано в предыдущем примере. Временами могут встречаться случаи, когда можно сформулировать, как именно должна выглядеть «средняя» гистограмма, и использовать ее как задаваемую гистограмму. В таких случаях метод приведения гистограммы становится достаточно простым. Но, вообще говоря,

не существует общих правил по выбору гистограммы, и следует раз за разом прибегать к анализу в каждой из задач улучшения изображений.

3.3.3. Локальная гистограммная обработка

Рассмотренные в предыдущих двух разделах методы гистограммной обработки являлись *глобальными*, что означало построение функции преобразования на основе анализа распределения яркостей по всему изображению. Хотя такой глобальный подход и пригоден для улучшения в целом, существуют случаи, когда приходится улучшать детали на основе анализа малых областей изображения. Связано это с тем, что число пикселей в таких областях мало и не может оказывать заметного влияния на глобальную гистограмму, форма которой не обязательно соответствует необходимому локальному улучшению. Решение

а *в*
б
г



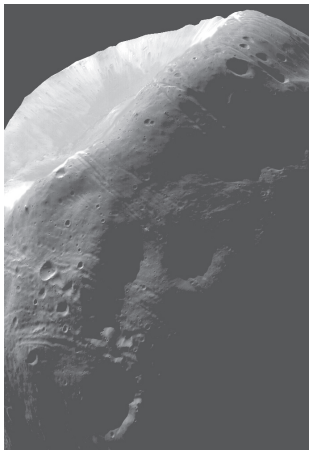


Рис. 3.25. (*а*) Заданная форма гистограммы. (*б*) Кривая (1) получена с помощью уравнения (3.3-14) из гистограммы (*а*); кривая (2) получена итеративным путем, который обсуждался в связи с неравенством (3.3-17). (*в*) Улучшение изображения при использовании кривой (2). (*г*) Гистограмма изображения (*в*)

состоит в разработке функции преобразования, основанной на распределении яркостей по окрестности каждого элемента изображения.

Описанные ранее методы гистограммной обработки могут быть легко применены и к локальному улучшению. Процедура состоит в том, что задается окрестность с центром в обрабатываемом элементе и центр этой окрестности передвигается от точки к точке. Для каждого нового положения окрестности подсчитывается гистограмма по входящим в нее точкам и находится функция преобразования эквализации или приведения гистограммы. Затем эта функция используется для отображения уровня яркости центрального элемента окрестности. Затем центр окрестности перемещается на соседний пиксель и процедура повторяется. Поскольку при перемещении окрестности от точки к точке меняется только один ее столбец или строка, то становится возможным обновление гистограммы, полученной на предыдущем шаге, путем добавления новых данных (задача 3.12). Такой подход имеет очевидные преимущества по сравнению с вычислением гистограммы заново по всем точкам окрестности при ее смещении всего на один элемент. Другим подходом, применяемым иногда для уменьшения количества вычислений, является использование непересекающихся областей[7], но такой метод обычно приводит к появлению нежелательного «блочного» эффекта.

Пример 3.10. Эквализация локальной гистограммы.

■ На рис. 3.26(*а*) представлено 8-битовое изображение размерами 512×512 элементов, которое, на первый взгляд, содержит пять черных квадратов на сером фоне. Изображение слегка зашумлено, но шум незаметен. На рис. 3.26(*б*) показан результат глобальной эквализации гистограммы. Как бывает часто с эквализацией гистограммы гладких, но зашумленных областей, произошло значительное усиление шума на изображении. Однако, за исключением шума, на рис. 3.26(*б*) не появилось никаких новых заметных деталей по сравнению с оригиналом, разве что очень слабый намек, что левый верхний и правый нижний квадраты

а *б* *в*

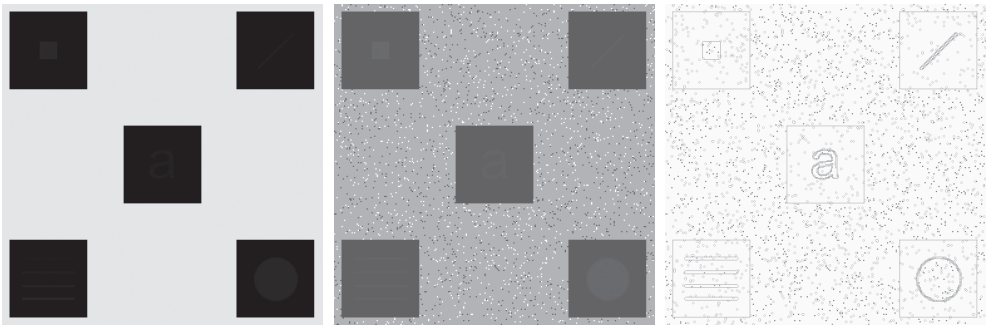


Рис. 3.26. (*а*) Исходное изображение. (*б*) Результат глобальной эквализации гистограммы. (*в*) Результат локальной эквализации гистограммы изображения (*а*) по окрестности 3×3 элементов вокруг каждого пикселя

[7] При этом одна и та же функция преобразования, построенная на основе анализа области, применяется ко всем пикселям данной области. — *Прим. перев.*

содержат какие-то объекты. Изображение на рис. 3.26(*в*) было получено использованием эквализации локальной гистограммы по окрестности размерами 3×3 пикселей. Здесь стали видны существенные детали, содержащиеся внутри черных квадратов. Значения яркостей этих объектов были слишком близки к яркостям окружающих квадратов, а размер их слишком мал, и поэтому они не могли достаточно заметно повлиять на глобальную гистограмму, чтобы можно было увидеть эти детали. ■

3.3.4. Использование гистограммных статистик для улучшения изображения

Статистики, получаемые непосредственного из гистограммы изображения, могут быть использованы для улучшения изображения. Пусть r обозначает дискретную случайную величину, представляющую квантованное значение яркости в диапазоне $[0, L-1]$, и пусть $p(r_i)$ — значение нормированной гистограммы, соответствующее значению r_i. Как уже говорилось ранее, $p(r_i)$ может рассматриваться как оценка вероятности того, что уровень яркости r_i появится на изображении, по которому посчитана гистограммы.

Как уже обсуждалось в разделе 2.6.8, центральный момент порядка n случайной величины r есть

$$\mu_n(r) = \sum_{i=0}^{L-1} (r_i - m)^n p(r_i), \tag{3.3-17}$$

где m — значение математического ожидания r (т. е. средний уровень яркости пикселей изображения):

$$m = \sum_{i=0}^{L-1} r_i p(r_i). \tag{3.3-18}$$

> Используя символ m для обозначения среднего, мы следуем традиции. Не следует путать его с тем же символом, используемым для обозначения числа строк в окрестности $m \times n$, где мы также следуем традиционной нотации.

Особенно важен второй момент:

$$\mu_2(r) = \sum_{i=0}^{L-1} (r_i - m)^2 p(r_i). \tag{3.3-19}$$

Это выражение есть не что иное, как дисперсия значений яркости r, которая обычно обозначается $\sigma^2(r)$. (Напомним, что стандартное отклонение σ есть квадратный корень из дисперсии.) Тогда как математическое ожидание является мерой средней яркости, дисперсия (или стандартное отклонение) является мерой контраста на изображении. Заметим, что все моменты легко вычисляются с помощью приведенных выше выражений по гистограмме изображения.

Когда требуются лишь среднее и дисперсия, часто их оценивают непосредственно по значениям отсчетов, без вычисления гистограммы. Эти оценки называются *выборочное среднее* и *выборочная дисперсия* соответственно. Они задаются следующими выражениями, хорошо знакомыми из основ статистики:

$$m = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y) \tag{3.3-20}$$

и

$$\sigma^2 = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[f(x,y)-m\right]^2 \tag{3.3-21}$$

для $x = 0, 1, 2, ..., M-1$ и $y = 0, 1, 2, ..., N-1$. Другими словами, как известно, средняя яркость изображения может быть получена просто суммированием значений всех пикселей и делением полученной суммы на общее число пикселей на изображении. Подобная интерпретация применима и к уравнению (3.3-21). Как показано в следующем примере, результаты, получаемые использованием этих двух уравнений, идентичны результатам, получаемым при помощи уравнений (3.3-18) и (3.3-19) при условии, что гистограмма вычисляется по тому же самому изображению.

Иногда в знаменателе уравнения (3.3-21) вместо MN пишут $MN - 1$. Это делается для того, чтобы получить так называемую несмещенную оценку дисперсии. Однако нас больше интересует соответствие уравнений (3.3-21) и (3.3-19), когда гистограмма в последнем уравнении подсчитывается по тому же самому изображению, что и в уравнении (3.3-21). Для этого нам требуется термин MN. Для изображения любого существенного для практики размера разница ничтожна.

Пример 3.11. Вычисление гистограммных статистик.

■ Прежде чем перейти далее, будет полезно проработать простой численный пример, чтобы закрепить материал. Рассмотрим следующее двухбитовое изображение размерами 5×5:

0	0	1	1	2
1	2	3	0	1
3	3	2	2	0
2	3	1	0	0
1	1	3	2	2

Значения пикселей представлены двумя битами; таким образом, $L = 4$, а значения яркостей находятся в диапазоне [0, 3]. Общее число пикселей равно 25, так что гистограмма содержит следующие компоненты:

$$p(r_0) = \frac{6}{25} = 0{,}24; \qquad p(r_1) = \frac{7}{25} = 0{,}28;$$

$$p(r_2) = \frac{7}{25} = 0{,}28; \qquad p(r_3) = \frac{5}{25} = 0{,}20,$$

где числитель в $p(r_i)$ есть число пикселей на изображении с яркостным уровнем r_i. Можно вычислить среднее значение яркостей элементов изображения, используя уравнение (3.3-18):

$$m = \sum_{i=0}^{3} r_i p(r_i) = 0\cdot 0{,}24 + 1\cdot 0{,}28 + 2\cdot 0{,}28 + 3\cdot 0{,}20 = 1{,}44.$$

Обозначая $f(x, y)$ тот же массив 5×5 элементов и используя уравнение (3.3-20), получим:

$$m = \frac{1}{25}\sum_{x=0}^{4}\sum_{y=0}^{4} f(x, y) = 1{,}44\,.$$

Как и ожидалось, результаты совпадают. Аналогично результаты для дисперсии равны 1,1264 как при использовании (3.3-19), так и (3.3-21). ■

В задаче улучшения используются, как правило, два типа измерения среднего и дисперсии. *Глобальные* среднее и дисперсия вычисляются по всему изображению и обычно применяются для общего выравнивания интенсивности и контраста изображения в целом. Более эффективным является применение этих двух параметров в методах локального улучшения, где *локальные* среднее и дисперсия используются как основа для проведения изменений, зависящих от характеристик окрестности вокруг каждого элемента изображения.

Пусть (x, y) — координаты некоторого элемента изображения, а S_{xy} означает окрестность (подмножество изображения) заданных размеров с центром в (x, y). Среднее значение пикселей в этой окрестности может быть вычислено по формуле

$$m_{S_{xy}} = \sum_{i=0}^{L-1} r_i p_{S_{xy}}(r_i)\,, \tag{3.3-22}$$

где $p_{S_{xy}}$ — значение гистограммы по окрестности S_{xy}. Эта гистограмма содержит L компонент, соответствующих L возможным значениям яркостей исходного изображения. Однако многие компоненты равны 0, что зависит от размера S_{xy}. Например, если окрестность имеет размеры 3×3 элементов и $L = 256$, только от 1 до 9 компонентов гистограммы по окрестности из 256 будут ненулевыми. Число ненулевых значений будет соответствовать числу различающихся яркостей в S_{xy} (максимальное число различающихся яркостей в окрестности 3×3 равно 9, а минимальное равно 1).

Аналогичным образом дисперсия яркостей пикселей в окрестности равна

$$\sigma^2_{S_{xy}} = \sum_{i=0}^{L-1} (r_i - m_{S_{xy}})^2 p_{S_{xy}}(r_i)\,. \tag{3.3-23}$$

Как и ранее, локальное среднее есть показатель средней яркости по окрестности S_{xy}, а дисперсия (или стандартное отклонение) есть показатель контраста по той же окрестности. Для окрестностей могут быть написаны уравнения, аналогичные (3.3-20) и (3.3-21). Просто при суммировании следует использовать значения пикселей окрестности, а в знаменателе — общее число пикселей в окрестности.

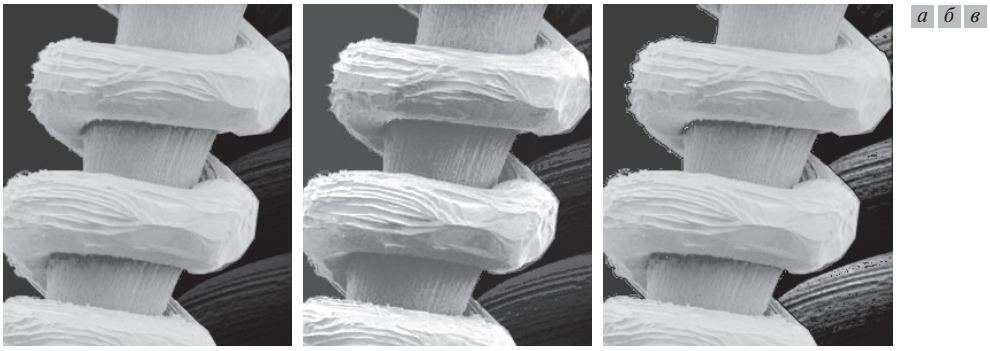
Как показано в следующем примере, важным аспектом обработки изображений при использовании значений локальных среднего и дисперсии является гибкость, которая позволяет разрабатывать простые, но мощные методы улучшения, основанные на статистических измерениях, имеющих близкое и предсказуемое соответствие виду самого изображения.

Пример 3.12. Локальное улучшение, основанное на гистограммных статистиках.

■ На рисунке 3.27(*а*) представлен снимок, полученный сканирующим электронным микроскопом, на котором изображена вольфрамовая нить накала, намотанная на держатель. И нить, и держатель достаточно хорошо видны. Однако в правой, темной части снимка имеется еще одна нить, которая почти неразличима, но ее размеры, а также остальные детали не так легко разглядеть. Локальное улучшение с изменением контраста является идеальным подходом для решения подобных задач, в которых отдельные части изображения могут содержать скрытые интересующие детали.

В данном частном случае задача состоит в улучшении темных областей при сохранении светлых областей по возможности неизменными, поскольку они не требуют улучшения. Можно использовать концепции, изложенные в данном разделе, чтобы построить метод улучшения, позволяющий различить светлые и темные участки, а затем улучшить только темные области. Критерий того, является ли область, к которой относится точка (x, y), темной или светлой, основан на сравнении локальной средней яркости $m_{S_{xy}}$ со средним значением яркости на изображении, называемым *глобальным средним* и обозначаемым m_G. Это последнее значение может быть получено с помощью уравнений (3.3-18) или (3.3-18), используя все изображение. Таким образом, возникает первый элемент схемы улучшения: пиксель в точке (x, y) является кандидатом на обработку, если $m_{S_{xy}} \le k_0 m_G$ (здесь $k_0 < 1{,}0$).

Поскольку интерес представляет улучшение областей, имеющих низкий контраст, то также необходим критерий для оценки контраста, с тем, чтобы отнести область к кандидатам на улучшение и по данному признаку. Таким образом, пиксель в точке (x, y) является кандидатом на улучшение, если $\sigma_{S_{xy}} \le k_2 \sigma_G$, где σ_G есть *глобальное стандартное отклонение*, полученное с помощью уравнения (3.3-19) или (3.3-21), а k_2 — положительная константа. Значение этой константы должно быть больше единицы, если требуется улучшение светлых областей, и меньше единицы для темных областей.



а б в

Рис. 3.27. (*а*) Снимок вольфрамовой нити накала, полученный сканирующим электронным микроскопом; увеличение приблизительно 130×. (*б*) Результат глобальной эквализации гистограммы. (*в*) Улучшение изображения при использовании статистик локальной гистограммы. (Исходное изображение предоставил Майкл Шаффер, факультет геологических наук Орегонского университета, Юджин)

Наконец, необходимо задать нижний уровень контраста для обрабатываемых областей, иначе процедура будет пытаться улучшать даже области постоянной яркости, на которых стандартное отклонение равно нулю. Тем самым задается также нижний предел для локального стандартного отклонения k_1, такой, чтобы $k_1\sigma_G \le \sigma_{S_{xy}}$, причем $k_1 < k_2$. Если пиксель расположен в точке (x, y), которая удовлетворяет всем вышеперечисленным условиям, необходимым для локального улучшения, то его обработка сводится к умножению значения пикселя на заданную константу E. При этом его яркость возрастает (или уменьшается) относительно остальной части изображения. Значения остальных пикселей, не удовлетворяющих условиям улучшения, остаются без изменения.

Рассмотренный выше подход можно подытожить следующим образом. Пусть $f(x, y)$ есть значение пикселя в точке (x, y), и пусть $g(x, y)$ есть значение того же пикселя, полученное в результате улучшения. Тогда

$$g(x,y) = \begin{cases} E \cdot f(x,y) & \text{если } m_{S_{xy}} \le k_0 m_G \text{ и } k_1\sigma_G \le \sigma_{S_{xy}} \le k_2\sigma_G \\ f(x,y) & \text{в противном случае} \end{cases} \qquad (3.3\text{-}24)$$

для $x = 0, 1, 2, ..., M - 1$ и $y = 0, 1, 2, ..., N - 1$, где, как указывалось ранее, E, k_0, k_1 и k_2 являются заданными параметрами; m_G — глобальное среднее, а σ_G — глобальное стандартное отклонение. Параметры $m_{S_{xy}}$ и $\sigma_{S_{xy}}$ суть локальное среднее и локальное стандартное отклонение соответственно. Как обычно, M и N — число строк и столбцов изображения, т. е. его размеры.

Как правило, подбор подходящих параметров в уравнении (3.3-24) требует определенного количества экспериментов, чтобы «почувствовать» конкретное изображение или класс изображений. В качестве рекомендуемых значений можно указать следующие: $E = 4{,}0$, $k_0 = 0{,}4$, $k_1 = 0{,}02$ и $k_2 = 0{,}4$. Относительно низкое значение 4,0 для E было выбрано по следующей причине. Когда E умножается на значение яркости элемента (который является темным, поскольку $f(x, y)$ мало), то результат $E \cdot f(x, y)$ по-прежнему должен быть темным, чтобы сохранить визуальный баланс яркостей на изображении. Значение k_0 было выбрано равным 0,4, поскольку очевидно, что темные области, требующие улучшения, определенно должны быть темнее, чем половина значения глобального среднего. Похожие рассуждения и анализ позволяют выбрать значения для параметров k_1 и k_2. Подбор всех этих констант в целом не является трудным, но их выбор определенно должен быть предварен логическим анализом имеющейся проблемы улучшения. Наконец, размеры локальной окрестности S_{xy} должны выбираться как можно меньшими для того, чтобы сохранить детали и уменьшить объем вычислений. В приведенных примерах мы выбирали локальную окрестность 3×3 пикселя.

В качестве основы для сравнения на рис. 3.27(*б*) приведен результат обработки изображения с помощью глобальной эквализации гистограммы. Темные области несколько улучшились, однако детали все еще трудно разглядеть; изменились также и светлые области, что иногда является нежелательным. На рис. 3.27(*в*) представлен результат применения метода локальных статистик, рассмотренный выше. Сравнивая это изображение с оригиналом на рис. 3.27(*а*) или с результатом эквализации гистограммы на рис. 3.27(*б*), легко заметить очевидные детали в правой части рис. 3.27(*в*), как бы вытащенные из темноты. Отметим, например, четкость контуров темной нити накаливания. Заслуживает

внимания тот факт, что яркие области в левой части остались почти неизменными, что являлось одним из первоначальных требований. ■

3.4. Основы пространственной фильтрации

В настоящем разделе выносятся на рассмотрение несколько основных концепций, лежащих в основе пространственной фильтрации в обработке изображений. Пространственная фильтрация является одним из основных инструментов, используемых в данной области для широкого спектра приложений, поэтому крайне рекомендуется выработать твердое понимание этих концепций. Как отмечалось в начале этой главы, приводимые здесь примеры в основном имеют отношение к применению пространственных фильтров для улучшения изображений. Другие приложения пространственной фильтрации рассматриваются в дальнейших главах.

Термин *фильтр* заимствован из области частотной фильтрации, которая есть тема следующей главы, где «фильтрация» относится к принятию (пропусканию) или подавлению определенных частотных компонент. Например, фильтр, который пропускает нижние частоты, называется *низкочастотным* фильтром. Цель, на которую направлен эффект, производимый низкочастотным фильтром, заключается в размывании (сглаживании) изображения. Подобное сглаживание можно выполнить прямо по изображению путем использования пространственных фильтров (также называемых пространственными *масками*, *ядрами*, *шаблонами* или *окнами*)[8]. Фактически, как будет показано в главе 4, имеется взаимно однозначное соответствие между линейными пространственными фильтрами и фильтрами в частотной области. Однако пространственные фильтры предлагают значительно большую гибкость, поскольку, как будет видно далее, они могут быть использованы также для нелинейной фильтрации, что не может быть реализовано в частотной области.

См. раздел 2.6.2 касательно линейности.

3.4.1. Механизмы пространственной фильтрации

На рис. 3.1 мы кратко обрисовали, что пространственный фильтр состоит из (1) *окрестности* (обычно небольшого прямоугольника) и (2) *заданной операции*, которая выполняется над пикселями изображения, попадающими в окрестность. Фильтрация создает новый пиксель, значение которого зависит от оператора фильтра, а координаты совпадают с координатами центра окрестности.[9]

---

[8] Также часто используется термин *оператор*. — *Прим. перев.*

[9] Полученное на выходе фильтра значение обычно присваивается пикселю в соответствующем месте нового изображения, создаваемого как выход результата фильтрации изображения. В редких случаях значение на выходе фильтра замещает первоначальное значение пикселя в исходном изображении; в такой ситуации новое значение будет менять содержание изображения, все еще находящегося в процессе обработки.

Обработанное (фильтрованное) изображение возникает в процессе сканирования исходного изображения фильтром. Если оператор, выполняемый над пикселями исходного изображения, является линейным, то фильтр называют *линейным пространственным фильтром*. В противном случае фильтр является *нелинейным*. Первоначально внимание будет сосредоточено на линейных фильтрах, а затем будут проиллюстрированы некоторые простые нелинейные фильтры. Раздел 5.3 содержит более полный список нелинейных фильтров и их приложений.

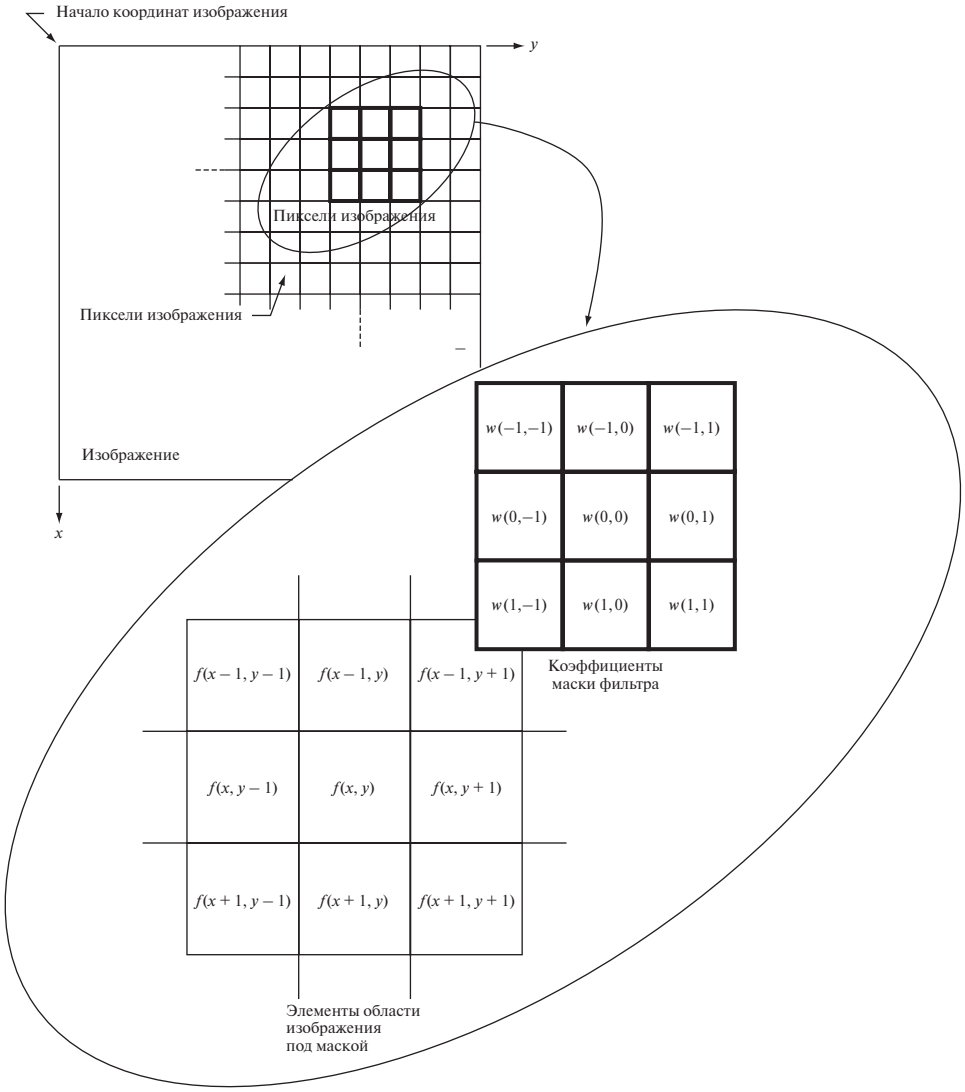


Рис. 3.28. Схема линейной пространственной фильтрации при использовании маски фильтра 3×3. Выбранная форма обозначения координат коэффициентов маски фильтра упрощает запись выражений для линейной фильтрации

На рис. 3.28 иллюстрируется схема линейной пространственной фильтрации с использованием окрестности 3×3 пикселей. В каждой точке (x, y) изображения отклик $g(x, y)$ задается суммой произведений коэффициентов фильтра на соответствующие значения пикселей в области, покрытой маской фильтра:

$$g(x,y) = w(-1,-1)f(x-1,y-1) + w(-1,0)f(x-1,y) + \ldots$$
$$+ w(0,0)f(x,y) + \ldots + w(1,0)f(x+1,y) + w(1,1)f(x+1,y+1).$$

Заметим, что центральный коэффициент $w(0, 0)$ стоит при значении пикселя в точке (x, y). В случае маски размерами $m \times n$ будем полагать, что $m = 2a + 1$ и $n = 2b + 1$, где a и b суть неотрицательные целые. Это означает, что в дальнейшем будут рассматриваться маски нечетных размеров, причем наименьшей будет маска 3×3 элемента. В общем виде фильтрация изображения размерами $M \times N$ с помощью фильтра размерами $m \times n$ задается выражением

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t),$$

где x и y изменяются так, что каждая точка оператора w в какой-то момент попадает на каждый пиксель изображения f.

В принципе, возможно работать и с окрестностями четного размера и даже смесью четного и нечетного размеров. Однако работа с нечетными размерами упрощает индексирование и является интуитивно более естественной, поскольку центр в таких фильтрах имеет целочисленные координаты.

3.4.2. Пространственная корреляция и свертка

Существуют две близкородственные концепции, которые при выполнении линейной пространственной фильтрации необходимо четко понимать. Одна из них — *корреляция*, а другая — *свертка*. Корреляция есть процесс движения маски фильтра по изображению и вычисление суммы произведений значений элементов маски и значений пикселей, на которые попадают соответствующие элементы маски, для всех точек изображения; в точности, как объяснялось в предыдущем разделе. Механизмы свертки такие же, но за исключением того, что предварительно маска фильтра поворачивается на 180°. Лучшим способом объяснить различия между двумя концепциями является пример. Начнем с одномерной иллюстрации.

На рис. 3.29(*а*) показана одномерная функция f вместе с некоторым фильтром w, а рис. 3.29(*б*) показывает начальную позицию для выполнения корреляции. Первое, что мы замечаем, — это что имеются участки функций, которые не перекрываются. Решение этой проблемы состоит в расширении f добавлением нулей с обеих сторон в количестве, достаточном для перекрытия с w. Если фильтр имеет размер m, то необходимо добавить $m - 1$ нулей с каждой стороны f. На рис. 3.29(*в*) показана корректно расширенная функция. Первым значением корреляции является сумма произведений элементов f и w для начальной позиции, показанной на том же рисунке (сумма произведений равна 0). Этот результат соответствует смещению $x = 0$. Чтобы получить второе значе-

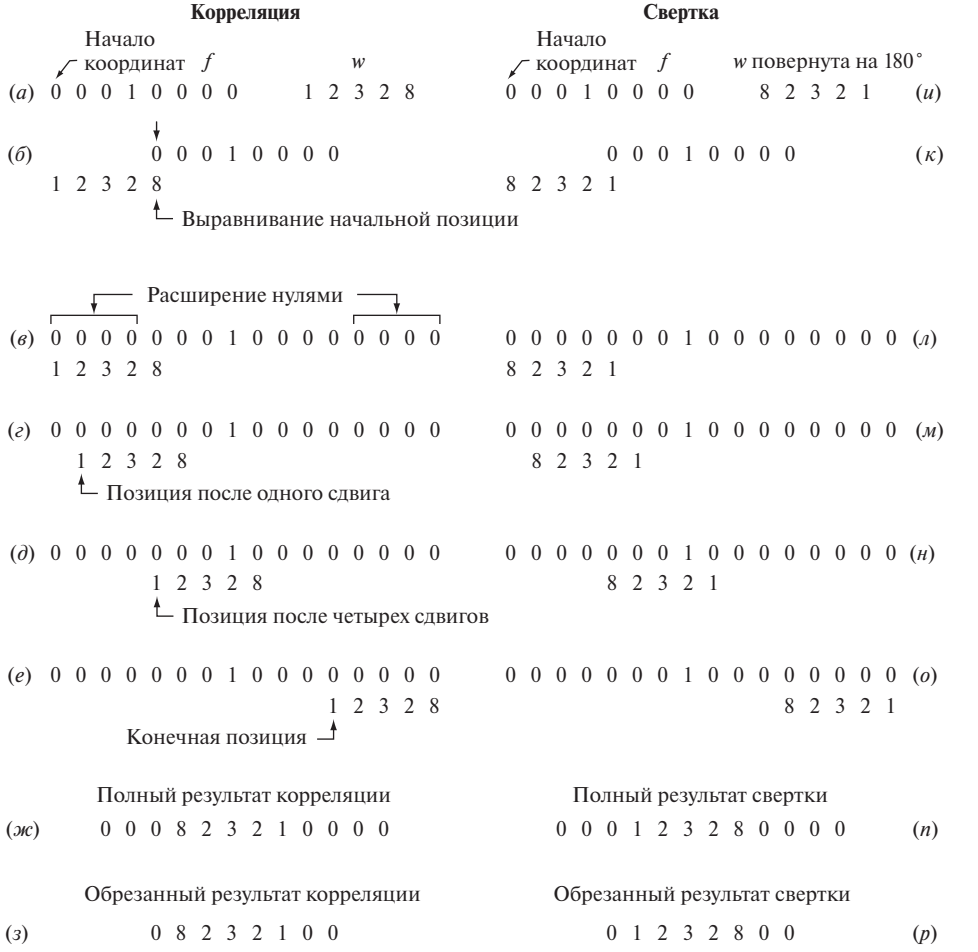


Рис. 3.29. Иллюстрация одномерной корреляции и свертки фильтра с дискретным единичным импульсом. Заметим, что и корреляция, и свертка являются функциями *смещения*

ние корреляции, w сдвигается вправо на один пиксель (смещение $x = 1$), и опять вычисляется сумма произведений. Результат опять равен 0. В действительности первый ненулевой результат появится при $x = 3$, когда восьмерка в w наползет на единичку в f и результат корреляции будет равен 8. Продолжая в таком духе, мы получим полный результат корреляции, показанный на рис. 3.29(*ж*). Заметим, что в итоге получилось 12 значений x (т.е. $x = 0, 1, 2, ..., 11$) для полного сдвига w по f, так что каждый элемент w соответствовал каждому пикселю f. Часто удобно работать с корреляционными массивами, имеющими тот же размер, что и f, и в таком случае приходится обрезать полную корреляцию до размеров исходной функции, как это показано на рис. 3.29(*з*).

Из предыдущего параграфа следует отметить два важных факта. Во-первых, корреляция есть функция *смещения* фильтра. Другими словами, значение корреляции в нулевой позиции соответствует нулевому смещению, значение в первой позиции — единичному смещению и так далее. Второй факт, который сле-

дует отметить, — это то, что корреляция фильтра w с функцией, содержащей единицу в одной точке и нули во всех остальных, приводит к результату, являющемуся копией w, но повернутому на 180°. Такую функцию называют *дискретным единичным импульсом*. Итак, можно заключить, что корреляция некоторой функции с дискретным единичным импульсом приводит к зеркальному повороту функции на 180° в месте положения импульса.

Расширение нулями не является единственным возможным вариантом. Так, например, можно $m - 1$ раз дублировать значения первого и последнего элементов f, или зеркально перевернуть $m - 1$ первых и последних элементов f и использовать их для расширения.

Концепция *свертки* является краеугольным камнем теории линейных систем. Как будет обсуждаться в главе 4, фундаментальным свойством свертки является то, что свертка функции с единичным импульсом дает в результате копию функции в месте импульса. В предыдущем параграфе было показано, что корреляция также дает в результате копию функции, однако повернутую на 180°. Таким образом, если предварительно повернуть фильтр и затем выполнить скользящую сумму операций умножения, то в результате мы должны получить желаемый результат. Как видно из правого столбца на рис. 3.29, это и в самом деле так. Таким образом, видно, что для выполнения свертки достаточно перевернуть одну из функций на 180° и выполнить те же операции, что и в случае корреляции. Оказывается, не имеет значения, которую из двух функций переворачивать.

Заметим, что поворот функции на 180° эквивалентен ее горизонтальному перевороту.

Рассмотренные выше принципы легко расширяются на изображения, что иллюстрируется на рис. 3.30. В случае фильтра размерами $m \times n$ изображение расширяется минимум на $m - 1$ строку нулей сверху и снизу, а также на $n - 1$ столбец нулей слева и справа. В данном случае m и n равны 3, так что f расширяется на две строки нулей сверху и снизу и на два столбца нулей справа и слева, как это видно на рис. 3.30(*б*). На рис. 3.30(*в*) показано начальное положение маски фильтра для выполнения корреляции. На рис. 3.30(*г*) представлен полный результат корреляции, а на рис. 3.30(*д*) — соответствующий обрезанный результат. Еще раз заметим, что в результате получили повернутые на 180° значения элементов фильтра. В случае свертки требуется предварительно, как и ранее, повернуть маску и затем выполнить скользящее суммирование произведений. Результат представлен на рис. 3.30(*е*)—(*з*). Еще раз видно, что свертка функции с единичным импульсом копирует функцию в месте импульса. Ясно, что если маска фильтра является симметричной, то и корреляция, и свертка дадут одинаковый результат.

В двумерном случае поворот на 180° эквивалентен перевороту маски сначала по одной оси, а затем по другой.

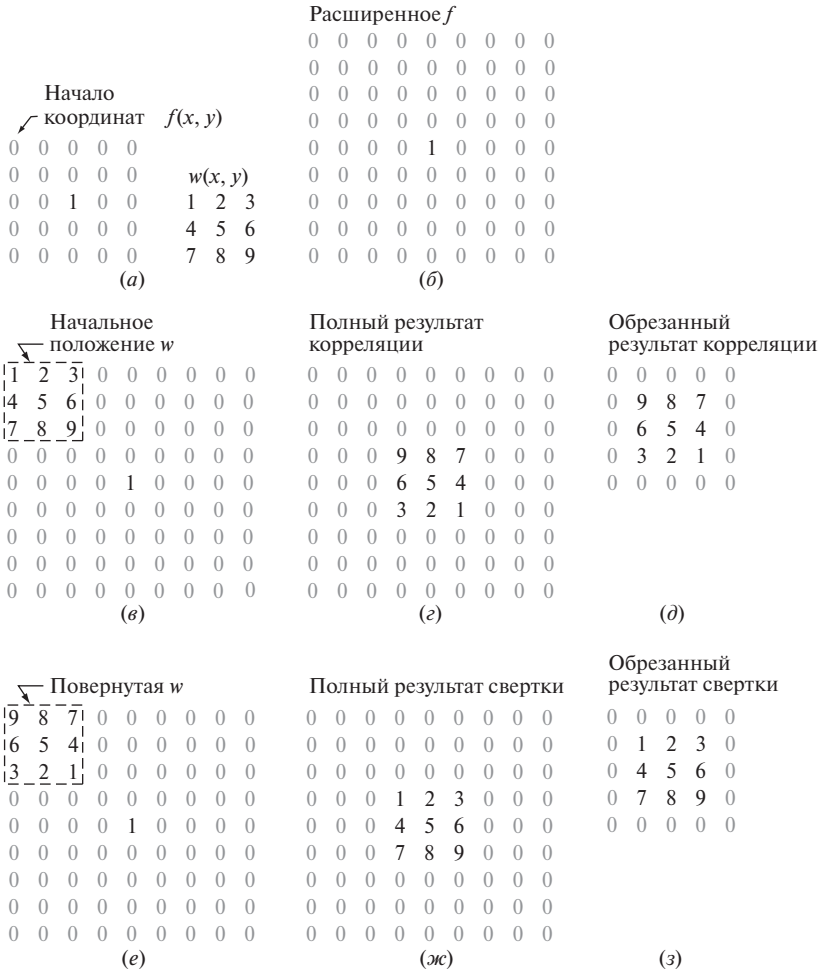


Рис. 3.30. Корреляция (средний ряд) и свертка (нижний ряд) двумерного фильтра с двумерным единичным импульсом. Нули показаны серым для удобства визуального анализа

Подводя итог обсуждению, сформулируем сказанное в аналитическом виде. Корреляция фильтра $w(x,y)$ размерами $m \times n$ с изображением $f(x,y)$, обозначаемая как $w(x,y) \star f(x,y)$, задается уравнением, записанным в конце предыдущего раздела, которое повторим здесь для удобства:

$$w(x,y) \star f(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t). \qquad (3.4\text{-}1)$$

Это уравнение вычисляется для всех значений переменных смещения x и y, так что все элементы из w попадают на каждый из пикселей f, при этом предполагается, что f расширено соответствующим образом. Как объяснено ранее, $a = (m-1)/2$, $b = (n-1)/2$ и для удобства обозначений предполагается, что m и n — целые нечетные.

Подобным же образом свертка $w(x, y)$ с $f(x, y)$, обозначаемая как $w(x, y) \star f(x, y)$[10], задается уравнением

$$w(x,y)\star f(x,y)=\sum_{s=-a}^{a}\sum_{t=-b}^{b}w(s,t)f(x-s,y-t), \qquad (3.4\text{-}2)$$

где знак минус действует как смена порядка на обратный (т. е. поворот на 180°). Переворот и сдвиг f вместо w сделан для упрощения обозначений, а также следуя общепринятым правилам. Результат тот же самый. Как и в случае корреляции, это уравнение вычисляется для всех значений переменных смещения x и y, так что все элементы из w попадают на каждый из пикселей f, расширенного соответствующим образом. Вам следует раскрыть уравнение (3.4-2) для случая маски 3×3 и убедиться, что результат, получаемый с помощью данного уравнения, идентичен примеру на рис. 3.30. На практике часто приходится работать с алгоритмом, который реализует уравнение (3.4-1). Если необходимо выполнить операцию корреляции, следует на вход алгоритма подавать w; если же необходима свертка, то предварительно следует повернуть w на 180°. Обратный вариант будет в случае, если реализован алгоритм по уравнению (3.4-2).

> Достаточно часто, когда не возникает неясности, результат корреляции или свертки обозначают через $g(x, y)$, а не $w(x, y) \star f(x, y)$ или $w(x, y) \star f(x, y)$. Например, см. уравнение в конце предыдущего раздела или уравнение (3.5-1).

Как уже упоминалось ранее, свертка является краеугольным камнем теории линейных систем. Как будет показано в главе 4, свойство свертки функции с единичным импульсом копировать функцию в место расположения импульса играет центральную роль во многих важных выводах. Мы вернемся к свертке в главе 4 в контексте преобразования Фурье и теоремы о свертке. Однако, в отличие от уравнения (3.4-2), мы будем иметь дело со сверткой функций одинаковых размеров. Форма уравнения остается той же самой, но пределы суммирования становятся другими.

Использовать корреляцию или свертку для выполнения пространственной фильтрации — это вопрос предпочтения. Поскольку в реальности как формула (3.4-1), так и (3.4-2) могут быть использованы для выполнения функции другой, достаточно лишь перевернуть фильтр, важным становится то, что маска фильтра, используемая в данной задаче фильтрации, должна быть задана способом, соответствующим планируемой операции. В настоящей главе все результаты линейной пространственной фильтрации основываются на уравнении(3.4-1).

Наконец, мы обращаем внимание, что в литературе по обработке изображений встречаются термины *фильтр свертки*, *маска свертки* или *ядро свертки*. Как правило, эти термины используются для обозначения пространственного фильтра и не обязательно, чтобы этот фильтр использовался именно для свертки. Аналогично фраза «свертка маски с изображением» часто используется для обозначения скользящего процесса суммирования произведений, который мы только что рассмотрели, а не обязательно для дифференциации между корреля-

---

[10] Поскольку свертка является коммутативной операцией, имеем $w(x, y) \star f(x, y) = f(x, y) \star w(x, y)$. Это свойство не выполняется для корреляции, как можно видеть, например, поменяв местами функции на рис. 3.29(*a*).

цией и сверткой. Вернее, она используется для общего указания любой из этих двух операций. Эта неточность терминологии является частым источником путаницы.

3.4.3. Векторное представление линейной фильтрации

Когда интерес представляет характеристика отклика R некоторой маски (либо корреляции, либо свертки), иногда удобно записать сумму произведений в виде

$$R = w_1 z_1 + w_2 z_2 + \ldots + w_{mn} z_{mn} = \sum_{k=1}^{mn} w_k z_k = \mathbf{w}^T \mathbf{z}, \tag{3.4-3}$$

где множество w — коэффициенты фильтра размерами mn, а z — соответствующие значения яркостей пикселей изображения, покрываемых фильтром. Чтобы использовать это уравнение для свертки, мы просто поворачиваем маску на 180°, как объяснено в предыдущем разделе. Подразумевается, что уравнение (3.4-3) справедливо для конкретной пары координат (x, y). В следующем разделе будет видно, почему такая форма записи удобна для объяснения характеристик заданного линейного фильтра.

Обратитесь к разделу обучающих материалов на интернет-сайте книги для краткого обзора векторов и матриц.

В качестве примера на рис. 3.31 показана общая маска 3×3 с коэффициентами, обозначенными так же, как и выше. В этом случае уравнение (3.4-3) принимает вид

$$R = w_1 z_1 + w_2 z_2 + \ldots + w_9 z_9 = \sum_{k=1}^{9} w_k z_k = \mathbf{w}^T \mathbf{z}, \tag{3.4-4}$$

где **w** и **z** — 9-мерные вектора, сформированные из коэффициентов маски и, соответственно, значений яркостей пикселей изображения, покрываемых фильтром.

3.4.4. Формирование масок пространственных фильтров

Для формирования *линейного* пространственного фильтра размерами $m \times n$ требуется задать mn коэффициентов маски. И наоборот, выбор этих коэффициен-

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Рис. 3.31. Другой вариант представления часто используемой маски фильтра по окрестности 3×3 элемента

тов базируется на том, какие действия ожидаются от фильтра при условии, что в случае линейной фильтрации мы можем реализовать лишь сумму произведений. Предположим, например, что требуется заменить значение пикселя на изображении значением средней яркости по окрестности 3×3 с центром в данном пикселе. Среднее значение в любой точке (x, y) изображения есть сумма девяти значений яркости в окрестности 3×3 с центром в (x, y), деленная на 9. Полагая, что z_i, $i = 1, 2, ..., 9$, обозначает эти яркости, среднее будет равно

$$R = \frac{1}{9}\sum_{i=1}^{9} z_i .$$

Но эта формула совпадает с (3.4-4) при значениях коэффициентов $w_i = 1/9$. Другими словами, операция линейной фильтрации по маске 3×3, коэффициенты которой равны 1/9, выполняет желаемое усреднение. Как будет обсуждаться в следующем разделе, эта операция приводит к сглаживанию изображения. Далее будут обсуждаться многие другие маски фильтров, основанных на этом фундаментальном подходе.

В некоторых приложениях встречаются непрерывные функции двух переменных, и целью является получение маски пространственного фильтра, основанного на таких функциях. Например, гауссова функция двух переменных имеет основной вид

$$h(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}},$$

где σ означает стандартное отклонение и, как обычно, предполагается, что координаты x и y являются целыми. Чтобы сформировать из этой функции, скажем, маску фильтра 3×3, берутся значения этой функции вокруг ее центра. Так, $w_1 = h(-1, -1)$, $w_2 = h(-1, 0)$, ..., $w_9 = h(1, 1)$. Маска фильтра размерами $m \times n$ формируется аналогичным образом. Напомним, что двумерная гауссова функция имеет колоколообразную форму и что стандартное отклонение задает степень «растянутости» колокола.

Формирование *нелинейного* фильтра требует, чтобы был задан размер окрестности и операция (операции), выполняющаяся над значениями пикселей изображения, попадающими в окрестность. Например, напоминая, что операция выбора максимума является нелинейной (см. раздел 2.6.2), фильтр максимума по окрестности 5×5, центрированный в любой точке (x, y) изображения, дает в результате значение максимума из 25 пикселей и присваивает его пикселю обрабатываемого изображения в точке (x, y). Нелинейные фильтры являются весьма мощными средствами и в некоторых приложениях могут осуществлять операции, выходящие далеко за возможности линейных фильтров, как будет видно из последующих обсуждений в настоящей главе, а также в главе 5.

3.5. Сглаживающие пространственные фильтры

Сглаживающие фильтры применяются для расфокусировки изображения и подавления шума. Расфокусировка может применяться в задачах предварительной обработки изображения, например для удаления мелких деталей перед обнаружением больших объектов или же для устранения разрывов в линиях или

деталях. Для подавления шумов может использоваться расфокусировка с применением как линейной, так и нелинейной фильтрации.

3.5.1. Линейные сглаживающие фильтры

Выход (отклик) простейшего линейного сглаживающего пространственного фильтра есть среднее значение элементов по окрестности, покрытой маской фильтра. Такие фильтры иногда называют *усредняющими* или *сглаживающими фильтрами*. Как упоминалось в предыдущем разделе, их также называют *низкочастотными фильтрами*.

Идея применения сглаживающих фильтров достаточно ясна. Заменой исходных значений элементов изображения на средние значения по маске фильтра достигается уменьшение «резких» переходов уровней яркости. Поскольку случайный шум как раз характеризуется резкими скачками яркости, наиболее очевидным применением сглаживания является подавление шума. Однако контуры (которые обычно представляют интерес на изображении) также характеризуются резкими перепадами яркостей, поэтому негативной стороной применения сглаживающих фильтров является расфокусировка контуров. Другим применением такой процедуры может быть сглаживание ложных контуров, которые возникают при преобразованиях с недостаточным числом уровней яркости, как это обсуждалось в разделе 2.4.3. Главное использование сглаживающих фильтров состоит в подавлении «несущественных» деталей на изображении. Под «несущественными» здесь понимаются совокупности пикселей, которые малы по сравнению с размерами маски фильтра. Это последнее применение будет проиллюстрировано ниже.

На рис. 3.32 показаны два сглаживающих фильтра по окрестности 3×3. Первый из них дает обычное среднее значение по маске. Подстановкой коэффициентов маски в уравнение (3.4-4) получим

$$R = \frac{1}{9}\sum_{i=1}^{9} z_i,$$

что как раз и дает среднее значение яркостей по окрестности 3×3, как отмечалось ранее. Заметим, что коэффициенты фильтра указаны как единицы вместо 1/9. Причина в том, что такой вариант является более эффективным при ком-

а б

$\frac{1}{9} \times$

1	1	1
1	1	1
1	1	1

$\frac{1}{16} \times$

1	2	1
2	4	2
1	2	1

Рис. 3.32. Две маски сглаживающих (усредняющих) фильтров по окрестности 3×3. Постоянный множитель перед каждой из масок равен единице, деленной на сумму значений коэффициентов, как это необходимо для нормировки

пьютерных вычислениях. По окончании процесса суммирования полученное значение делится на 9. Маска размерами $m \times n$ будет иметь нормировочный коэффициент, равный $1/mn$. Такой пространственный фильтр, все коэффициенты которого одинаковы, иногда называют *однородным усредняющим фильтром*.

Вторая маска на рис. 3.32 несколько более интересна. Эта маска дает так называемое *взвешенное среднее*; этот термин применяется, чтобы показать, что значения элементов суммируются с разными коэффициентами, что позволяет присвоить им как бы разные «важности» (веса) по сравнению с другими. В маске на рис. 3.32(*б*) коэффициент в центре маски имеет самое большее значение (вес), тем самым давая соответствующему элементу бо́льшую важность при вычислении среднего. Значения остальных коэффициентов в маске уменьшаются по мере удаления от центра маски. Диагональные члены, по сравнению с ортогональными, расположены от центра дальше и, таким образом, «весят» меньше, чем ближайшие соседи центрального элемента. Основная стратегия присвоения центральному пикселю наибольшего веса, а остальным — обратно пропорционально их расстоянию, имеет целью уменьшение расфокусировки при сглаживании. Можно было бы выбрать и другие значения коэффициентов маски для достижения поставленной цели, но сумма коэффициентов, приведенных на рис. 3.32(*б*), равна 16, что удобно при компьютерной реализации, поскольку это степень двойки. Следует отметить, что на практике достаточно трудно увидеть разницу между изображениями, сглаженными фильтрами по одной из масок на рис. 3.32 или какими-то другими, аналогичными по конструкции, поскольку размеры области, покрываемые маской при фильтрации одного элемента, очень малы.

Как следует из уравнения (3.4-1), общая формула фильтрации изображения размерами $M \times N$ фильтром взвешенного среднего по окрестности $m \times n$ (m и n — нечетные) задается выражением

$$g(x,y) = \frac{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)}{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)}. \qquad (3.5\text{-}1)$$

Параметры в этом уравнении такие же, как были определены для уравнения (3.4-1). Как и ранее, подразумевается, что полная фильтрация изображения достигается применением формулы (3.5-1) ко всем парам $x = 0, 1, 2, ..., M-1$ и $y = 0, 1, 2, ..., N-1$. Знаменатель в (3.5-1) есть просто сумма всех коэффициентов маски, следовательно, является константой и требует вычисления лишь один раз.

Пример 3.13. Сглаживание изображения по маскам различных размеров.

■ Зависимость эффекта сглаживания от размера фильтра проиллюстрирована на рис. 3.33, где представлено исходное изображение и результаты сглаживания усредняющим фильтром с квадратной окрестностью размерами m = 3, 5, 9, 15, и 35 элементов. Главные результаты этого эксперимента таковы. При m = 3 можно заметить легкую расфокусировку на всем изображении, однако, как и следовало ожидать, более всего искажены мелкие детали, имеющие размеры, близкие к размерам фильтра. На изображениях, сглаженных по окрестностям 3×3 и 5×5, уже видна значительная расфокусировка на маленькой букве «а» и на мелкозернистом шуме. Заметно заглажены выступающие края букв, а также серые кру-

а б
в г
д е

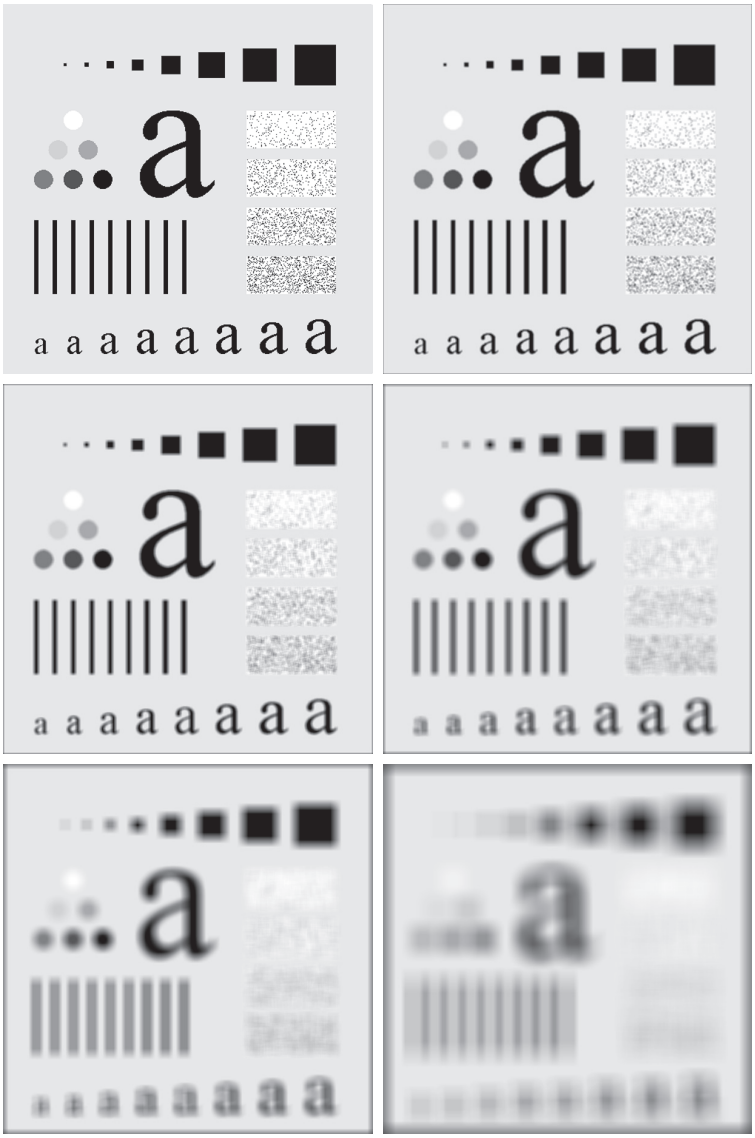


Рис. 3.33. (*а*) Исходное изображение размерами 500×500 элементов. (*б*)—(*е*) Результаты сглаживания усредняющим фильтром с квадратной маской размерами $m = 3, 5, 9, 15$ и 35. Черные квадратики в верхней части имеют размеры 3, 5, 9, 15, 25, 35, 45 и 55 пикселей соответственно и расположены на расстоянии в 25 пикселей друг от друга. Буквы внизу изменяются в размерах от 10 до 24 пунктов, с увеличением каждый раз на 2 пункта; размер самой большой буквы — 60 пунктов. Вертикальные полосы имеют 5 пикселей в ширину и 100 пикселей в высоту; расстояния между ними — 20 пикселей. Диаметр кружков — 25 пикселей, и расположены они на расстоянии 15 пикселей друг от друга; их уровни яркостей изменяются от 0 % до 100 % черного с шагом 20 %. Общий фон изображения — 10 % черного. Прямоугольники с шумом имеют размеры 50×120 пикселей

жочки. Заметим, что шум стал значительно менее выраженным, а зубцы букв стали сглаженными.

Небольшое увеличение размеров окрестности сглаживания (больше $m = 5$) не приводит к сильным изменениям результатов. Но при $m = 9$ можно видеть уже значительно более сильную расфокусировку и 20-процентный черный кружочек уже не так сильно отличается от фона, как на предыдущих трех изображениях, тем самым иллюстрируя эффект смешивания, являющийся результатом расфокусировки. Также видно значительно более сильное сглаживание зашумленных прямоугольников. Результаты с $m = 15$ и 35 следует считать предельными по отношению к размерам объектов на изображении. Такое чрезмерное сглаживание используется для удаления с изображения мелких объектов. Например, три маленьких квадрата, два круга и большинство из зашумленных прямоугольников на рис. 3.33(*е*) неотличимы от фона. На этом изображении стал сильно заметен темный бордюр, который является следствием расширения изображения перед операцией сглаживания нулями (т. е. черным фоном) и последующего обрезания краев после фильтрации. Этот эффект проявляется на всех изображениях, но наиболее заметен на изображении, сглаженном фильтром самого большого размера. ■

Как отмечалось выше, важным применением пространственного сглаживания является расфокусировка изображения, позволяющая создать грубый образ объектов, которые могут представлять интерес. При этом яркость мелких объектов смешивается с фоном, в то время как большие объекты остаются в виде пятен и могут быть легко обнаружены. Размеры объектов, которые будут смешиваться с фоном, приблизительно совпадают с размерами маски сглаживающего фильтра. В качестве примера рассмотрим изображение на рис. 3.34(*а*), полученное телескопом «Хаббл» с орбиты Земли. Результат применения усредняющего фильтра с маской 15×15 элементов показан на рис. 3.34(*б*). Можно видеть, что многие из объектов или оказались смешаны с фоном, или же их яркость значительно уменьшилась. Обычно после такого сглаживания следует опера-

а б в

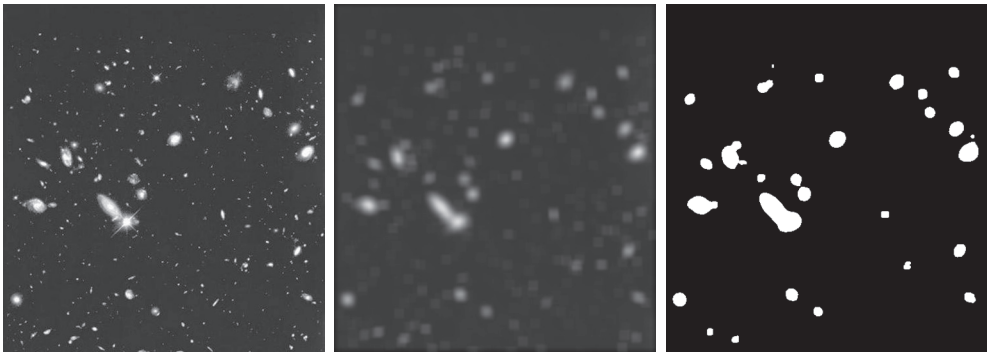


Рис. 3.34. (*а*) Изображение размерами 528×485 пикселей, полученное космическим телескопом «Хаббл». (*б*) Изображение, обработанное сглаживающей маской размерами 15×15 элементов. (*в*) Результат применения порогового обнаружения к изображению (*б*). (Исходное изображение предоставлено агентством NASA)

ция разделения по порогу, позволяющая убрать объекты малой интенсивности. Результат применения пороговой операции к изображению на рис. 3.2(*б*) с уровнем порога в 25 % от наибольшей яркости показан на рис. 3.34(*в*). Сравнивая это изображение с исходным, можно сделать вывод, что полученный результат был бы приемлемым с позиции поиска самых больших и ярких объектов.

3.5.2. Фильтры, основанные на порядковых статистиках (нелинейные фильтры)

Фильтры, основанные на порядковых статистиках, относятся к классу нелинейных пространственных фильтров. Отклик такого фильтра определяется предварительным упорядочиванием (ранжированием) значений пикселей, покрываемых маской фильтра, и последующим выбором значения, находящегося на определенной позиции упорядоченной последовательности (т. е. имеющего определенный ранг). Собственно фильтрация сводится к замещению исходного значения пикселя (в центре маски) на полученное значение отклика фильтра. Наиболее известен *медианный фильтр*, который, как следует из названия, заменяет значение пикселя на значение медианы распределения яркостей всех пикселей в окрестности (включая и исходный). Медианные фильтры весьма популярны потому, что для определенных типов случайных шумов они демонстрируют отличные возможности подавления шума при значительно меньшем эффекте расфокусировки, чем у линейных сглаживающих фильтров с аналогичными размерами. В частности, медианные фильтры эффективны при фильтрации *импульсных шумов*, иногда называемых *шумами «соль и перец»*, которые выглядят как наложение на изображение случайных черных и белых точек.

Медиана набора чисел есть такое число ξ, что половина чисел из набора меньше или равны ξ, а другая половина — больше или равны ξ. Чтобы выполнить медианную фильтрацию для элемента изображения, необходимо сначала упорядочить по возрастанию значения пикселей внутри окрестности, затем найти значение медианы и, наконец, присвоить полученное значение обрабатываемому элементу. Так, для окрестности 3×3 элементов медианой будет пятое значение по величине, для окрестности 5×5 — тринадцатое значение и так далее. Если несколько элементов в окрестности имеют одинаковые значения, эти значения будут сгруппированы. Например, пусть в окрестности 3×3 элементы имеют следующие значения: (10, 20, 20, 20, 15, 20, 20, 25, 100). После упорядочивания они будут расположены следующим образом: (10, 15, 20, 20, 20, 20, 20, 25, 100), а, следовательно, медианой будет значение 20. Можно сказать, что основная функция медианного фильтра заключается в замене отличающегося от фона значения пикселя на другое, более близкое к его соседям. На самом деле изолированные темные или светлые (по сравнению с окружающим фоном) кластеры, имеющие площадь не более чем $m^2/2$ (половина площади маски фильтра), будут удалены медианным фильтром с маской размерами $m \times m$. В данном случае «удалены» означает, что значения пикселей в соответствующих точках будут заменены на значения медиан по окрестностям. Кластеры больших размеров искажаются значительно меньше.

Хотя медианный фильтр значительно более распространен в обработке изображений, чем остальные виды фильтров, основанные на порядковых статистиках, тем не менее он не является единственным. Медиана представляет собой

а б в

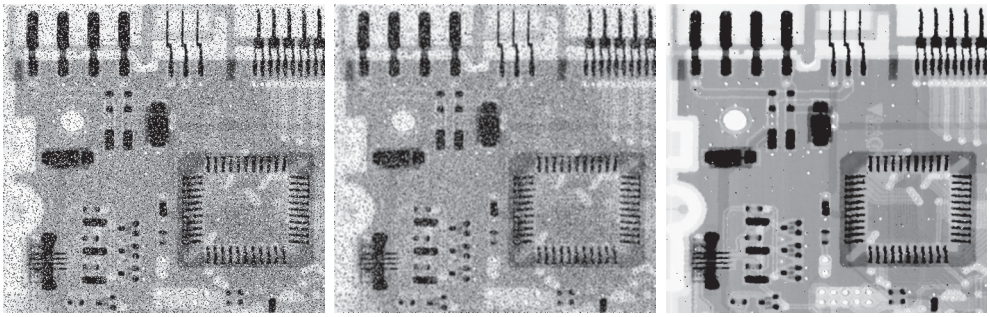


Рис. 3.35. (*а*) Рентгеновский снимок монтажной платы, искаженный импульсным шумом. (*б*) Подавление шума усредняющим фильтром по окрестности 3×3. (*в*) Подавление шума медианным фильтром по окрестности 3×3. (Исходное изображение предоставил Джозеф Пасенте, компания Lixi, Inc.)

квантиль уровня 0,5 упорядоченного набора чисел, но, как следует из основ статистики, упорядочивание предоставляет много других возможностей. Например, использование квантиля уровня 1 приводит к так называемому *фильтру максимума*, который полезен при поиске на изображении наиболее ярких точек по отношению к окружающему фону. Отклик фильтра максимума по окрестности 3×3 задается выражением $R = \max\{z_k | k = 1, 2, ..., 9\}$. Квантиль уровня 0 является *фильтром минимума*, используемым для поиска противоположных значений. Медианный фильтр, фильтры максимума и минимума, а также некоторые другие фильтры будут более детально рассмотрены в главе 5.

Касательно квантилей см. раздел 10.3.5.

Пример 3.14. Использование медианной фильтрации для подавления шума.

■ На рис. 3.35(*а*) представлен рентгеновский снимок монтажной платы, сильно искаженный импульсным шумом. Чтобы проиллюстрировать преимущество медианной фильтрации по сравнению с усредняющим фильтром, на рис. 3.35(*б*) показан результат обработки зашумленного изображения усредняющим фильтром по окрестности 3×3, а на рис. 3.35(*в*) — результат медианной фильтрации по окрестности 3×3. Усредняющий фильтр делает изображение нерезким, и его способность снижения шума невысока. При этом совершенно очевидно преимущество медианного фильтра во всех отношениях. Вообще медианная фильтрация намного больше подходит для удаления импульсного шума, чем усредняющая фильтрация. ■

3.6. Пространственные фильтры повышения резкости

Главная цель повышения резкости заключается в том, чтобы подчеркнуть яркостные переходы. Повышение резкости изображений используется достаточно ши-

роко — от цифровой печати и медицинской интроскопии до технического контроля в промышленности и систем автоматического наведения в военной сфере. В предыдущем разделе мы видели, что расфокусировка изображения может быть достигнута пространственной операцией усреднения значений точек по окрестности. Поскольку усреднение аналогично интегрированию, то логично прийти к выводу, что повышение резкости, будучи явлением, обратным по отношению к расфокусировке, может быть достигнуто пространственным дифференцированием. Это действительно так, и в настоящем разделе будут обсуждаться различные способы задания и использования операторов повышения резкости путем численного дифференцирования. С принципиальной точки зрения величина отклика оператора производной в точке изображения пропорциональна степени разрывности изображения в данной точке. Таким образом, дифференцирование изображения позволяет усилить перепады и другие разрывы (например шумы) и не подчеркивать области с медленными изменениями уровней яркостей.

3.6.1. Основы

В последующих двух разделах будут рассмотрены фильтры повышения резкости, основанные на первой и второй производных. Однако прежде чем перейти к непосредственному обсуждению, необходимо остановиться на некоторых фундаментальных свойствах этих производных в контексте цифровых методов. Для простоты изложения первоначально остановимся на одномерных производных. В частности, представляет интерес поведение этих производных на областях постоянной яркости, в начале и в конце разрывов (разрывы в виде ступенек и участков изменения яркости — склонов), а также на протяжении самих склонов. Как будет показано в главе 10, эти типы разрывов могут использоваться для описания шумовых всплесков, линий и контуров на изображении. Также важным является поведение производной на протяжении перехода от начала до окончания указанных особенностей.

Производные дискретной функции определяются в терминах разностей. Эти разности можно задать различными способами, однако мы будем руководствоваться следующим. *Первая производная* должна быть: (1) равной нулю на областях с постоянным уровнем яркости; (2) ненулевой в начале и в конце ступеньки или склона яркости; (3) ненулевой на склонах яркости. Аналогично *вторая производная* должна быть: (1) равной нулю на плоских участках; (2) ненулевой в начале и в конце ступеньки или склона яркости; (3) равной нулю на склонах постоянной крутизны. Так как мы оперируем ограниченными численными значениями, максимальное значение изменения яркости также конечно, а кратчайшее расстояние, на котором это изменение может происходить, есть расстояние между соседними пикселями.

Первая производная одномерной функции $f(x)$ определяется как разность значений соседних элементов:

$$\frac{\partial f}{\partial x} = f(x+1) - f(x). \tag{3.6-1}$$

Мы вернемся к уравнению (3.6-1) в разделе 10.2.1 и покажем, как оно следует из разложения в ряд Тейлора. На данный момент мы принимаем его как определение.

Здесь использована запись в виде частной производной для того, чтобы сохранить те же обозначения в случае двух переменных $f(x, y)$, где придется иметь дело с частными производными по двум пространственным осям. Использование частной производной не меняет существа рассмотрения. Ясно, что $\partial f/\partial x = \mathrm{d}f/dx$, когда у функции имеется только одна переменная; то же верно в случае второй производной.

Вторая производная определяется как разность соседних значений первой производной:

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x). \qquad (3.6\text{-}2)$$

Легко проверить, что оба данных определения удовлетворяют сформулированным ранее условиям касательно производных первого и второго порядков. Чтобы проиллюстрировать это, а также рассмотреть основные сходства и различия между производными первого и второго порядков в контексте обработки изображений, рассмотрим пример на рис. 3.36.

На рис. 3.36(*б*) (в центре рисунка) показан участок строки изображения (профиль яркости). Числа в квадратиках суть значения яркостей элементов строки изображения, которые отмечены черными точками на рис. 3.36(*а*). Для визуаль-

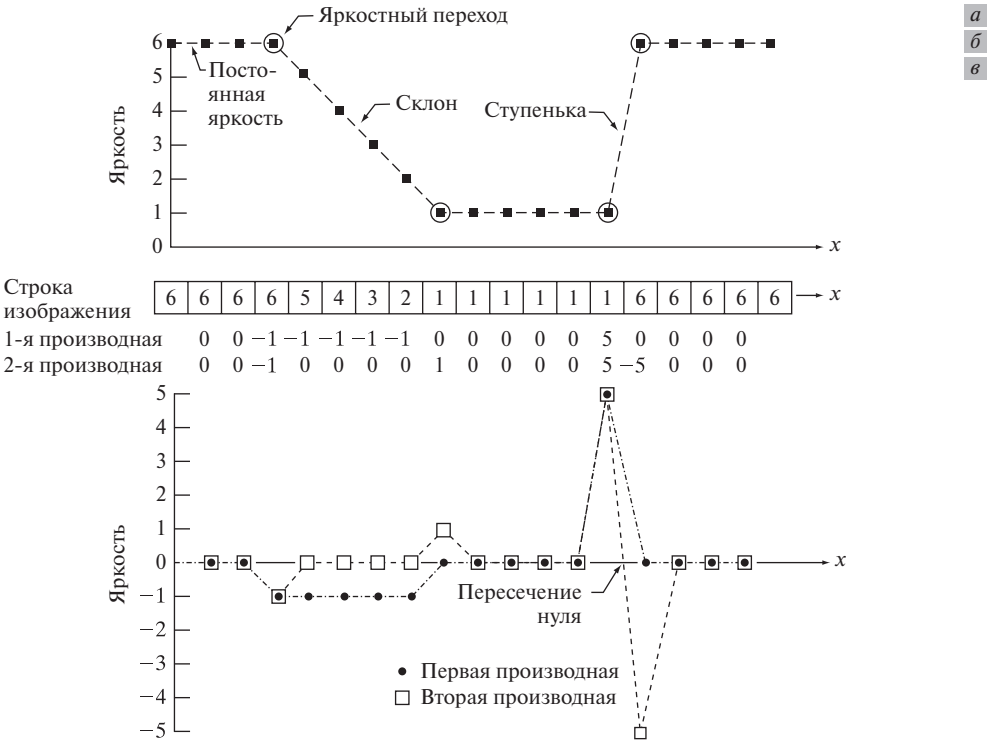


Рис. 3.36. Иллюстрация первой и второй производных одномерной дискретной функции, представляющей собой участок горизонтального профиля яркости некоторого изображения. На (*а*) и (*в*) точки отсчетов соединены пунктирными линиями для визуального удобства

ного удобства добавлена пунктирная линия, соединяющая эти точки. Как видно, на участке строки содержатся три участка постоянной яркости, участок склона и яркостная ступенька. Кружки показывают начало или конец яркостных переходов. Первая и вторая производные, вычисленные на основе данных выше двух определений, приведены ниже строки изображения на рис. 3.36(*б*), а также в виде графиков на рис. 3.36(*в*). При вычислении первой производной в точке *x* значение в данной точке вычиталось из значения в следующей точке, так что это операция с «предварительным просмотром». Аналогично для вычисления второй производной в точке *x* использовались значения в предыдущей и следующей точках. Чтобы отвлечься от вопроса выхода предыдущей или следующей точки за границы участка строки, значения производных на рис. 3.36 приведены лишь от второго до предпоследнего элемента.

Рассмотрим поведение первой и второй производных при движении вдоль профиля слева направо. В начале находится область с постоянной яркостью, на которой, как видно на рис. 3.36(*б*) и (*в*), обе производные равны нулю, значит, для обеих выполняется условие (1). Затем идет склон яркости, за которым следуют площадка и ступенька. Отметим, что первая производная отлична от нуля на склоне и на ступеньке; аналогично вторая производная отлична от нуля в начале и в конце как склона, так и ступеньки; таким образом, свойство (2) удовлетворяется для обеих производных. Наконец, свойство (3) также выполняется для обеих производных, поскольку на склоне первая производная не равна нулю, а вторая равна нулю. Заметим, что знак второй производной меняется в начале и в конце ступеньки или склона. Действительно, на рис. 3.36(*в*) видно, что линия, соединяющая два значения производных на участке ступенчатого перехода, в середине между экстремумами пересекает горизонтальную ось. Это свойство пересечения нулевого уровня весьма полезно для локализации контуров, как будет показано в главе 10.

На цифровых изображениях контурные переходы зачастую выглядят как склоны яркостей, на которых первая производная будет диагностировать в результате широкие контуры, поскольку она отлична от нуля на всем протяжении склона. Напротив, вторая производная даст двойной контур толщиной в одну линию и разделенный нулями. Отсюда можно сделать вывод, что вторая производная усиливает мелкие детали значительно лучше, чем первая, — свойство, идеально подходящее для повышения резкости изображений. Также, как будет показано далее в настоящем разделе, вторую производную значительно легче реализовать, чем первую; так что вначале наше внимание будет сконцентрировано на вторых производных.

3.6.2. Повышение резкости изображений с использованием вторых производных: лапласиан

В данном разделе мы рассмотрим применение двумерной второй производной в задачах повышения резкости изображений. Мы вернемся к рассмотрению второй производной в главе 10, где она будет использоваться для сегментации изображений. Подход сводится к выбору дискретной формулировки второй производной и к последующему построению маски фильтра, основанной на данной формулировке. Рассматриваться будут *изотропные* фильтры, отклик которых

не зависит от направления неоднородностей на обрабатываемом изображении. Другими словами, изотропные фильтры являются *инвариантными к повороту* в том смысле, что поворот изображения и последующее применение фильтра дает тот же результат, что и первоначальное применение фильтра с последующим поворотом результата[11].

Можно показать ([Rosenfeld и Kak, 1982]), что простейшим изотропным оператором, основанным на производных, является *лапласиан* (оператор Лапласа), который в случае функции двух переменных $f(x, y)$ определяется как

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}. \tag{3.6-3}$$

Поскольку производные любого порядка являются линейными операторами, то, значит, и лапласиан является линейным оператором. Чтобы выразить это уравнение в дискретной форме, используем определение, выраженное уравнением (3.6-2), принимая во внимание, что теперь имеются две переменные. Для частной второй производной по x будет использоваться следующая формула:

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y), \tag{3.6-4}$$

и аналогично для производной по y:

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y). \tag{3.6-5}$$

Таким образом, из предыдущих трех уравнений следует, что дискретная формулировка двумерного лапласиана двух переменных будет

$$\nabla^2 f(x, y) = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y). \tag{3.6-6}$$

Это уравнение может быть реализовано с помощью маски фильтра, представленной на рис. 3.37(*а*), которая дает изотропный результат для поворотов на углы, кратные 90°. Способы реализации такие же, как в разделе (3.5.1) для линейных сглаживающих фильтров. Здесь попросту используются другие значения коэффициентов.

Диагональные направления могут быть включены в формулу дискретного лапласиана (3.6-6) добавлением еще двух членов — по одному для каждого из диагональных направлений. Вид каждого из них такой же, как в уравнении (3.6-4) или (3.6-5), но указываются координаты точек, расположенных по диагоналям. Поскольку каждая диагональная добавка включает член $-2f(x, y)$, то суммарный вычитаемый из суммы член составит $-8f(x, y)$. Маска фильтра, соответствующая такому новому определению, представлена на рис. 3.37(*б*). Такая маска является изотропной для поворотов на углы, кратные 45°. Две оставшиеся маски, показанные на рис. 3.37(*в*) и (*г*), также часто используются на практике. Они получены из определения лапласиана, являющегося «негативным» по от-

[11] В применении к дискретному сигналу это утверждение верно лишь настолько, насколько точными в его отношении можно считать операцию поворота и круговую симметрию масок фильтров. — *Прим. перев.*

а *б*
в *г*

0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

Рис. 3.37. (*а*) Маска фильтра, используемая для реализации уравнения (3.6-6). (*б*) Маска, используемая для реализации расширения этого уравнения путем добавления диагональных членов. (*в*) и (*г*) Две другие реализации лапласиана, часто встречаемые на практике

ношению к тому, который использовался в уравнениях (3.6-4) и (3.6-5). По существу, они дают идентичный результат, но различие в знаке должно учитываться при комбинации — операцией сложения или вычитания — изображения, отфильтрованного лапласианом, с другим изображением.

Поскольку оператор Лапласа по сути является второй производной, его применение подчеркивает разрывы уровней яркостей на изображении и подавляет области со слабыми изменениями яркостей. Это приводит к получению изображения, содержащего сероватые линии на месте контуров и других разрывов, наложенные на темный фон без особенностей. Но фон можно «восстановить», сохранив при этом эффект повышения резкости, достигаемый лапласианом. Для этого достаточно прибавить изображение-лапласиан к исходному изображению. Как было сказано в предыдущем абзаце, при этом необходимо помнить, какое из определений лапласиана было использовано. Если использовалось определение, использующее отрицательные центральные коэффициенты, тогда для получения эффекта повышения резкости изображение-лапласиан следует *вычитать*, а не прибавлять. Таким образом, обобщенный алгоритм использования лапласиана для повышения резкости изображений сводится к следующему:

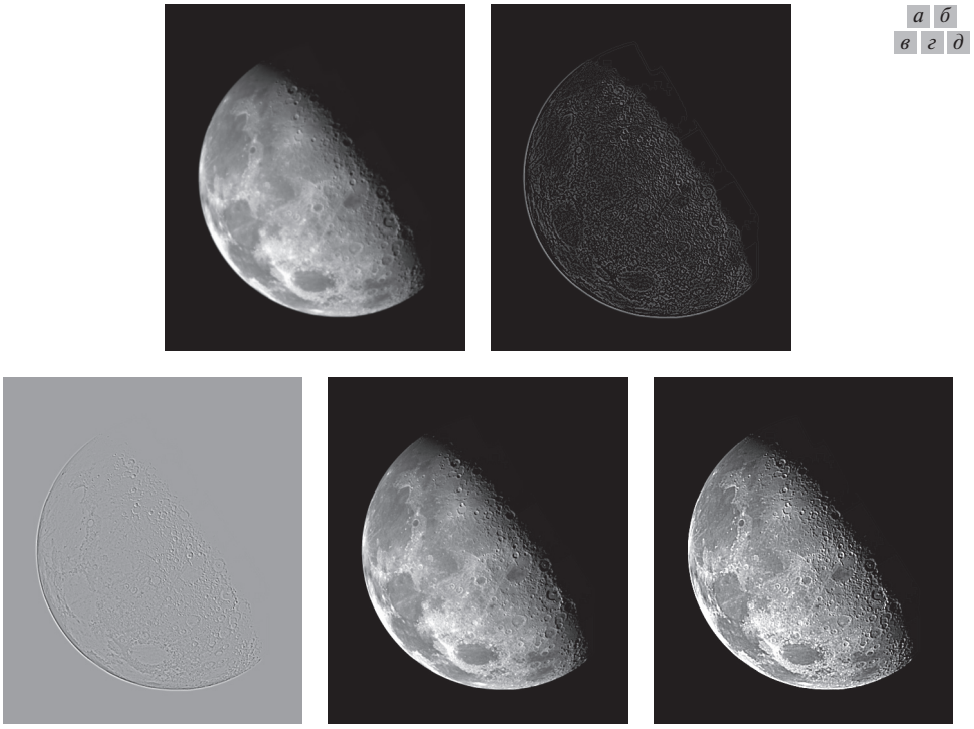
$$g(x,y) = f(x,y) + c\left[\nabla^2 f(x,y)\right], \tag{3.6-7}$$

где $f(x,y)$ и $g(x,y)$ — исходное изображение и изображение с повышенной резкостью. Константа $c = -1$, если использована маска фильтра лапласиана на рис. 3.37(*а*) или (*б*), и $c = 1$, если использован один из оставшихся двух фильтров.

Пример 3.15. Повышение резкости изображения с помощью лапласиана.

■ На рис. 3.38(*а*) представлено слегка нерезкое изображение Северного полюса Луны. На рис. 3.38(*б*) показан результат фильтрации данного изображения лапласианом с маской на рис. 3.37(*б*). Большие области этого изображения являются черными, поскольку лапласиан имеет как положительные, так и отрицательные значения, а все отрицательные обрезаются до 0 при воспроизведении.

Обычный способ преобразовать изображение-лапласиан — подвергнуть его градационной коррекции: прибавить значение его минимума ко всем значениям пикселей, а затем растянуть полученный результат на весь диапазон яркостей [0, $L - 1$] согласно уравнениям (2.6-10) и (2.6-11). Именно таким образом обработано изображение на рис. 3.38(*в*). Видно, что основными деталями данного изображения являются контуры и резкие перепады яркости различного уровня. Фон, ранее черный, теперь, вследствие градационной коррекции, стал серым. Такое сероватое проявление фона является типичным для правильно откорректированных изображений-лапласианов. На рис. 3.38(*г*) показан результат, полученный с использованием уравнения (3.6-7) с $c = -1$. Детали на этом изображении видны значительно более чистыми и резкими, чем на исходном изображении. Добавление исходного изображения к лапласиану восстанови-



а б
в г д

Рис. 3.38. (*а*) Нерезкое изображение Северного полюса Луны. (*б*) Изображение-лапласиан без масштабирования. (*в*) Изображение-лапласиан, подвергнутое градационной коррекции. (*г*) Изображение с повышенной резкостью при использовании маски на рис. 3.37(*а*). (*д*) Изображение с повышенной резкостью при использовании маски на рис. 3.37(*б*). (Исходное изображение предоставлено агентством NASA)

ло общий диапазон изменения яркостей на изображении, а лапласиан усилил контрасты в местах яркостных разрывов. Конечным результатом стало изображение, на котором мелкие детали улучшены, а фоновые полутона достаточно хорошо сохранены. Наконец, на рис. 3.38(*д*) показан результат, почти повторяющий предыдущую процедуру с использованием фильтра на рис. 3.37(*б*). Здесь заметно значительное повышение резкости по сравнению с 3.38(*г*). Это не является неожиданным, поскольку использование фильтра на рис. 3.37(*б*) добавляет дополнительное дифференцирование (повышение резкости) по диагональным направлениям. Результаты, подобные представленным на рис. 3.38(*г*) и (*д*), сделали лапласиан часто используемым инструментом для повышения резкости цифровых изображений. ■

3.6.3. Нерезкое маскирование и фильтрация с подъемом высоких частот

Процедура, много лет использующаяся в полиграфии и в издательском деле для повышения резкости изображений, заключается в вычитании из изображения его нерезкой (сглаженной) копии. Эта процедура, называемая *нерезким маскированием*, состоит из следующих шагов.

1. Создание нерезкого изображения.

2. Вычитание нерезкого изображения из исходного изображения (результирующая разность называется *маска*).

3. Прибавление маски к исходному изображению.

Обозначая через $\bar{f}(x,y)$ сглаженное изображение, нерезкое маскирование выражается в формульном виде следующим образом. Первоначально формируется маска

$$g_{\text{mask}}(x,y) = f(x,y) - \bar{f}(x,y). \tag{3.6-8}$$

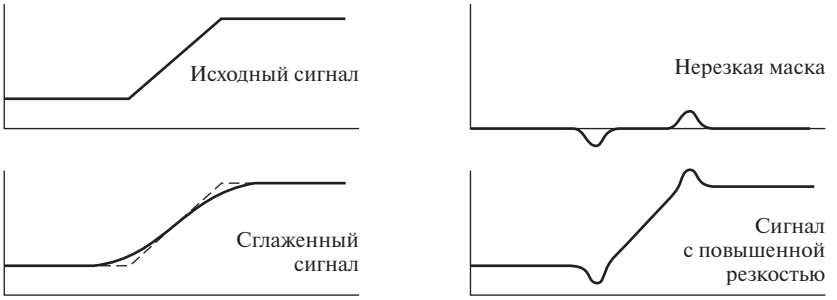
Затем маска с заданным весом прибавляется к исходному изображению:

$$g(x,y) = f(x,y) + k\, g_{\text{mask}}(x,y), \tag{3.6-9}$$

где для общности введен весовой коэффициент k ($k \geq 0$). При $k = 1$ мы получаем нерезкое маскирование, как определено выше. При $k > 1$ процесс называется *фильтрацией с подъемом высоких частот*. Выбор $k < 1$ снижает вклад нерезкой маски.

На рис. 3.39 объяснено, как действует нерезкое маскирование. График на рис. 3.39(*а*) можно интерпретировать как профиль яркости строки, пересекающей вертикальный контур на переходе от темной к светлой области на изображении. На рис. 3.39(*б*) показан результат сглаживания, наложенный на график исходного сигнала (показанный пунктиром) для ориентира. График на рис. 3.39(*в*) — нерезкая маска, полученная вычитанием сглаженного сигнала из исходного[12]. Сравнивая этот результат с участком на рис. 3.36(*в*), соответствующим склону на рис. 3.36(*а*), можно отметить, что нерезкая маска

[12] На рис. 3.39(*в*) и (*г*) представлены не вполне корректные рисунки. Вершинки пиков должны быть острыми. — *Прим. перев.*



а в
б г

Рис. 3.39. Одномерная иллюстрация механизма нерезкого маскирования. (*а*) Исходный сигнал. (*б*) Сглаженный сигнал; исходный для сравнения показан пунктиром. (*в*) Нерезкая маска. (*г*) Сигнал с повышенной резкостью, полученный прибавлением (*в*) к (*а*)

на рис. 3.39(*в*) очень похожа на то, что можно было бы получить при помощи второй производной. Конечный результат с повышенной резкостью, полученный прибавлением маски к исходному сигналу, показан на рис. 3.39(*г*). Точки, в которых были изменения наклона графика яркости, теперь оказались усиленными (повышена резкость). Заметим, что отрицательные значения также прибавляются к исходному сигналу. Тем самым в результате суммирования могут появиться отрицательные значения, если на исходном изображении были близкие к нулю значения или если значение коэффициента k выбрано излишне большим, в результате чего пики маски могут выйти за допустимый диапазон яркостей. Отрицательные значения могут вызвать появление темного ореола вокруг контуров, что при достаточно большом k может привести к нежелательному результату.

Пример 3.16. Повышение резкости изображения с помощью нерезкого маскирования.

■ На рис. 3.40 (*а*) представлено слегка сглаженное изображение белого текста на темно-сером фоне. Изображение на рис. 3.40(*б*) получено с помощью гауссова сглаживающего фильтра (см. раздел 3.4.4) размерами 5×5 с $\sigma = 3$. Рис. 3.40(*в*) — нерезкая маска, полученная преобразованием согласно уравнению (3.6-8). Изображение на рис. 3.40(*г*) получено использованием нерезкого маскирования (уравнение (3.6-9) с $k = 1$). Это изображение несколько лучше оригинала, но можно достичь большего. На рис. 3.40(*д*) показано применение уравнения (3.6-9) с $k = 4{,}5$, наибольшим значением, которое можно использовать, чтобы

а г
б д
в

Рис. 3.40. (*а*) Исходное изображение. (*б*) Результат сглаживания гауссовым фильтром. (*в*) Нерезкая маска. (*г*) Результат нерезкого маскирования. (*д*) Результат фильтрации с подъемом высоких частот

значения всех пикселей конечного результата оставались положительными. Улучшение на этом изображении по сравнению с исходным является значительным. ■

3.6.4. Использование производных первого порядка для (нелинейного) повышения резкости изображений: градиент

В обработке изображений первые производные реализуются через модуль градиента. Для функции $f(x,y)$ градиент f в точке (x,y) определяется как двумерный *вектор-столбец*:

$$\nabla \mathbf{f} \equiv \text{grad}(f) \equiv \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}. \tag{3.6-10}$$

Этот вектор имеет важное геометрическое свойство — он указывает на направление наибольшего возрастания f в точке (x,y).

> Детальное обсуждение градиента дано в разделе 10.2.5. Здесь же нас интересует лишь использование модуля градиента для повышения резкости изображений.

Модуль (*длина*) вектора ∇f обозначается как $M(x,y)$, где

$$M(x,y) = \text{mag}(\nabla f) = \sqrt{g_x^2 + g_y^2} \tag{3.6-11}$$

есть *величина* скорости изменения f в направлении вектора градиента. Заметим, что сформированное изображение $M(x,y)$ имеет тот же размер, что и исходное изображение f, поскольку x и y имеют тот же диапазон изменения. Обычной практикой является называть $M(x,y)$ *градиентным изображением* или просто *градиентом*, если смысл ясен.

Поскольку компоненты вектора градиента являются производными, они представляют собой линейные операторы. Однако модуль вектора таковым не является, поскольку он выражается через операции возведения в квадрат и извлечения квадратного корня. С другой стороны, частные производные в уравнении (3.6-10) не являются инвариантными к повороту (изотропными), но модуль вектора градиента таковым является. В некоторых приложениях бывает более выгодно с позиций вычислений вместо квадратов и квадратных корней использовать приближение значения модуля градиента суммой модулей отдельных компонентов:

$$M(x,y) \approx |g_x| + |g_y|. \tag{3.6-12}$$

Это выражение все еще сохраняет относительные изменения в уровнях яркостей, но свойство изотропности, вообще говоря, пропадает. Однако, как и в случае лапласиана, для определяемого в следующем абзаце дискретного градиента свойства изотропности сохраняются только для ограниченного набора угловых приращений, которые зависят от масок фильтров, используемых для прибли-

жения производных. Как оказывается, маски, наиболее часто используемые для приближения градиента, являются изотропными для углов, кратных 90°. Эти результаты не зависят от того, какое уравнение используется — (3.6-11) или (3.6-12), так что при использовании последнего уравнения ничего существенного не теряется.

Аналогично лапласиану первоначально будут определены дискретные приближения приведенных выше уравнений, а затем уже по ним будут сформированы соответствующие маски фильтров. Чтобы упростить последующие рассуждения, для указания точек в окрестности 3×3 будут использоваться обозначения, показанные на рис. 3.41(*а*). Так, например, центральная точка z_5 обозначает $f(x, y)$ в произвольной точке (x, y), z_1 обозначает $f(x - 1, y - 1)$ и так далее, используя нотацию, указанную на рис. 3.28. Как отмечено в разделе 3.6.1, простейшими приближениями первой производной, которые удовлетворяют сформулированным в том же разделе условиям, являются следующие: $g_x = (z_8 - z_5)$ и $g_y = (z_6 - z_5)$. Два других определения, предложенные Робертсом [Roberts, 1965] в ранних исследованиях по цифровой обработке изображений, используют перекрестные направления

$$g_x = (z_9 - z_5) \quad \text{и} \quad g_y = (z_8 - z_6). \tag{3.6-13}$$

Если выбрать уравнения (3.6-11) и (3.6-13), то градиент будет вычисляться по следующей формуле:

$$M(x, y) = \left[(z_9 - z_5)^2 + (z_8 - z_6)^2\right]^{1/2}. \tag{3.6-14}$$

Если используются уравнения (3.6-12) и (3.6-13), то

$$M(x, y) \approx |z_9 - z_5| + |z_8 - z_6|, \tag{3.6-15}$$

где x и y изменяются в пределах всего изображения так же, как и ранее. Частные производные, использующиеся в (3.6-13), могут быть реализованы при помощи двух масок линейных фильтров, показанных на рис. 3.41(*б*) и (*в*). Эти маски называют *перекрестным градиентным оператором Робертса*.

Однако маски четного размера реализовывать неудобно, поскольку они не имеют элемента, являющегося центром симметрии. Наименьшая маска фильтра, которая нас интересует, имеет размеры 3×3. Приближениями к g_x и g_y, использующими окрестность 3×3, центрированную относительно z_5, будут

$$g_x = \frac{\partial f}{\partial x} = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3) \tag{3.6-16}$$

и

$$g_y = \frac{\partial f}{\partial y} = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7). \tag{3.6-17}$$

Эти уравнения могут быть реализованы при помощи масок на рис. 3.41(*г*) и (*д*). Разность между значениями пикселей в третьей и первой строках окрестности 3×3 аппроксимирует значение производной по направлению x, а разность между значениями пикселей в третьем и первом столбцах — значение производной по направлению y. После вычисления производных при помощи этих масок амплитуда градиента вычисляется так же, как и ранее. Так, например, подставляя g_x и g_y в уравнение (3.6-12), получим

а
б *в*
г *д*

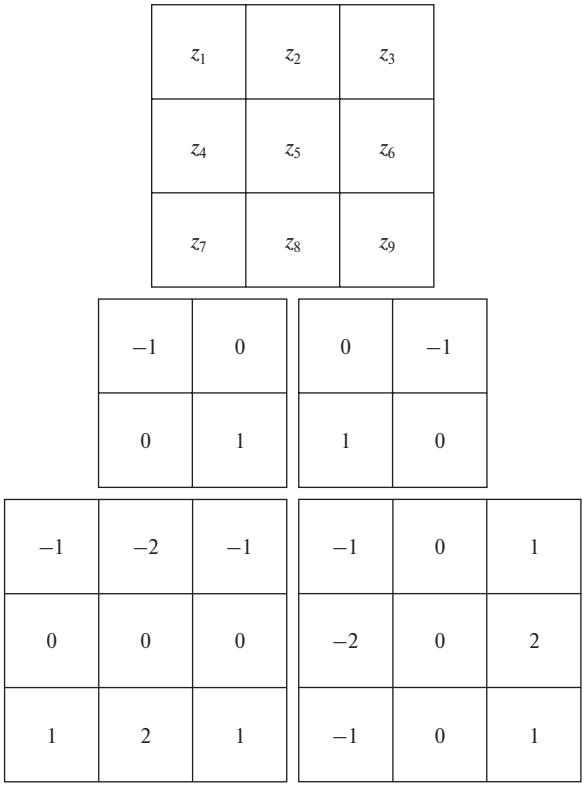


Рис. 3.41. (*а*) Область изображения размерами 3×3 элемента ($z_1, ..., z_9$ — значения яркости соответствующих пикселей под маской). (*б*)—(*в*) Перекрестный градиентный оператор Робертса. (*г*)—(*д*) Оператор Собела. Суммы коэффициентов по каждой из масок равны нулю, как и должно быть у градиентных операторов

$$M(x,y) \approx \left|(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)\right| + \left|(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)\right|. \qquad (3.6\text{-}18)$$

Маски, представленные на рис. 3.41(*г*) и (*д*), называют *оператором Собела*. Использование значения 2 у центральных весовых коэффициентов (в строках и столбцах) основано на стремлении достичь большей гладкости за счет присвоения центральным точкам большей значимости (этот вопрос детально будет рассмотрен в главе 10). Заметим, что суммы коэффициентов каждой из масок, показанных на рис. 3.41, равны 0, и это означает, что на участках изображения с постоянным уровнем яркости отклик любого из приведенных операторов будет равен 0, как и должно быть у оператора, соответствующего первой производной.

Как отмечалось ранее, вычисления g_x и g_y являются линейными операциями, поскольку основаны на производных, и, таким образом, могут реализовываться как суммы произведений, в которых используются коэффициенты пространственных масок на рис. 3.41. Нелинейный аспект повышения резкости при помощи градиента состоит в использовании при вычислении $M(x, y)$ операций возведения в квадрат и извлечения квадратного корня либо абсолютных значений; каждая из этих операций является нелинейной. Эти операции выполняются *после* линейного процесса, формирующего g_x и g_y.

Пример 3.17. Применение градиента для улучшения контуров.

■ Градиент часто используется в техническом контроле для того, чтобы помочь человеку обнаружить дефекты, или, что является более общим, как предвари-

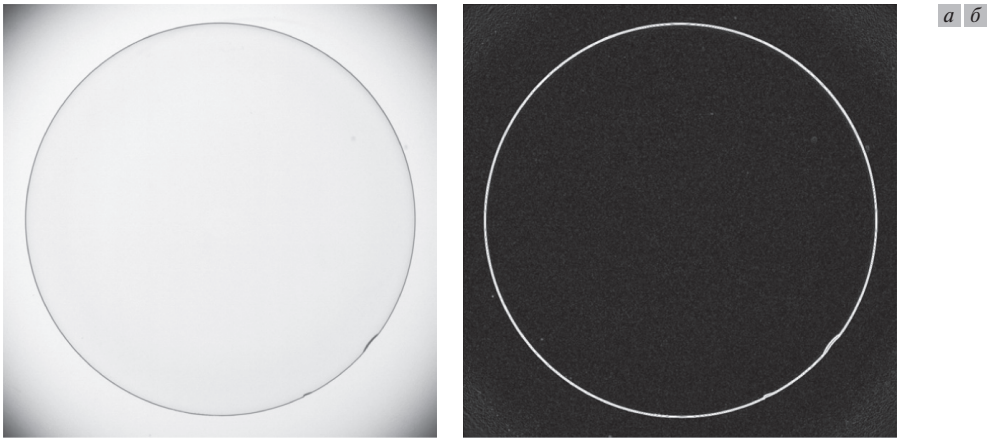


Рис. 3.42. (*а*) Оптическое изображение контактной линзы (дефекты видны на краях окружности в позициях «4 и 5 часов»). (*б*) Градиент, полученный оператором Собела. (Исходное изображение предоставил Пит Сайтс, Perceptics Corporation)

тельная обработка в автоматизированном контроле. Об этом будет говориться в главах 10 и 11. Однако представляется полезным рассмотреть некоторый простой пример прямо сейчас, чтобы показать, как может быть использован градиент для улучшения видимости дефектов и для удаления слабо меняющихся характеристик фона. В данном примере улучшение используется в качества шага предварительной обработки для последующего автоматического контроля, а не для визуального анализа.

На рис. 3.42(*а*) показано оптическое изображение контактной линзы, освещенной специальной световой установкой, предназначенной для подчеркивания неточностей. В данном случае можно заметить два дефекта на краях (они видны на окружности в позициях «4 и 5 часов»). На рис. 3.42(*б*) показан градиент, подсчитанный по формуле (3.6-12) с масками оператора Собела, показанными на рис. 3.41(*г*) и (*д*). Дефекты краев видны как на изображении 3.42(*а*), так и на 3.42(*б*), но преимуществом второго изображения является то, что устранены медленные изменения яркостей, тем самым упрощена задача автоматического контроля. Градиентная обработка может быть также использована для подчеркивания мелких дефектов, которые нелегко заметить на полутоновом изображении (такими дефектами могут быть чужеродные вкрапления, воздушные карманы в держателе, а также неточности мениска линзы). Способность подчеркнуть мелкие неоднородности на ровном сером поле есть еще одна важная характеристика градиента. ■

3.7. Комбинирование методов пространственного улучшения

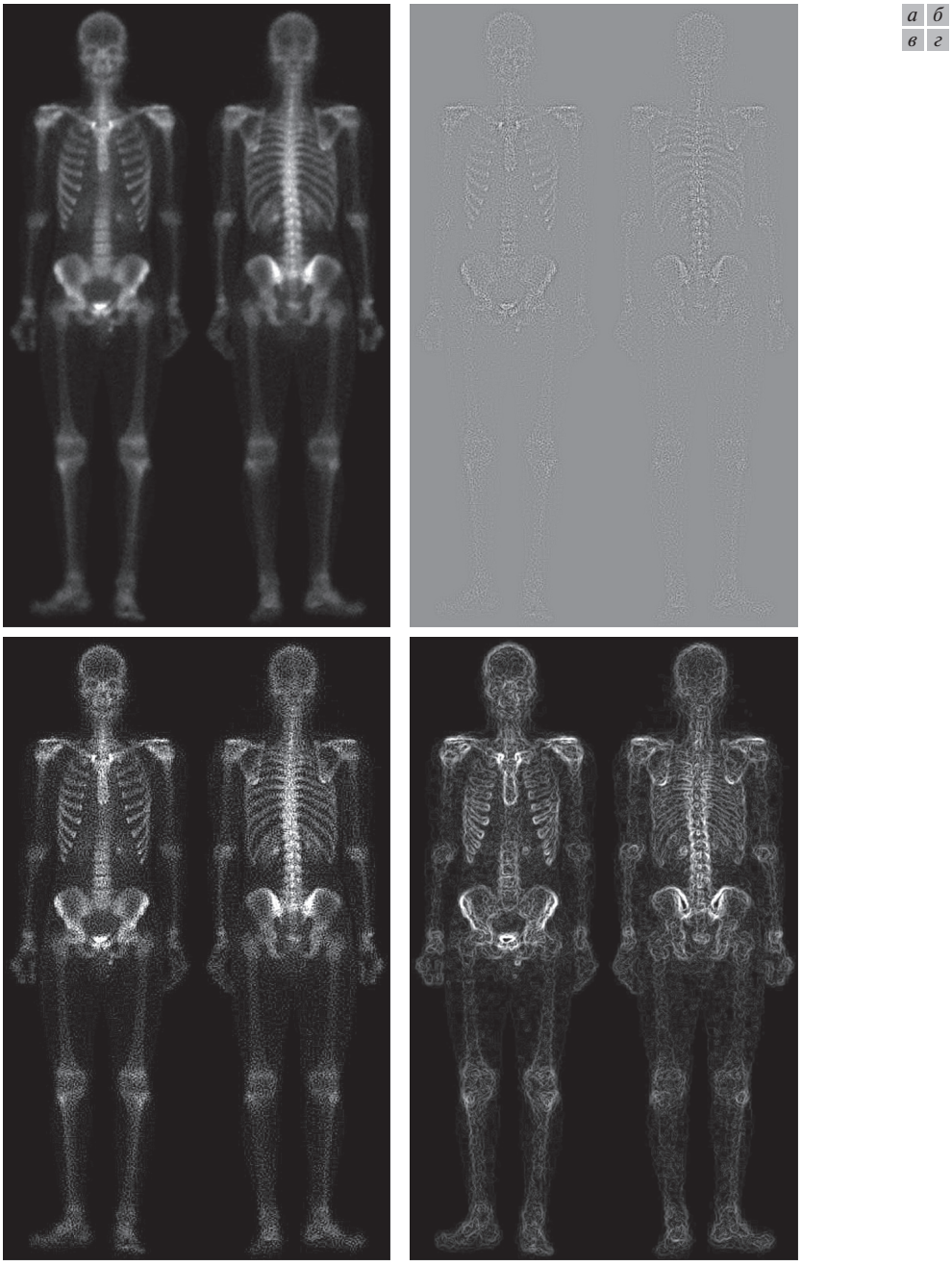
За некоторыми исключениями, как, например, в случае комбинации сглаживания с пороговым преобразованием (рис. 3.34), до настоящего момента внимание

уделялось лишь каждому из методов улучшения по отдельности. Однако при решении конкретных задач для достижения приемлемых результатов может потребоваться применить несколько дополняющих друг друга методов улучшения. В данном разделе на отдельных примерах будет проиллюстрировано, как скомбинировать отдельные подходы, разработанные к данному моменту в настоящей главе, для решения более сложной задачи улучшения изображений.

Изображение на рис. 3.43(*а*) есть полный снимок скелета, полученный с помощью гамма-лучей (см. раздел 1.3.1), использующийся для обнаружения таких заболеваний, как костные инфекции и опухоли. Целью является улучшение данного изображения путем повышения его резкости и более подробного выявления деталей скелета. Малый динамический диапазон уровней яркости и высокое содержание шума делают это изображение трудным для улучшения. Стратегия состоит в использовании лапласиана для выделения мелких деталей и градиента для улучшения выступающих краев. По причинам, которые будут объяснены ниже, сглаженная копия градиентного изображения будет использоваться для маскирования лапласиана (касательно маскирования см. рис. 2.30). В конце для увеличения динамического диапазона яркостей будет использоваться градационное преобразование.

На рис. 3.43(*б*) показан лапласиан, полученный фильтрацией исходного изображения с использованием фильтра на рис. 3.37(*г*). Для лучшей визуализации это изображение было подвергнуто градационной коррекции, такой же, как и изображение на рис. 3.38(*в*). Более резкое изображение можно получить прямо сейчас простым сложением изображений на рис. 3.43(*а*) и (*б*) в соответствии с уравнением (3.6-7). Уже глядя на уровень шумов на изображении (*б*), можно ожидать, что сложение изображений рис. 3.43(*а*) и (*б*) приведет к получению более резкого, но весьма зашумленного изображения, что и подтверждается результатом, показанным на рис. 3.43(*в*). Уменьшить уровень шума можно было бы с помощью медианного фильтра. Однако медианная фильтрация является нелинейной операцией, способной удалить детали изображения, что неприемлемо при обработке медицинских изображений.

Альтернативным подходом является использование маски, формируемой путем сглаживания градиента исходного изображения. Это нетрудно пояснить на основании свойств первой и второй производных, рассмотренных в разделе 3.6.1. Лапласиан, будучи оператором второй производной, имеет то несомненное преимущество, что является прекрасным способом улучшения мелких деталей. Однако из-за этого он усиливает шум в значительно большей степени, чем градиент. Этот шум более неприятен на гладких областях, где он весьма заметен. Градиент, по сравнению с лапласианом, дает более сильный отклик в областях со значительными изменениями яркости (на яркостных переходах и ступеньках). Отклик градиента на шум и мелкие детали слабее, чем у лапласиана, и к тому же в дальнейшем может быть снижен путем сглаживания градиента усредняющим фильтром. Тем самым, идея сводится к сглаживанию градиента и умножению его на изображение-лапласиан. В этом контексте можно рассматривать градиент как изображение-маску. Их произведение позволит сохранить детали в областях изменения яркости и подавить шум на относительно плоских участках. Этот процесс можно огрубленно интерпретировать как объединение лучших качеств лапласиана и градиента. Результат произведения прибавляется



а б
в г

Рис. 3.43. (*а*) Полный снимок скелета, полученный с помощью гамма-лучей. (*б*) Применение оператора лапласиана к изображению (*а*). (*в*) Повышение резкости сложением изображений (*а*) и (*б*). (*г*) Применение градиентного оператора Собела к изображению (*а*). (Исходное изображение предоставлено компанией G. E. Medical Systems)

д *е*
ж *з*

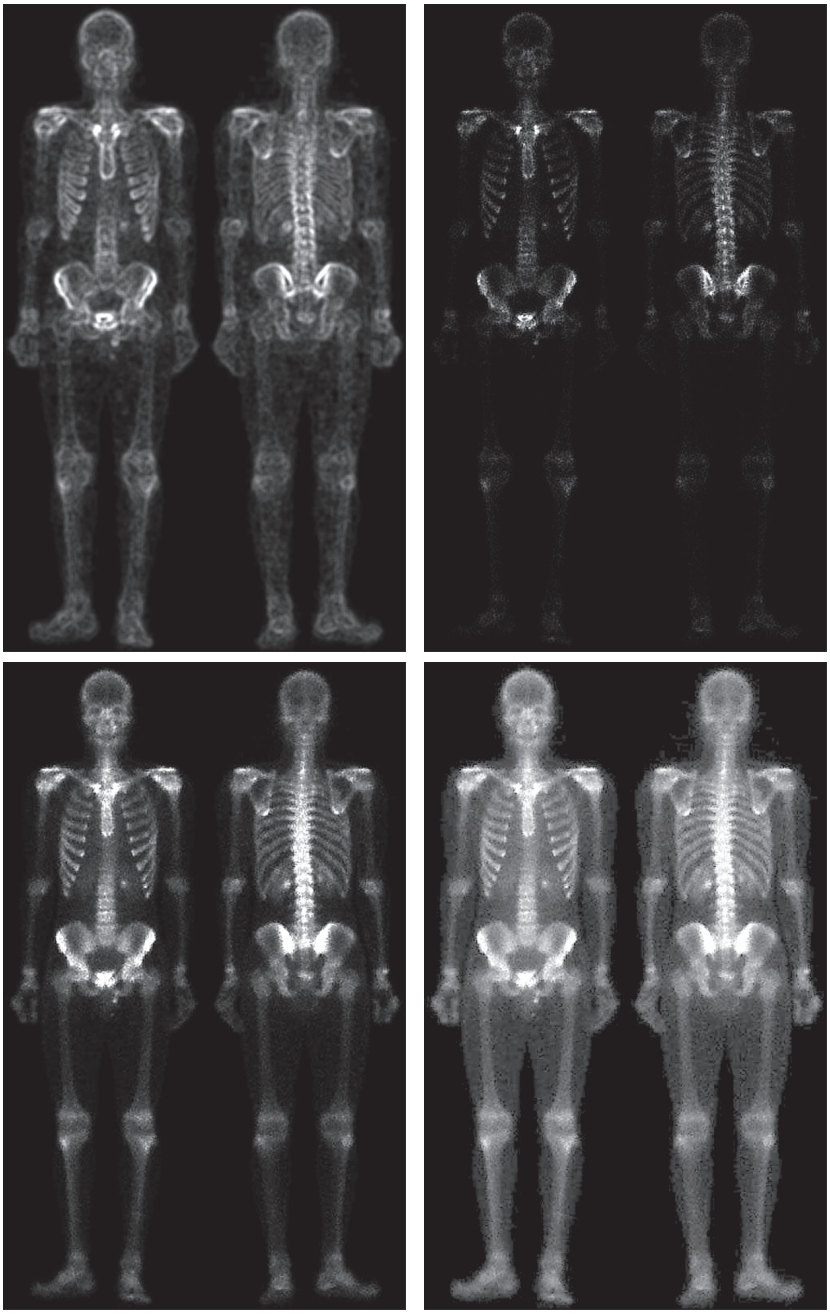


Рис. 3.43. (Продолжение.) (*д*) Изображение (*г*), сглаженное усредняющим фильтром по окрестности 5×5. (*е*) Изображение-маска, полученное перемножением изображений (*в*) и (*д*). (*ж*) Изображение с повышенной резкостью, полученное сложением изображений (*а*) и (*е*). (*з*) Конечный результат, полученный из изображения (*ж*) градационной коррекцией по степенному закону. Сравните изображения (*ж*) и (*з*) с исходным изображением (*а*)

к исходному изображению, чтобы получить окончательное изображение с повышенной резкостью.

На рис. 3.43(*г*) показан градиент, полученный фильтрацией исходного изображения оператором Собела по формуле (3.6-12). Компоненты g_x и g_y были получены с использованием масок, показанных на рис. 3.41(*г*) и (*д*) соответственно. Как и следовало ожидать, контуры на этом изображении выделяются значительно сильнее, чем на изображении-лапласиане. Сглаженное градиентное изображение, представленное на рис. 3.43(*д*), было получено с использованием усредняющего фильтра размерами 5×5. Оба градиентных изображения перед воспроизведением были подвергнуты тому же градационному преобразованию, что и изображение-лапласиан. Поскольку на градиентном изображении наименьшее допустимое значение равно 0, фон на этом изображении остается черным, а не серым, как в случае лапласиана на рис. 3.43(*б*). Тот факт, что изображения на рис. 3.43(*г*) и (*д*) выглядят значительно более яркими, чем изображение на рис. 3.43(*б*), является очевидным следствием того, что на изображении со значительным содержанием контуров градиент, как правило, имеет более высокие значения, чем лапласиан.

Результат перемножения лапласиана и сглаженного градиента показан на рис. 3.43(*е*). Заметно преобладание резких контуров и относительное снижение уровня наблюдаемого шума, что и являлось основной целью маскирования лапласиана сглаженным градиентным изображением. Прибавление полученного произведения к исходному изображению дает в результате изображение с повышенной резкостью, показанное на рис. 3.43(*ж*). Значительное повышение резкости деталей по сравнению с исходным изображением заметно на большей части данного изображения, включая ребра, позвоночник, таз и череп. Такое улучшение недостижимо при использовании одного только лапласиана или градиента.

Только что рассмотренная процедура повышения резкости не влияет в заметной степени на динамический диапазон яркостей изображения. Таким образом, финальный шаг в данной задаче улучшения состоит в увеличении динамического диапазона полученного результата. Как было детально рассмотрено в разделах 3.2 и 3.3, существует множество функций градационной коррекции, которые позволяют достичь этой цели. По результатам раздела 3.3.2 известно, что эквализация гистограммы, вероятно, не даст хорошие результаты на изображениях, распределение яркости которых сдвинуто в область черного, как на изображениях из данного примера. Решением могло бы стать задание гистограммы, однако общий черный фон изображения, с которым приходится иметь дело, заставляет сделать выбор в пользу степенного преобразования. Так как требуется растяжение уровней яркостей, то, следовательно, значение γ в уравнении (3.2-3) должно быть меньше 1. Несколько экспериментов с различными значениями параметров позволяют получить окончательный результат, показанный на рис. 3.46(*з*), при $\gamma = 0{,}5$ и $c = 1$. Сравнение данного изображения с изображением на рис. 3.43(*ж*) показывает, что стало заметным значительно большее число деталей. Участки вблизи запястий, кистей рук, лодыжек и ступней являются тому хорошими примерами. Структура костей скелета, включая кости рук и ног, также стала более отчетливой. На исходном изображении весьма слабо видны контуры тела и мягкие ткани. «Вытаскивание» подобных деталей с помощью растяжения динамического диапазона яркостей усиливает также

и шум, но, тем не менее, можно отметить, что на рис. 3.43(*з*) имеется значительное визуальное улучшение исходного изображения.

Только что рассмотренный подход показателен как пример использования взаимосвязанных процедур для получения конечного результата, недостижимого с помощью какого-то одного метода. Последовательность применения процедур зависит от задачи. Конечным пользователем класса изображений, к которому относится рассмотренное в данном примере изображение, скорее всего, является врач-рентгенолог. По причинам, которые выходят за рамки нашего рассмотрения, врачи не любят при диагностике основываться на результатах улучшения изображений. Тем не менее улучшенные изображения весьма полезны в выявлении деталей, которые могут оказаться ключевыми для последующего анализа исходного изображения или последовательности изображений. В других же областях результаты улучшения изображений могут действительно стать конечным продуктом. Примеры можно найти в полиграфии, в системах контроля продукции на основе анализа изображений, в криминалистике, в микроскопии, в системах слежения, а также во многих других областях, где главной целью улучшения является получение изображения с более высоким содержанием видимых деталей.

3.8. Применение нечетких методов для яркостных преобразований и пространственной фильтрации

В завершение настоящей главы рассмотрим введение в нечеткие множества и их применение в задачах яркостных преобразований и пространственной фильтрации, которые являлись основными темами обсуждения предыдущих разделов. Как оказывается, эти два вопроса находятся среди наиболее частых областей приложения нечетких методов для обработки изображений. В качестве отправной точки в конце главы даны ссылки на литературу по вопросам нечетких множеств и другим приложениям нечетких методов в обработке изображений. Как будет видно из последующих обсуждений, нечеткие множества обеспечивают основу для подключения человеческих знаний при решении проблем, формулировка которых базируется на неточных представлениях.

3.8.1. Введение

Как отмечалось в разделе 2.6.4, *множество* есть совокупность объектов (элементов), а *теория множеств* — набор приемов, имеющих отношение к операциям над и между множествами. Теория множеств вместе с математической логикой являются одной из аксиоматических основ классической математики. Центральным в теории множеств является понятие *принадлежности множеству*. Мы обычно имеем дело с так называемыми «четкими» множествами, принадлежность к которым может быть только *истинной* или *ложной* в традиционном духе двузначной булевой логики, где *истина* обычно обозначается как 1, а *ложь* как 0. Например, пусть Z означает множество всех людей; предположим, что мы хотим определить подмножество A в Z, называемое «множество молодых лю-

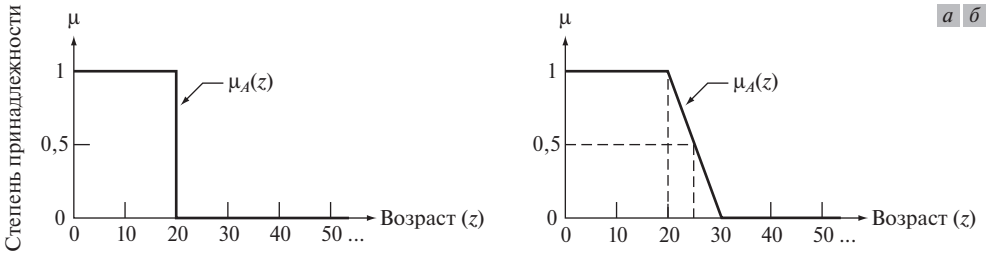


Рис. 3.44. Функции принадлежности, используемые для порождения (*а*) четкого множества и (*б*) нечеткого множества

дей». Чтобы дать определение этому подмножеству, необходимо задать *функцию принадлежности*, которая присвоит 1 или 0 каждому элементу z из Z. Поскольку мы имеем дело с двузначной логикой, функция принадлежности просто задает порог возраста, ниже которого или совпадая с которым персона признается молодой, а выше которого — не молодой. Рис. 3.44(*а*) иллюстрирует этот подход, используя возрастной порог в 20 лет и приводя только что рассмотренную функцию принадлежности, обозначенную как $\mu_A(z)$.

Функции принадлежности также называются характеристическими функциями.

Сразу видна некоторая трудность, возникающая с этим определением: персона возрастом в 20 лет признается молодой, а персона в 20 лет и 1 секунду уже не относится ко множеству молодых людей. Это является фундаментальной проблемой в работе с четкими множествами, которая ограничивает использование классической теории множеств во многих практических приложениях. Нам необходима бóльшая гибкость в том, что мы понимаем под понятием «молодой», т. е. *постепенный* переход от молодого к не молодому. На рис. 3.44(*б*) показан один вариант. Ключевым свойством этой функции является то, что она является многозначной, обеспечивая тем самым непрерывный переход между понятиями «молодой» и «не молодой». Это дает возможность иметь *степень* «молодости». Теперь можно ввести такие формулировки, как: «персона молодая» (верхняя плоская часть графика), «сравнительно молодая» (верхняя часть склона), «молодая на 50 %» (в середине склона), «не слишком молодая» (в конце склона) и так далее (заметим, что уменьшение крутизны спада функции на рис. 3.44(*б*) дает больше неопределенности в том, что понимать под «молодым»). Эти типы неопределенности (*нечеткости*) в определениях лучше согласуются с тем, что имеется в виду, когда люди говорят неточно о возрасте. Таким образом, можно интерпретировать многозначную функцию принадлежности как фундамент *нечеткой логики*, а множества, возникающие при ее использовании, могут быть рассмотрены как *нечеткие множества*. Эти идеи формализуются в следующих разделах.

3.8.2. Начала теории нечетких множеств

Теория нечетких множеств была введена Заде (L. A. Zadeh) в статье, опубликованной более сорока лет тому назад [Zadeh, 1965]. Как показано в дальнейших обсуждениях, нечеткие множества дают формализм для работы с неточной информацией.

В данном приложении для обозначения генеральной совокупности мы следуем обычной нотации нечетких множеств, используя Z вместо более традиционной нотации для множеств U.

Определения

Пусть Z есть множество элементов (объектов) с обобщенным элементом из Z, обозначаемым z; т.е. $Z = \{z\}$. Это множество называется *предметная область*. *Нечеткое множество*[13] A в Z характеризуется *функцией принадлежности* $\mu_A(z)$, которая ассоциирует с каждым элементом Z некоторое действительное число в интервале [0, 1]. Значение $\mu_A(z)$ представляет собой *степень принадлежности* z множеству A. Чем ближе значение $\mu_A(z)$ к единице, тем выше степень принадлежности z множеству A, и наоборот, когда $\mu_A(z)$ ближе к нулю. Концепция «принадлежит», столь привычная в обычных множествах, не имеет того же самого смысла в теории нечетких множеств. В обычной теории множеств мы говорим, что некоторый элемент либо принадлежит, либо не принадлежит рассматриваемому множеству. В случае нечетких множеств утверждается, что все z, для которых $\mu_A(z) = 1$, являются *полноправными* членами множества, все z, для которых $\mu_A(z) = 0$, *не* являются членами множества, а все z, для которых $\mu_A(z)$ находится между 0 и 1, имеют *частичное* членство в множестве. Таким образом, нечеткое множество есть набор *упорядоченных пар*, состоящих из значений z и соответствующей функции принадлежности, определяющей степень принадлежности каждого z. То есть

$$A = \{z, \mu_A(z) \mid z \in Z\}. \tag{3.8-1}$$

Если значения непрерывны, множество A в данном уравнении может содержать бесконечное число элементов. Если значения z дискретны, можно явно перечислить элементы A. Например, если ограничить возрастные изменения на рис. 3.44 целыми годами, то получим:

A = {(1; 1), (2; 1), (3; 1), ..., (20; 1), (21; 0,9), (22; 0,8), ..., (25; 0,5), (26; 0,4), ..., (29; 0,1)},

где, например, элемент (22; 0,8) означает, что возраст 22 имеет степень 0,8 принадлежности множеству «молодой». Все элементы с возрастом 20 и меньше являются полноправными членами этого множества, а с возрастом 30 и больше не являются членами этого множества. Заметим, что графиком этого множества будет просто дискретный набор точек, лежащих на кривой рис. 3.44(*б*), так что $\mu_A(z)$ полностью определяет A. С другой стороны, видно, что (дискретное) нечеткое множество есть не что иное, как набор точек функции, которая отображает каждый элемент проблемной области (предметной области) в числа, большие 0 и меньшие или равные 1. Тем самым часто наблюдается, что термины *нечеткое множество* и *функция принадлежности* используются наравне.

Если $\mu_A(z)$ допускает лишь два значения, скажем, 0 и 1, то *функция* принадлежности редуцируется до привычной характеристической функции обычного (четкого) множества A. Таким образом, обычные множества являются частным случаем нечетких множеств. Далее приводятся несколько определений, каса-

[13] В литературе также используется термин *нечеткое подмножество*, показывающий, что A является подмножеством Z. Однако *нечеткое множество* используется более часто.

ющихся нечетких множеств и являющихся расширениями соответствующих определений из обычной теории множеств.

Пустое множество: нечеткое множество является *пустым* тогда и только тогда, когда его функция принадлежности тождественно равна 0 на Z.

Равенство: два нечетких множества A и B *равны* (пишется $A = B$) тогда и только тогда, когда $\mu_A(z) = \mu_B(z)$ для всех $z \in Z$.

Запись «для всех $z \in Z$» читается: «для всех z, принадлежащих Z».

Дополнение: *дополнение* (NOT) нечеткого множества A, обозначаемое $\bar{A}$ или NOT(A), определяется как множество, функция принадлежности которого составляет

$$\mu_{\bar{A}}(z) = 1 - \mu_A(z) \tag{3.8-2}$$

для всех $z \in Z$.

Подмножество: нечеткое множество A является *подмножеством* B тогда и только тогда, когда

$$\mu_A(z) \le \mu_B(z) \tag{3.8-3}$$

для всех $z \in Z$.

Объединение: *объединение* (OR) двух нечетких множеств A и B, обозначаемое $A \cup B$ или A OR B, есть нечеткое множество U с функцией принадлежности

$$\mu_U(z) = \max[\mu_A(z), \mu_B(z)] \tag{3.8-4}$$

для всех $z \in Z$.

Пересечение: *пересечение* (AND) двух нечетких множеств A и B, обозначаемое $A \cap B$ или A AND B, есть нечеткое множество U с функцией принадлежности

$$\mu_U(z) = \min[\mu_A(z), \mu_B(z)] \tag{3.8-4}$$

для всех $z \in Z$.

Заметим, что хорошо известные термины NOT, OR и AND используются при работе с нечеткими множествами для обозначения соответственно дополнения, объединения и пересечения.

Пример 3.18. Иллюстрация определений, касающихся нечетких множеств.

■ Рис. 3.45 иллюстрирует некоторые из приведенных выше определений. На рис. 3.45(*а*) показана функция принадлежности двух множеств A и B, а на рис. 3.45(*б*) — функция принадлежности дополнения A. На рис. 3.45(*в*) показана функция принадлежности объединения A и B, а на рис. 3.45(*г*) — соответствующий результат для пересечения этих двух множеств. Заметим, что эти рисунки согласуются с привычной нотацией дополнения, объединения и пересечения четких множеств.[14] ■

[14] Вероятно, в литературе могут встретиться примеры, в которых площадь под кривой функции принадлежности, скажем, пересечения двух множеств, будет заштрихована, чтобы указать результат операции. Такой механический перенос обычных операций над множествами является неправильным. При работе с нечеткими множествами смысл имеют только точки вдоль самой кривой функции принадлежности.

а б
в г

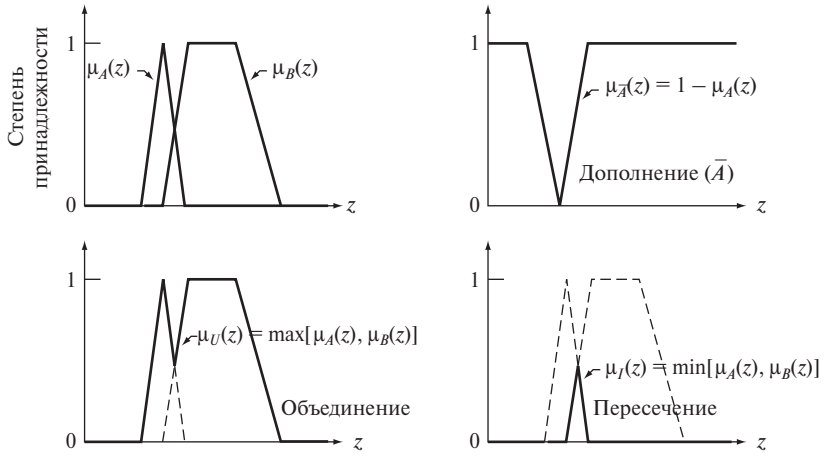


Рис. 3.45. (*а*) Функция принадлежности двух множеств *A* и *B*. (*б*) Функция принадлежности дополнения *A*. (*в*) и (*г*) Функции принадлежности объединения и пересечения двух множеств *A* и *B*

Хотя нечеткая логика и вероятность оперируют внутри одного и того же интервала значений [0, 1], между ними имеется существенная разница, которая должна быть отмечена. Рассмотрим пример на рис. 3.44. Вероятностное утверждение могло бы формулироваться: «Имеется вероятность в 50 %, что персона является молодой», тогда как нечеткое утверждение могло бы звучать: «Степень принадлежности персоны к множеству молодых составляет 0,5». Разница между двумя этими утверждениями является важной. В первом утверждении персона рассматривается как попадающая либо в множество молодых, либо в множество не молодых; просто имеется вероятность в 50 %, что персона относится к тому или иному множеству. Второе утверждение предполагает, что персона является молодой в некоторой степени, и эта степень равна 0,5 в данном случае. Другая интерпретация состоит в утверждении, что это некоторая «средняя» персона: не абсолютно молодая, но и не слишком немолодая. Другими словами, нечеткая логика не является вероятностной совершенно; она просто имеет дело со степенью принадлежности некоторому множеству. В этом случае можно сказать, что концепции нечеткой логики находят свое приложение в ситуациях, характеризующихся неопределенностью и неточностью, а не случайностью.

Некоторые распространенные функции принадлежности

Используемые на практике типы функций принадлежности включают следующие.

Треугольная:

$$\mu(z) = \begin{cases} 1-(a-z)/b & a-b \le z < a \\ 1-(z-a)/c & a \le z \le a+c \\ 0 & \text{в остальных случаях.} \end{cases} \qquad (3.8\text{-}6)$$

Трапециевидная:

$$\mu(z)=\begin{cases}1-(a-z)/c & a-c\le z<a\\ 1 & a\le z<b\\ 1-(z-b)/d & b\le z\le b+d\\ 0 & \text{в остальных случаях.}\end{cases} \tag{3.8-7}$$

Сигма:

$$\mu(z)=\begin{cases}1-(a-z)/b & a-b\le z\le a\\ 1 & z>a\\ 0 & \text{в остальных случаях.}\end{cases} \tag{3.8-8}$$

S-образная:

$$S(z;a,b,c)=\begin{cases}0 & z<a\\ 2\left(\dfrac{z-a}{c-a}\right)^2 & a\le z\le b\\ 1-2\left(\dfrac{z-a}{c-a}\right)^2 & b<z\le c\\ 1 & z>c.\end{cases} \tag{3.8-9}$$

Колоколообразная:

$$\mu(z)=\begin{cases}S(z;c-b,c-b/2,c) & z\le c\\ 1-S(z;c,c+b/2,c+b) & z>c.\end{cases} \tag{3.8-10}$$

Усеченный гауссиан:

$$\mu(z)=\begin{cases}e^{-\frac{(z-a)^2}{2b^2}} & a-c\le z\le a+c\\ 0 & \text{в остальных случаях.}\end{cases} \tag{3.8-11}$$

Для упрощения выражений при записи $\mu(z)$ обычно оставляют лишь независимую переменную z. В уравнении (3.8-9) было сделано исключение, чтобы использовать эту форму в уравнении (3.8-10). На рис. 3.46 показаны примеры рассмотренных функций принадлежности. Первые три функции являются кусочно-линейными, следующие две — гладкими, а последняя — ограниченная гауссова функция. Уравнение (3.8-9) задает важную *S*-образную функцию, которая часто используется с нечеткими множествами. Значение $z = b$, при котором $S = 0{,}5$, является *точкой перегиба*. Как видно на рис. 3.46(*г*), это та точка, в которой меняется знак кривизны линии. Нетрудно показать (задача 3.31), что $b = (a + c)/2$. В колоколообразной кривой на рис. 3.46(*д*) значение b определяет *размах* кривой.

3.8.3. Использование нечетких множеств

В настоящем разделе закладываются основы для использования нечетких множеств и иллюстрируются итоговые концепции исходя из простых, знакомых ситуаций. Затем в разделах 3.8.4 и 3.8.5 полученные результаты используются для обработки изображений. При таком изложении значительно облегчается понимание материала, особенно для новичков в этой области.

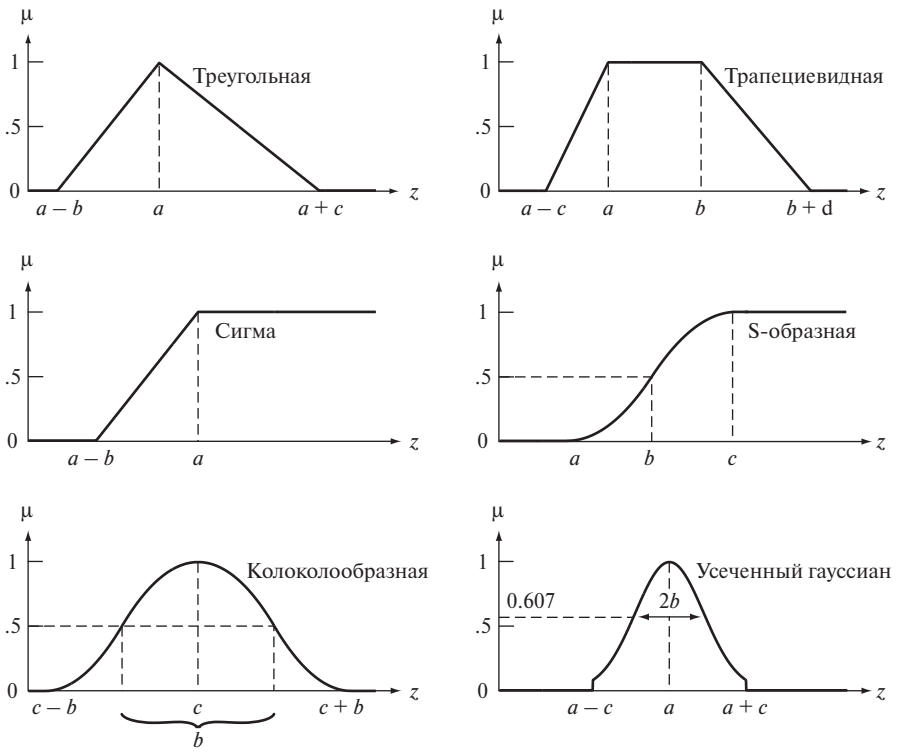


Рис. 3.46. (*а*) Функции принадлежности, соответствующие уравнениям (3.8-6)—(3.8-11)

Предположим, нас интересует использование цвета для классификации некоторого типа фруктов в три категории: неспелый, полуспелый и спелый. Предположим, что наблюдение фруктов на разных стадиях спелости позволяет сделать вывод, что неспелые фрукты выглядят зелеными, полуспелые — желтыми, а спелые — красными. Обозначения *зеленый*, *желтый* и *красный* являются не слишком четкими описаниями цветовых ощущений. В качестве отправной точки эти обозначения должны быть выражены в нечеткой форме. То есть им должна быть придана *нечеткость*. Это достигается определением принадлежности как функции цвета (длины волны света), как показано на рис. 3.47(*а*). В этом контексте *цвет* является *лингвистической переменной*, а конкретный цвет (т. е. *красный* фиксированной длины волны) является *лингвистической величиной*. Лингвистическая величина z сделана нечеткой путем использования функций принадлежности для отображения ее на интервал [0, 1], как показано на рис. 3.47(*б*).

Только что описанное проблемно-ориентированное *знание* может быть формализовано в форме следующих *нечетких правил* ЕСЛИ—ТО:

*R*1: ЕСЛИ цвет *зеленый*, ТО фрукт *неспелый*.

ИЛИ

*R*2: ЕСЛИ цвет *желтый*, ТО фрукт *полуспелый*.

ИЛИ

*R*3: ЕСЛИ цвет *красный*, ТО фрукт *спелый*.

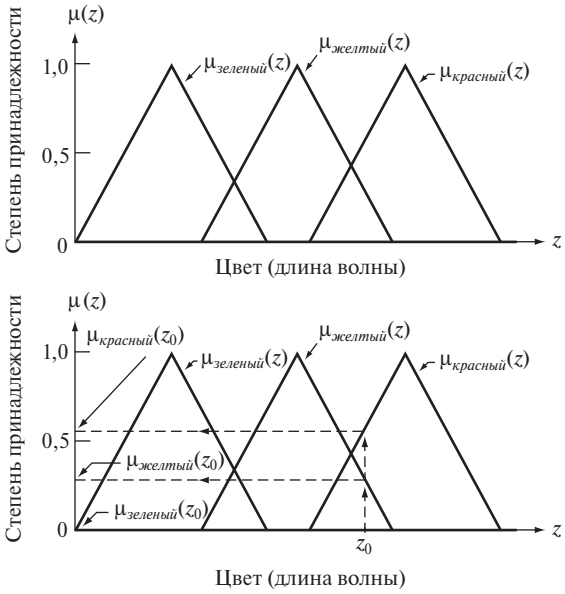


Рис. 3.47. (*а*) Функции принадлежности, используемые для придания нечеткости цветам. (*б*) Получение значения нечеткости для конкретного цвета z_0. (Кривые, описывающие цветовые ощущения, являются колоколообразными; см., например, раздел 6.1. Однако использование треугольных форм в качестве приближения является обычной практикой при работе с нечеткими множествами)

Эти правила представляют итоговый результат нашего знания об этой проблеме; реально они не являются ничем, кроме формализма для процесса мышления.

> Та часть ЕСЛИ—ТО правила, которая находится слева от ТО, часто называется посылкой (или исходным условием). Часть справа называется следствием (или выводом).

Следующий шаг процедуры — нахождение способа использовать входы (цвета) и базу знаний, представленную правилами ЕСЛИ—ТО, чтобы получить выход нечеткой системы. Этот процесс называется *импликация* или *логический вывод.* Однако до того, как может осуществиться импликация, исходные условия по каждому из правил должны быть обработаны и получено *единое* значение. Как показано в конце данного раздела, многочисленные части исходных условий связываются при помощи операторов AND (И) и OR (ИЛИ). На основе на определений раздела 3.8.2 это означает выполнение операций минимума и максимума. Чтобы упростить объяснение, мы изначально рассматриваем правила, исходные условия которых содержат только одну часть.

Поскольку нам приходится иметь дело с нечеткими входами, выходы тем более являются нечеткими, значит, функции принадлежности должны быть определены также и для выходов. На рис. 3.48 показаны функции принадлежности нечетких выходов, которые мы собираемся использовать в этом примере.

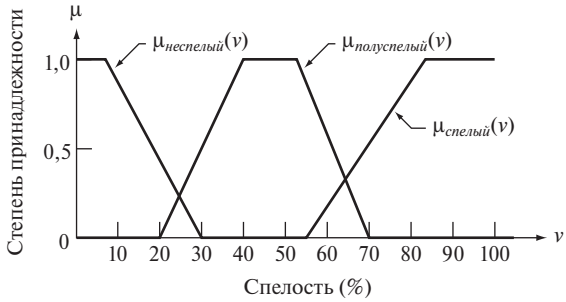


Рис. 3.48. Функции принадлежности, характеризующие выходы *неспелый*, *полуспелый* и *спелый*

Заметим, что независимой переменной на выходе является *спелость*, которая отличается от независимых переменных на входах.

Рис. 3.47 и 3.48 вместе с базой правил содержат всю информацию, требуемую для связи входов с выходами. Например, мы знаем, что выражение «*красный* И *спелый*» есть не что иное, как операция пересечения (И), определенная ранее. В данном случае независимые переменные функций принадлежности входов и выходов являются различными, так что результат будет двумерным. Например, на рис. 3.49(*а*) и (*б*) показаны функции принадлежности *красного* и *спелого*, а рис. 3.49(*в*) иллюстрирует, как они связаны в двух измерениях. Чтобы найти результат операции И между этими двумя функциями, напомним из уравнения (3.8-5), что И определяется как минимум двух функции принадлежности, т. е.

$$\mu_3(z, v) = \min\{\mu_{красный}(z), \mu_{спелый}(v)\}, \qquad (3.8\text{-}12)$$

где цифра 3 в индексе означает, что это результат применения правила R_3 из базы знаний. На рис. 3.49(*г*) приведен результат операции И[15].

Уравнение (3.8-12) является общей зависимостью, включающей две функции принадлежности. На практике нас интересует выход, получающийся при *конкретном* входе. Пусть z_0 — некоторое конкретное значение *красного*. Степень принадлежности красной цветовой компоненты в ответ на этот вход есть просто скалярная величина $\mu_{красный}(z_0)$. Ищется выход, соответствующий правилу R_3 и конкретному значению z_0 выполнением операции И между $\mu_{красный}(z_0)$ и результатом общего вида $\mu_3(z, v)$, также вычисленным при z_0. Как отмечалось ранее, операция И реализуется через отыскание минимума:

$$Q_3(v) = \min\{\mu_{красный}(z_0), \mu_3(z_0, v)\}, \qquad (3.8\text{-}13)$$

---

[15] Заметим, что уравнение (3.8-12) сформировано из упорядоченных пар значений $\{\mu_{красный}(z), \mu_{спелый}(v)\}$, а также напомним, что множество упорядоченных пар обычно называется *декартовым произведением*, обозначаемым $X \times V$, где X — набор значений $\{\mu_{красный}(z_1), \mu_{красный}(z_2), ..., \mu_{красный}(z_n)\}$, получаемых из $\mu_{красный}(z)$ варьированием z, а V — аналогичный набор из n значений $\mu_{спелый}(v)$, получаемый варьированием v. Таким образом, $X \times V = \{(\mu_{красный}(z_1), \mu_{спелый}(v_1)), ..., (\mu_{красный}(z_n), \mu_{спелый}(v_n))\}$, и из рис. 3.49(*г*) видно, что операция И, включающая две переменные, может быть выражена как отображение из $X \times V$ в диапазон [0, 1], обозначаемое $X \times V \rightarrow [0, 1]$. Хотя в настоящем рассмотрении эти обозначения и не используются, мы упоминаем их здесь, поскольку они, вероятно, могут встретиться в литературе по нечетким множествам.

где $Q_3(v)$ означает нечеткий выход в результате правила R_3 и конкретного входа. Единственной переменной в Q_3 является выходная переменная v, как и ожидалось.

Чтобы интерпретировать уравнение (3.8-13) графически, рассмотрим опять рис. 3.49(*г*), на котором приведена обобщенная функция $\mu_3(z, v)$. При выполнении операции минимума данной функции с положительной константой c все значения $\mu_3(z, v)$, бóльшие константы, будут урезаны, как это показано на рис. 3.50(*а*). Однако нас интересует только значение z_0 на цветовой оси, поэтому важен только профиль функции, урезанной плоскостью $z = z_0$, вдоль оси спелости, как показано на рис. 3.50(*б*) (рис. 3.50(*а*) соответствует правилу R_3, следовательно, $c = \mu_{красный}(z_0)$). Уравнение (3.8-13) есть выражение для этого профиля.

Используя ту же самую нить рассуждений, можно получить нечеткие отклики согласно остальным двум правилам и выбранному входу z_0 следующим образом:

$$Q_2(v) = \min\{\mu_{желтый}(z_0), \mu_2(z_0, v)\} \tag{3.8-14}$$

и

$$Q_1(v) = \min\{\mu_{зеленый}(z_0), \mu_1(z_0, v)\}. \tag{3.8-15}$$

Каждое из этих уравнений есть выход, ассоциированный с конкретным правилом и заданным входом. То есть они представляют результат процесса импликации, приведенного несколькими абзацами выше. Следует иметь в виду, что каждый из этих трех откликов есть нечеткое множество, несмотря на то, что значение на входе скалярное.

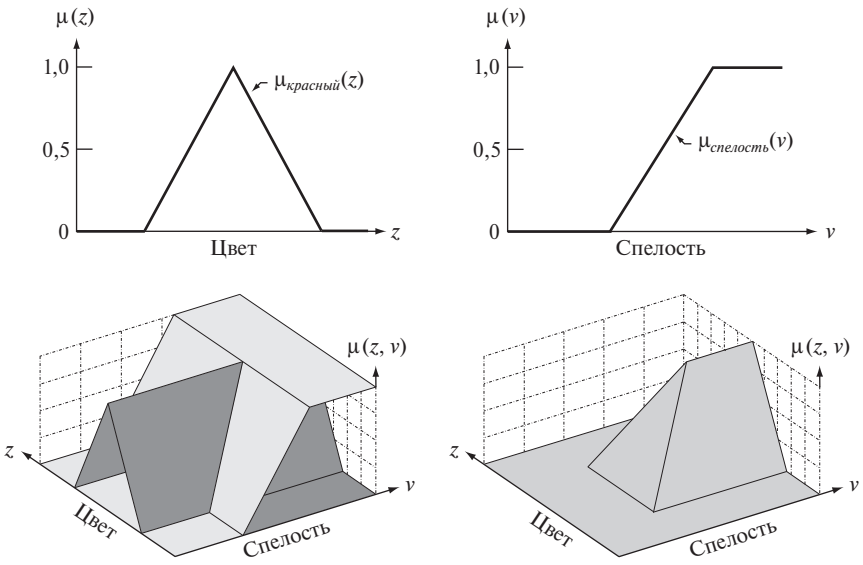


Рис. 3.49. (*а*) Форма функции принадлежности, ассоциируемой с цветом «красный», и (*б*) соответствующая выходная функция принадлежности. Эти две функции связаны правилом R_3. (*в*) Объединенное представление двух функций. Представление является двумерным, поскольку независимые переменные в (*а*) и (*б*) разные. (*г*) Результат И для функций (*а*) и (*б*), как определено в уравнении (3.8-5)

а б

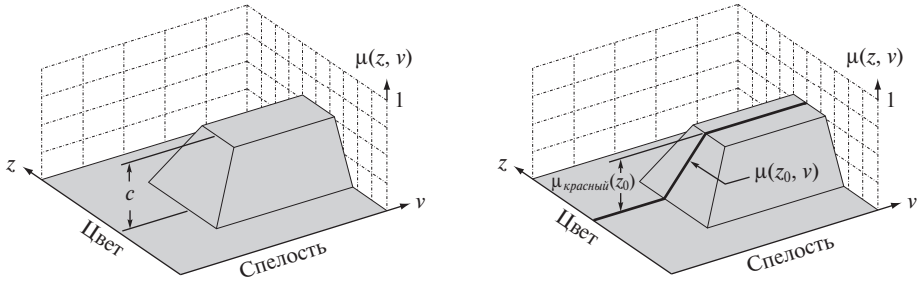


Рис. 3.50. (*а*) Результат вычисления минимума для функции $\mu_3(z, v)$ из уравнения (3.8-12) и произвольной константы *c*. Минимум эквивалентен операции И. (*б*) Профиль (темная линия) по значению конкретного цвета z_0

Чтобы получить общий отклик, осуществляется *группировка* отдельных откликов. В базовых правилах, сформулированных в начале данного раздела, три правила объединены операцией ИЛИ. Тем самым конечный (сгруппированный) нечеткий выход задается следующим образом:

$$Q = Q_1 \text{ ИЛИ } Q_2 \text{ ИЛИ } Q_3, \tag{3.8-16}$$

т.е. общий отклик есть объединение трех отдельных нечетких множеств. Поскольку операция ИЛИ определена как операция максимума, можно записать результат следующим образом:

$$Q(v) = \max_r \left\{ \min_s \left\{ \mu_s(z_0), \mu_r(z_0, v) \right\} \right\} \tag{3.8-17}$$

для $r = \{1, 2, 3\}$ и s = {*зеленый*, *желтый*, *красный*}. Хотя результат получен в контексте некоторого примера, данное выражение является совершенно общим; чтобы расширить его на *n* правил, достаточно принять $r = \{1, 2, ..., n\}$; аналогично можно расширить *s* до любого конечного числа функций принадлежности. Уравнения (3.8-16) и (3.8-17) утверждают одно и то же: «Отклик *Q* нечеткой системы есть объединение отдельных нечетких множеств, возникающих по каждому из правил в процессе импликации».

Рис. 3.51 графическим образом суммирует обсуждение, проводимое до настоящего момента. На рис. 3.51(*а*) показаны три входные функции принадлежности, оцененные в точке z_0, а на рис. 3.51(*б*) — выходы в ответ на значение входа z_0. Эти нечеткие множества являются урезанными профилями, рассмотренными в контексте рис. 3.50(*б*). Заметим, что численно Q_1 состоит из одних нулей, поскольку $\mu_{зеленый}(z_0) = 0$; т.е. Q_1 является *пустым*, как это определено в разделе 3.8.2. На рис. 3.51(*в*) показан конечный результат *Q* отдельно как нечеткое множество, сформированное из объединения Q_1, Q_2 и Q_3.

Полный выход, соответствующий конкретному входу, успешно получен, но он все еще является нечетким множеством. Последним шагом является получение четкого выхода v_0 на основе нечеткого множества *Q* путем использования процесса, который уместно называть *устранением нечеткости*. Существует много способов устранения нечеткости *Q* для получения четкого выхода. Один из наиболее частых способов — вычисление центра тяжести этого множества (обсуждение других можно найти в источниках по ссылкам, приведенным

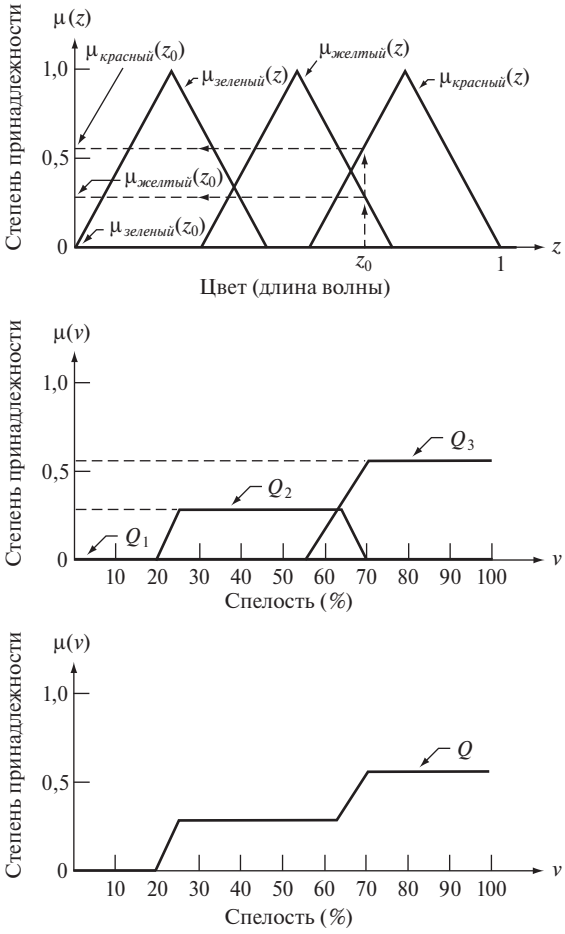


Рис. 3.51. (*а*) Функции принадлежности с отмеченным выбранным цветом z_0. (*б*) Отдельные нечеткие множества, полученные из уравнений (3.8-13)—(3.8-15). (*в*) Конечное нечеткое множество, полученное при помощи уравнений (3.8-16) или (3.8-17)

в конце настоящей главы). Если $Q(v)$ в уравнении (3.8-17) может принимать K значений $Q(1)$, $Q(2)$, ..., $Q(K)$, его центр тяжести находится следующим образом:

$$v_0 = \frac{\sum_{v=1}^{K} vQ(v)}{\sum_{v=1}^{K} Q(v)}. \qquad (3.8\text{-}18)$$

Подставляя в это уравнение (дискретные)[16] значения Q на рис. 3.51(*в*), получим $v_0 = 72{,}3$, что говорит о том, что заданный цвет z_0 означает приблизительно 72 % спелости фрукта.

[16] Нечеткое множество Q на рис. 3.51(*в*) показано непрерывным для удобства изложения, но следует иметь в виду, что в настоящей книге мы имеем дело с оцифрованными значениями, так что Q является дискретной функцией.

До настоящего момента мы рассматривали правила ЕСЛИ—ТО, исходные условия которых имели только один параметр, такой как «ЕСЛИ цвет *красный*». Правила, содержащие несколько посылок, должны быть скомбинированы так, чтобы в результате получилось *единственное* число, отражающее все части исходных условий этого правила. Например, предположим, имеется правило: «ЕСЛИ цвет *красный* ИЛИ состояние *мягкое*, ТО фрукт *спелый*». Для лингвистической переменной *мягкий* должна быть определена функция принадлежности. Чтобы получить единственное число для правила, учитывающего обе части исходных условий, первоначально оцениваются заданное значение входного цвета *красный*, используя функцию принадлежности *красного*, и заданное значение *состояния*, используя функцию принадлежности *мягкого*. Поскольку обе части связаны оператором ИЛИ, используется максимум двух результирующих значений[17]. Это значение затем используется в процессе импликации для «урезания» выходной функции принадлежности *спелости*, которая и является функцией, ассоциированной с этим правилом. Остаток процедуры выглядит так же, как и ранее, что иллюстрируется следующим кратким изложением.

На рис. 3.52 представлен пример с фруктом, использующий два входа: *цвет* и *состояние*. Можно использовать этот рисунок и предшествующий материал, чтобы суммировать основные шаги процедуры применения правил нечеткой логики.

1. *Размыть входные параметры* (*сделать нечеткими*). Для каждого скалярного входа найти соответствующие нечеткие значения путем отображения значения на входе в интервал [0, 1], используя подходящие функции принадлежности в каждом из правил, как это показано в первых двух столбцах на рис. 3.52.
2. *Выполнить все требуемые операции нечеткой логики*. Выходы всех частей исходных условий должны быть скомбинированы с целью получения *единого* значения, используя для этого операции максимума или минимума, в зависимости от того, связываются ли части операцией ИЛИ либо операцией И. На рис. 3.52 все части исходных условий связаны операциями ИЛИ, так что повсюду используется операция максимума. Число частей исходных условий и тип используемых для их связывания логических операций могут различаться от правила к правилу.
3. *Применить метод импликации*. Для создания выхода по каждому из правил используется единый выход исходных условий по этому правилу. Для импликации используется оператор И, который определен как операция минимума. Это урезает соответствующую функцию принадлежности выхода на уровне значения, определяемого исходными условиями, как это показано в третьем и четвертом столбцах на рис. 3.52.
4. *Применить метод объединения к нечетким множествам, полученным на шаге 3*. Как видно на последнем столбце рис. 3.52, выходом каждого из правил является нечеткое множество. Они должны быть объединены для формирования единственного выходного нечеткого множества. Используемый здесь подход — применение операции ИЛИ к результатам отдельных выходов, поэтому задействуется операция максимума.

[17] Исходные условия, части которых связаны оператором И, оцениваются аналогичным образом, используя операцию минимума.

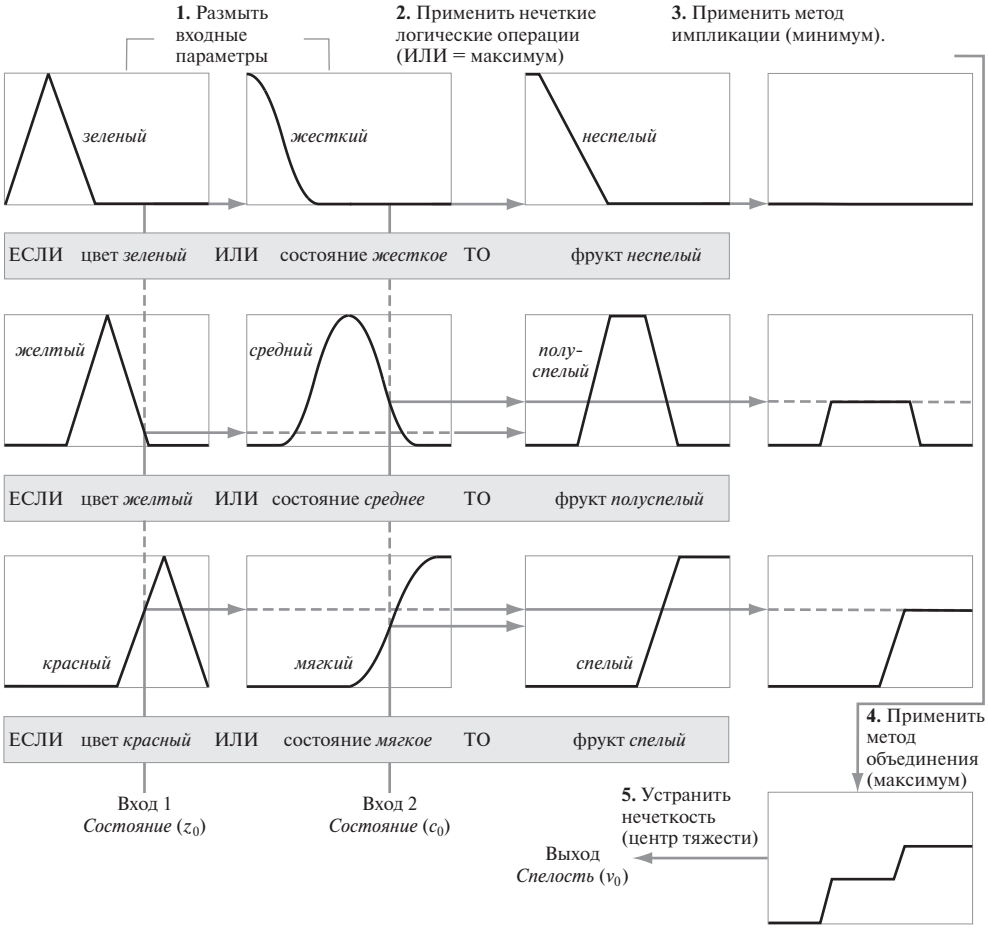


Рис. 3.52. Пример, иллюстрирующий пять основных шагов, обычно применяемых при использовании системы с правилами нечеткой логики: (1) размывание параметров, (2) логические операции (в данном примере используется только операция ИЛИ), (3) импликация, (4) объединение, (5) устранение нечеткости

5. *Устранить нечеткость выходного нечеткого множества* (*получить результат*). На этом последнем шаге выход получается в четком скалярном виде. Это достигается вычислением центра тяжести объединенного нечеткого множества, полученного на шаге 4.

Когда число переменных велико, обычной практикой является использование условных обозначений типа (переменная, нечеткое множество), позволяющее объединить в пары переменную с соответствующей ей функцией принадлежности. Например, правило «ЕСЛИ цвет *зеленый*, ТО фрукт *неспелый*» будет записано как «ЕСЛИ (z, зеленый) ТО (v, неспелый)». Здесь, как и ранее, переменные z и v представляют соответственно цвет и спелость, а *зеленый* и *неспелый* — два нечетких множества, определенных функциями принадлежности $\mu_{зеленый}(z)$ и $\mu_{неспелый}(v)$.

Вообще, при использовании M правил ЕСЛИ—ТО, N входных переменных $z_1, z_2, ..., z_N$ и одной выходной переменной v тип формулировок правил нечеткой логики, наиболее часто использующихся в обработке изображений, имеет вид:

ЕСЛИ (z_1, A_{11}) И (z_2, A_{12}) И ... И (z_N, A_{1N}) ТО (v, B_1)
ЕСЛИ (z_1, A_{21}) И (z_2, A_{22}) И ... И (z_N, A_{2N}) ТО (v, B_2)
.....
ЕСЛИ (z_1, A_{M1}) И (z_2, A_{M2}) И ... И (z_N, A_{MN}) ТО (v, B_M)
ИНАЧЕ (v, B_E), (3.8-19)

где A_{ij} — нечеткое множество, ассоциированное с i-м правилом и j-й входной переменной, B_i — нечеткое множество, ассоциированное с выходом i-того правила, и также предполагается, что составные части исходных условий правила связаны операторами И. Заметим, что введено правило ИНАЧЕ с ассоциированным нечетким множеством B_E. Это правило исполняется в случае, если ни одно из предшествующих правил не удовлетворяется полностью; его выход объясняется ниже.

Как уже отмечалось ранее, значения всех элементов исходных условий каждого из правил должны быть определены, чтобы получить единственное скалярное значение. На рис. 3.52 была использована операция максимума, поскольку правила основывались на нечетких операциях ИЛИ. Формулировки правил (3.8-19) используют операции И, значит, необходимо использовать операцию минимума. Определение исходных условий i-го правила в (3.8-19) дает в результате скалярное значение λ_i, задаваемое формулой

$$\lambda_i = \min\left\{\mu_{A_{ij}}(z_j); \quad j = 1,2,...,N\right\} \tag{3.8-20}$$

для $i = 1, 2, ..., M$, где $\mu_{A_{ij}}(z_j)$ является функцией принадлежности нечеткого множества A_{ij}, полученного по значению j-го входа. Часто λ_i называют *уровнем мощности* (или *уровнем включения*) i-го правила. Со ссылкой на предыдущее обсуждение λ_i является попросту значением, используемым для урезания выходной функции i-го правила.

> Использование операторов И или ИЛИ в наборе правил зависит от того, как правила сформулированы, что, в свою очередь, зависит от рассматриваемой задачи. Мы использовали операции ИЛИ на рис. 3.52 и операции И в правилах (3.8-19), чтобы познакомить с обеими формулировками.

Правило ИНАЧЕ исполняется в случае, если слабо удовлетворяются условия всех правил ТО (детальный пример того, как используются правила ИНАЧЕ, дан в разделе 3.8.5). Это правило должно давать сильный отклик в случае слабых откликов всех остальных. В известном смысле можно рассматривать правило ИНАЧЕ как выполнение операции НЕ по условиям всех остальных правил. Из раздела 3.8.2 известно, что $\mu_{\text{НЕ}(A)} = \mu_A(z) = 1 - \mu_A(z)$. Тогда, используя этот способ комбинирования (операций И) всех уровней условий правил ТО, получим следующий уровень включения для правила ИНАЧЕ:

$$\lambda_E = \min\{1 - \lambda_i; \quad i = 1, 2, ..., M\}. \tag{3.8-21}$$

Видно, что если хотя бы одно из правил ТО включается на «полный уровень» (его отклик равен 1), то отклик правила ИНАЧЕ равен 0, как и следовало ожи-

дать. В случае снижения уровней откликов условий правил ТО мощность правила ИНАЧЕ возрастает. Это является нечетким эквивалентом известных правил ЕСЛИ—ТО—ИНАЧЕ, используемых в обычном программировании.

В случае, когда в исходных условиях используются операторы ИЛИ, операторы И в правилах (3.8-19) просто заменяются на ИЛИ, а также min в уравнении (3.8-20) заменяется на max; уравнение (3.8-21) не изменяется. Хотя можно сформулировать и более сложные исходные условия и следствия правил, чем рассматривались здесь, изложенные формулировки, использующие только операторы И и ИЛИ, являются абсолютно общими и применяются в широком спектре приложений, связанных с обработкой изображений. Ссылки в конце настоящей главы содержат дополнительные (но реже используемые) определения нечетких логических операторов, а также обсуждения иных методов импликации (включая множественные выходы) и устранения нечеткости. Введение, изложенное в этом разделе, является фундаментальным и служит твердой основой для более продвинутого изучения этой темы. В следующих двух разделах будет показано, как применять концепции нечеткой логики при обработке изображений.

3.8.4. Использование нечетких множеств для яркостных преобразований

Рассмотрим обычную проблему повышения контрастов — одного из основных приложений из области яркостных преобразований. Можно сформулировать процесс повышения контрастов полутонового черно-белого изображения, используя следующие правила:

ЕСЛИ пиксель темный, ТО сделать его темнее.
ЕСЛИ пиксель серый, ТО сделать его серым.
ЕСЛИ пиксель светлый, ТО сделать его светлее.

Имея в виду, что это нечеткие термины, можно выразить концепции *темный*, *серый* и *светлый* при помощи функции принадлежности, показанной на рис. 3.53(*а*).

В терминах выходного результата можно рассматривать термин *темнее* как означающий «иметь степень яркости в области более темных значений» (100% черный означает предельный уровень темного); термин *светлее* означает «иметь степень яркости в области более светлых значений» (100% белый является пре-

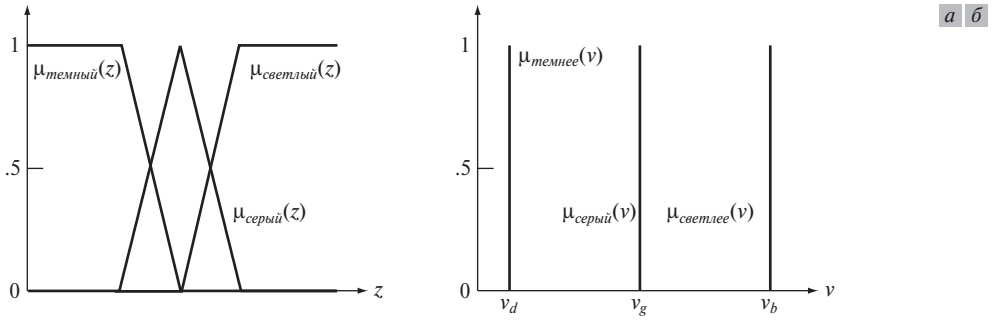


Рис. 3.53. (*а*) Входная и (*б*) выходная функции принадлежности для повышения контраста на основе правил нечеткой логики

дельным значением светлого); и *серый* означает «иметь степень яркости в области средних значений шкалы». Под степенью здесь понимается величина некоторой конкретной яркости; например, 80 % черный является очень темным серым. При модификации *константных* яркостей выходные функции принадлежности получают *одно значение* (т.е. также являются *константами*), как показано на рис. 3.53(*б*). Различные значения яркости в диапазоне [0, 1] появляются вследствие урезания опорных значений интенсивностью отклика соответствующих правил, как показано в четвертом столбце на рис. 3.52 (но следует учесть, что здесь мы работаем только с одним входом, а не с двумя, как на рисунке). Поскольку мы имеем дело с константами в выходных функциях принадлежности, из уравнения (3.8-18) следует, что выход v_0 при любом входе z_0 определяется формулой

$$v_0 = \frac{\mu_{темный}(z_0)v_d + \mu_{серый}(z_0)v_g + \mu_{светлый}(z_0)v_b}{\mu_{темный}(z_0) + \mu_{серый}(z_0) + \mu_{светлый}(z_0)}. \qquad (3.8\text{-}22)$$

В данном выражении суммирование в числителе и знаменателе проще, чем в уравнении (3.8-18), поскольку выходные функции принадлежности являются константами, модифицированными (урезанными) размытыми значениями.

Нечеткая обработка изображений требует интенсивных вычислений, поскольку весь процесс размывания, обработки исходных условий для всех правил, импликация, объединение и устранение нечеткости должны быть применены к *каждому* пикселю входного изображения. При использовании в уравнении (3.8-22) константных значений вычислительные требования значительно снижаются, поскольку упрощаются импликация, объединение и устранение нечеткости. Эта экономия может оказаться существенной в приложениях, где важным требованием является скорость обработки.

Пример 3.19. Иллюстрация улучшения изображения путем модификации контраста на основе правил нечеткой логики.

■ На рис. 3.54(*а*) представлено изображение с узким диапазоном яркостей (см. гистограмму изображения на рис. 3.55(*а*)), что приводит к впечатлению низкого контраста на изображении. В качестве базы для сравнения на рис. 3.54(*б*) показан результат эквализации гистограммы. Как видно из гистограммы после эквализации на рис. 3.55(*б*), растягивание диапазона усиливает контраст, однако в итоге значения пикселей оказываются в нижнем и верхнем концах шкалы яркостей, и изображение приобретает вид «переэкспонированного». В частности, детали на лбу и в волосах Альберта Эйнштейна по большей части утеряны. На рис. 3.54(*в*) представлен результат модификации контраста на основе использования правил нечеткой логики — подхода, обсуждавшегося в предыдущих параграфах. На рис. 3.55(*в*) приведены входные функции принадлежности, наложенные на гистограмму исходного изображения. Значения выходных опорных значений были выбраны как $v_d = 0$ (черный), $v_g = 127$ (серый) и $v_b = 255$ (белый).

При сравнении изображений на рис. 3.54(*б*) и (*в*) на последнем из них видно значительное улучшение тональности. Заметим, например, насколько лучше проработаны детали на лбу и в волосах по сравнению с рис. 3.54(*б*). Причина такого улучшения может быть легко объяснена путем анализа гистограммы на рис. 3.55(*г*). В отличие от гистограммы эквализованного изображения, дан-

а б в

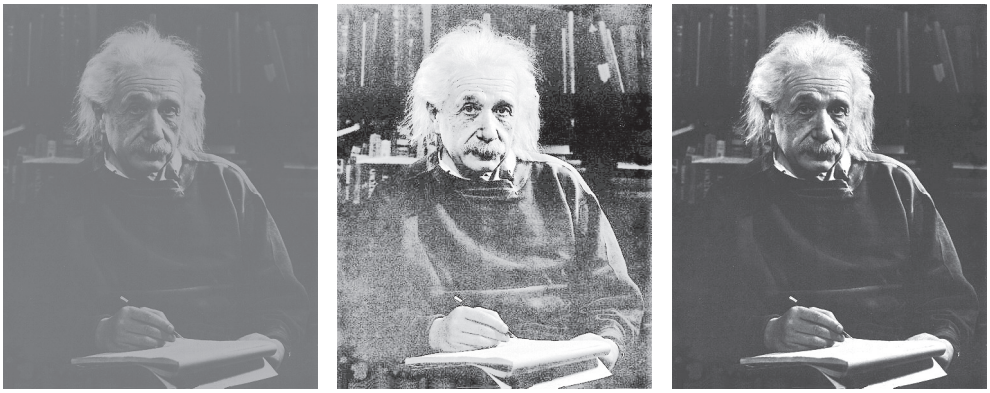


Рис. 3.54. (*а*) Малоконтрастное изображение. (*б*) Результат эквализации гистограммы. (*в*) Результат применения метода повышения контрастов, основанного на правилах нечеткой логики

ная гистограмма сохраняет ту же основную форму, которая была у гистограммы исходного изображения. Однако совершенно очевидно, что уровни темного (высокие пики в области меньших значений гистограммы) сдвинуты влево, т. е. стали еще темнее. Со светлыми уровнями произошло обратное — они сдвинуты в область более ярких значений. Средние уровни слегка растянуты, но значительно меньше, чем в случае эквализации гистограммы.

а б
в г

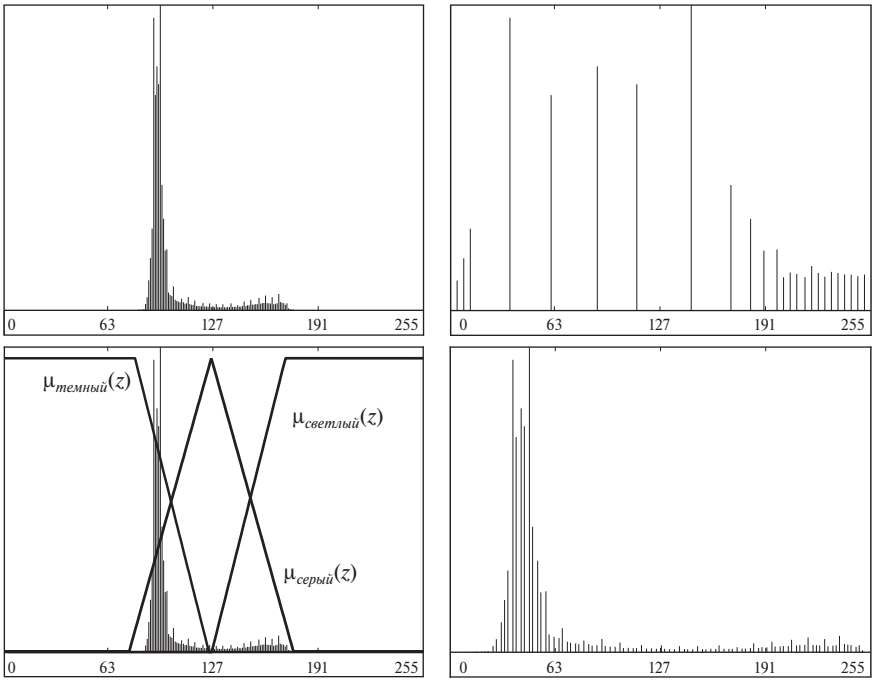


Рис. 3.55. (*а*), (*б*) Гистограммы рис. 3.54(*а*) и (*б*). (*в*) Входная функция принадлежности, наложенная на (*а*). (*в*) Гистограмма рис. 3.54(*в*)

Ценой такого улучшения в результатах является значительное усложнение вычислений. Практическим вариантом, когда важна потоковая скорость обработки многих изображений, может являться применение нечетких методов для определения того, как должна выглядеть гистограмма типичного, хорошо сбалансированного изображения. Затем для получения похожих результатов могут использоваться более быстрые методы, например такие, как задание гистограммы. Для этого следует выполнить отображение гистограммы исходного изображения на одну или больше «идеальных» гистограмм, определенных при помощи нечеткого подхода. ■

3.8.5. Использование нечетких множеств для пространственной фильтрации

В задаче применения нечетких множеств для пространственной фильтрации основной вопрос заключается в определении свойств окрестности, «отражающих» существо фильтров, которые предполагается использовать. Например, рассмотрим задачу обнаружения границ между областями на изображении. Она важна для многих приложений обработки изображений, таких как повышение резкости, как уже рассматривалось ранее в данном разделе, а также при сегментации изображений, как будет обсуждаться в главе 10.

Можно разработать алгоритм выделения границ, основанный на простой нечеткой концепции: *если пиксель принадлежит однородной области, то сделать его белым; иначе сделать его черным*. *Черное* и *белое* здесь суть нечеткие множества. Чтобы выразить концепцию «однородной области» в нечетких терминах, рассмотрим разности яркостей между центральным элементом окрестности и его соседями. Для окрестности 3×3 на рис. 3.56(*а*) разности между центральным элементом (отмеченным как z_5) и каждым соседним в окрестности составляют массив, показанный на рис. 3.56(*б*), где d_i означает разность яркостей i-того соседа и центрального элемента (т. е. $d_i = z_i - z_5$, где z_i — значения яркостей). Простой набор из четырех ЕСЛИ—ТО и одного ИНАЧЕ правил реализует существо нечеткой концепции, приведенной в начале параграфа:

ЕСЛИ d_2 есть *ноль* И d_6 есть *ноль*, ТО z_5 сделать *белым*;
ЕСЛИ d_6 есть *ноль* И d_8 есть *ноль*, ТО z_5 сделать *белым*;
ЕСЛИ d_8 есть *ноль* И d_4 есть *ноль*, ТО z_5 сделать *белым*;
ЕСЛИ d_4 есть *ноль* И d_2 есть *ноль*, ТО z_5 сделать *белым*;
ИНАЧЕ z_5 сделать *черным*.

а *б*

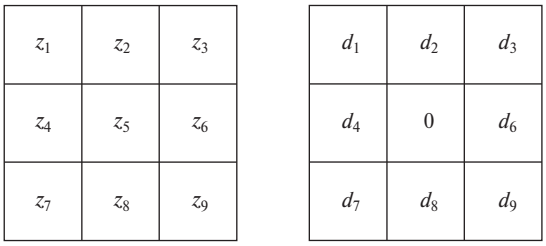


Рис. 3.56. (*а*) Окрестность 3×3 пикселя и (*б*) соответствующие разности яркостей между центральным пикселем и данным соседом. Для простоты обсуждения в рассматриваемом приложении использованы лишь d_2, d_4, d_6 и d_8

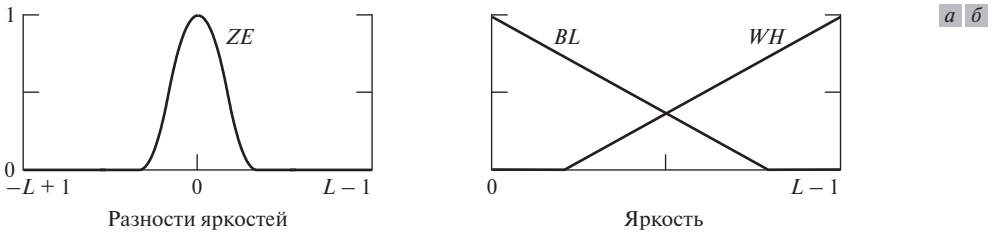


Рис. 3.57. (*а*) Функция принадлежности нечеткого множества *ноль* (*ZE*). (*б*) Функции принадлежности нечетких множеств *черное* (*BL*) и *белое* (*WH*)

Здесь *ноль* также является нечетким множеством. Следствие каждого из правил определяет значение яркости, в которое отображается центральный пиксель (z_5). Т. е. выражение «z_5 сделать *белым*» означает, что яркость пикселя в центре маски отображается в белое. Эти правила просто утверждают, что если яркостные разности, указанные в правилах, равны нулю (в нечетком смысле), то центральный пиксель рассматривается как часть однородной области. Иначе он рассматривается как пиксель границы.

Для простоты обсуждения использованы яркостные разности лишь четырех соседей. Использование восьми соседей является прямым расширением рассмотренного здесь подхода.

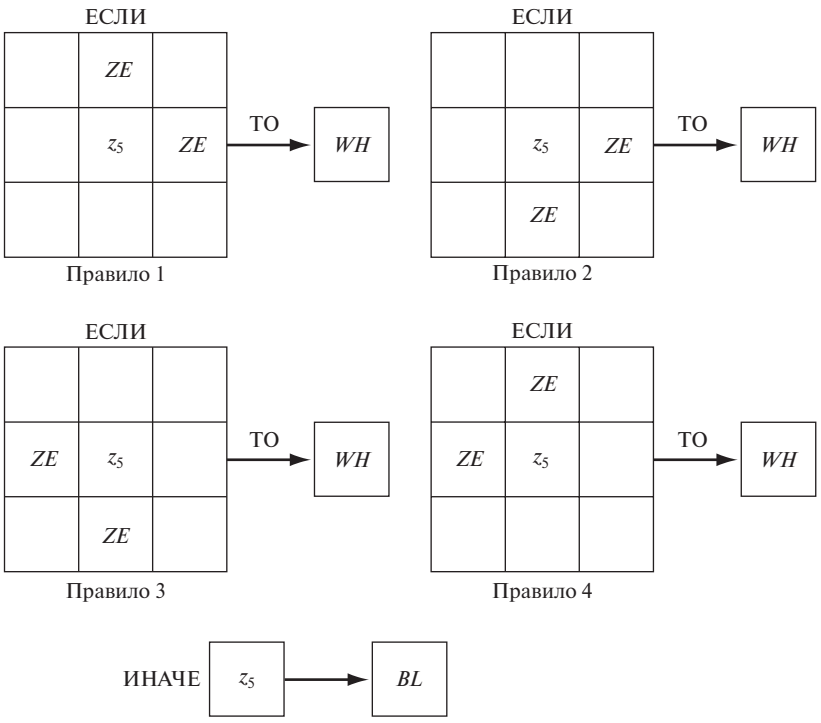


Рис. 3.58. Правила нечеткой логики для обнаружения границ

На рис. 3.57 показаны возможные функции принадлежности для нечетких множеств *ноль* (*ZE*), *черное* (*BL*) и *белое* (*WH*). Заметим, что диапазон независимой переменной нечеткого множества *ZE* с L возможными уровнями яркости составляет $[-L + 1, L - 1]$, поскольку разность яркостей может меняться от $(-L + 1)$ до $(L - 1)$. С другой стороны, диапазон выходных значений яркости составляет $[0, L - 1]$, как и у исходного изображения. На рис. 3.58 схематически показаны правила, сформулированные выше, где квадрат, отмеченный z_5, означает, что яркость центрального пикселя отображается на выходное значение *WH* или *BL*.

Пример 3.20. Иллюстрация улучшения границ областей при помощи пространственной фильтрации, основанной на правилах нечеткой логики.

■ На рис. 3.59(*а*) показан скан головы человека, полученный компьютерным томографом, а на рис. 3.59(*б*) — результат применения только что рассмотренного метода, основанного на нечеткой пространственной фильтрации. Отметим эффективность метода в выделении границ между областями, включая контур мозга (внутренняя серая область на рис. 3.59(*а*)). Однородные области представлены серым, поскольку когда разности яркостей, рассмотренные выше, оказываются около нуля, правила ТО дают сильный отклик. Эти отклики в свою очередь урезают функцию *WH*. Выход (центр тяжести срезанных треугольных областей) является константой между $(L - 1)/2$ и $(L - 1)$, давая тем самым сероватый тон на изображении. Контраст этого изображения может быть существенно улучшен путем растяжения диапазона яркостей. Так изображение на рис. 3.59(*в*) получено при помощи градационной коррекции, определенной уравнениями (2.6-10) и (2.6-11) с $K = L - 1$. Результатом является то, что значения яркостей на рис. 3.59(*в*) занимают весь диапазон от 0 до $(L - 1)$. ■

а б в

Рис. 3.59. (*а*) Скан головы человека при КТ. (*б*) Результат пространственной фильтрации на основе нечеткой логики при использовании функций принадлежности на рис. 3.57 и правил на рис. 3.58. (*в*) Результат после коррекции яркостей. Тонкие черные границы рисунков (*б*) и (*в*) добавлены для наглядности, они не являются частью данных. (Исходное изображение предоставил д-р Дэвид Пикенс, университет Вандербильта)

Заключение

Материалы, которые были только что рассмотрены, представляют типичные примеры современных методов градационных преобразований и пространственной обработки, обычно используемых на практике. Темы, включенные в данную главу, были выбраны по своей значимости как основные, которые могут служить фундаментом в данной развивающейся области. Хотя большинство примеров данной главы касаются улучшения изображений, представленные методы являются вполне общими и встретятся еще в последующих главах в контекстах, совершенно не относящихся к улучшению. В следующей главе вопросы фильтрации будут рассмотрены еще раз, но уже с позиции концепций частотной области. Как будет показано, имеется взаимно однозначное соответствие между линейными пространственными фильтрами, рассмотренными здесь, и фильтрами в частотной области.

Ссылки и литература для дальнейшего изучения

Материалы раздела 3.1 взяты из [Gonzalez, 1986]. Дополнительный материал по разделу 3.2 можно найти в [Schowengerdt, 1983], [Poyton, 1996] и [Russ, 1999]. Касательно оптимизации воспроизведения изображений см. [Tsujii et al., 1998]. Ранние ссылки на гистограммную обработку: [Hummel, 1974], [Gonzalez, Fittes, 1977] и [Woods, Gonzalez, 1981]. В работе [Stark, 2000] сделано интересное обобщение эквализации гистограммы для адаптивного улучшения контраста. Другие подходы к улучшению контраста проиллюстрированы в [Centeno, Haertel, 1997] и [Cheng, Xu, 2000]. По вопросу точного задания гистограммы см. [Coltuc, Bolon, Chassery, 2006]. Расширение локальной эквализации гистограммы рассмотрено в [Caselles et al., 1999] и [Zhu et al., 1999]. Использование локальных статистик для улучшения изображений рассмотрено в [Narendra, Fitch, 1981]. В статье [Kim et al., 1997] представлен интересный подход, в котором для улучшения изображения комбинируются градиент и локальные статистики.

В качестве дополнительного чтения по вопросам линейной пространственной фильтрации и ее применения рекомендуются работы [Umbaugh, 2005], [Jain, 1989] и [Rosenfeld, Kak, 1982]; в них также рассматриваются ранговые фильтры. В статье [Wilburn, 1998] рассматриваются обобщения фильтров, основанных на порядковых статистиках. В книге [Pitas, Venetsanopoulos, 1990] также рассматриваются медианный и другие нелинейные пространственные фильтры. Специальный выпуск журнала [*IEEE Transactions in Image Processing*, 1996] посвящен вопросу нелинейной обработки изображений. Материалы по вопросам фильтрации с подъемом высоких частот взяты из [Schowengerdt, 1983]. Многие из пространственных фильтров, которые были приведены в настоящей главе, будут вновь рассмотрены в связи с вопросами восстановления изображений (глава 5) и обнаружения контуров (глава 10).

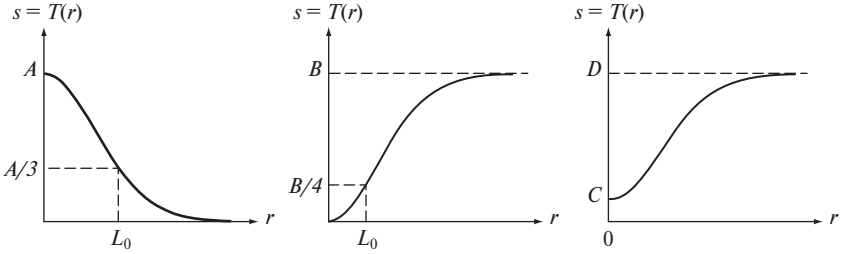
Фундаментальными работами по разделу 3.8 являются три работы Л. А. Заде по вопросам нечеткой логики [Zadeh, 1965, 1973, 1976]. Эти работы хорошо напи-

саны и заслуживают детального прочтения, поскольку они задают основы нечеткой логики и некоторых ее приложений. Обзор большого числа приложений методов нечеткой логики в обработке изображений может быть найден в книге [Kerre, Nachtegael, 2000]. Пример раздела 3.8.4 основан на простом приложении, рассмотренном в [Tizhoosh, 2000]. По поводу дополнительных примеров применения нечеткой логики для градационных преобразований и фильтрации изображений см. соответственно [Patrascu, 2004] и [Nie, Barner, 2006]. Упомянутый набор литературы, начиная с 1965 г. до 2006 г., является хорошей отправной точкой для более детального изучения многих приложений, в которых нечеткие множества могли бы быть использованы для обработки изображений. Программные реализации многих методов, рассмотренных в данной главе, могут быть найдены в [Gonzalez, Woods, Eddins, 2004].

Задачи

★3.1 Найдите однозначную функцию градационного преобразования для расширения диапазона яркостей изображения так, что наименьшая яркость равна C, а наибольшая равна $L - 1$.

3.2 Экспоненты вида $e^{-\alpha r^2}$ с положительной константой α применимы для построения гладкой функции градационной коррекции. Начните с этой базовой функции и найдите функции преобразования, вид которых показан на графиках. Указанные константы являются *входными* параметрами и должны входить в формулу предлагаемых Вами преобразований. (Для упрощения решения на третьей кривой L_0 не является требуемым параметром.)



3.3 ★ **(а)** Найдите непрерывную функцию, реализующую преобразование усиления контраста, показанное графиком на рис. 3.2(*а*). Кроме параметра m, Ваша функция должна включать параметр E для управления наклоном функции при переходе от низких значений яркости к высоким. Ваша функция должна быть нормализована так, что ее минимальное и максимальное значения должны быть равны 0 и 1 соответственно.

(б) Найдите семейство преобразований как функции параметра E для фиксированного значения $m = L/4$, где L — число уровней яркости на изображении.

(в) Чему равно наименьшее значение E, при котором Ваша функция *в действительности* будет выполнять преобразование, показанное

на рис. 3.2(*б*)? Другими словами, Ваша функция не обязана быть ей идентичной, а просто должна давать те же результаты в получении бинарного изображения. Предположите, что изображение является 8-битовым, и пусть $m = 128$. Пусть C есть наименьшее положительное число, представимое в используемом компьютере.

3.4 Предложите набор функций градационных преобразований, позволяющих получить все отдельные битовые плоскости 4-битового одноцветного изображения. (Например, функция преобразования с параметрами $T(r) = 0$ для r в диапазоне [0, 7] и $T(r) = 15$ для r в диапазоне [8, 15] дает изображение 4-й битовой плоскости 8-битового изображения.)

3.5 ★ **(а)** К какому эффекту на гистограмме приведет обнуление половины младших битовых плоскостей изображения?

(б) Как будет выглядеть гистограмма, если, наоборот, обнулить половину старших битовых плоскостей изображения?

★**3.6** Объясните, почему операция эквализации гистограммы в дискретном виде не приводит, вообще говоря, к равномерной гистограмме.

3.7 Предположим, что дискретное изображение было подвергнуто операции эквализации гистограммы. Покажите, что второй проход операции эквализации гистограммы (по изображению с уже эквализованной гистограммой) даст в точности тот же результат, что уже был получен после первого прохода.

3.8 В некоторых приложениях предполагается, что гистограмма входного изображения описывается гауссовой плотностью распределения вероятностей вида

$$p_r(r) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(r-m)^2}{2\sigma^2}},$$

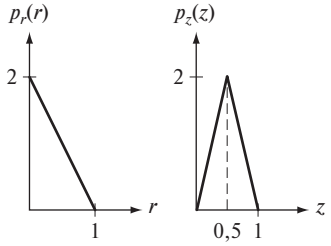
где m и σ суть значения среднего и стандартного отклонения гауссовой ПРВ. Данное приближение позволяет значениям m и σ являться мерами среднего уровня яркости и контраста такого изображения. Какой должна быть функция преобразования для эквализации гистограммы такого изображения?

★**3.9** Считая значения непрерывными, покажите на примере, что возможен случай, при котором функция преобразования, задаваемая уравнением (3.3-4), удовлетворяет условиям (а) и (б) раздела 3.3.1, но обратная к ней функция может являться неоднозначной.

3.10 **(а)** Покажите, что функция дискретного преобразования, задаваемая уравнением (3.3-8) для эквализации гистограммы, удовлетворяет условиям (а) и (б) раздела 3.3.1.

★ **(б)** Покажите, что обратное преобразование, задаваемое уравнением (3.3-9), удовлетворяет условиям (а') и (б') раздела 3.3.1, если ни один из уровней яркости r_k, $k = 0, 1, \ldots, L - 1$ не пропущен.

3.11 Изображение, яркости которого находятся в диапазоне [0, 1], имеет ПРВ $p_r(r)$, вид которой показан первым графиком на рисунке. Необходимо преобразовать уровни яркостей изображения так, чтобы плотность распределения вероятностей $p_z(z)$ преобразованного изображения имела вид, показанный на втором графике. Предполагая значения непрерывными, найдите преобразование (в терминах r и z), решающее поставленную задачу.



★**3.12** Предложите алгоритм обновления локальной гистограммы при передвижении окрестности на один элемент (см. метод локального улучшения в разделе 3.3.3).

3.13 Два изображения, $f(x,y)$ и $g(x,y)$, имеют гистограммы h_f и h_g. Найдите условия, при которых можно выразить через h_f и h_g гистограммы результатов следующих преобразований:

★ **(а)** $f(x,y) + g(x,y)$,
(б) $f(x,y) - g(x,y)$,
(в) $f(x,y) \times g(x,y)$,
(г) $f(x,y) \div g(x,y)$,
(д) $f(x,y) * g(x,y)$.

Объясните, как вычислить гистограммы для каждого из случаев.

3.14 Два изображения, показанные ниже, совершенно различны, однако имеют одинаковые гистограммы. Предположим, каждое из изображений сглажено усредняющей маской 3×3.

(а) Будут ли гистограммы сглаженных изображений по-прежнему одинаковы? Поясните.

(б) Если ответ отрицательный, изобразите обе гистограммы.

3.15 Реализация линейных пространственных фильтров требует перемещения центра маски по изображению и вычисления для каждого из положений маски суммы произведений коэффициентов маски на значения соответствующих пикселей (см. раздел 3.4). Можно реализовать низкочастотный фильтр, все коэффициенты которого равны 1, называемый *однородный усредняющий фильтр* или алгоритм *скользящего среднего.* Он основан на том, что при переходе от точки к точке обновляется только часть вычислений.

★ **(а)** Сформулируйте такой алгоритм для фильтра размерами $n \times n$, демонстрирующий характер взаимосвязи вычислений с последовательностью сканирования, использующейся при передвижении маски по изображению.

(б) Отношение числа операций, требуемых для реализации метода «в лоб», к числу операций, используемых алгоритмом скользящего среднего, называется *эффективностью алгоритма.* Подсчитайте эф-

фективность алгоритма для данного случая и изобразите ее в виде графика зависимости от n для $n > 1$. Коэффициент $1/n^2$ является общим для обоих случаев и поэтому не должен приниматься во внимание. Считайте, что изображение окружено бордюром из нулей достаточной ширины, чтобы не учитывать влияние граничных эффектов при вычислениях.

3.16 ★ **(а)** Предположим, что изображение $f(x, y)$ фильтруется при помощи пространственной маски фильтра $w(x, y)$ и свертка осуществляется согласно уравнению (3.4-2), где маска меньше размеров изображения по обоим направлениям. Важным свойством является то, что если сумма коэффициентов маски равна нулю, то сумма значений всех элементов массива, получающегося в результате свертки, также будет равна нулю. Покажите это. Можете пренебречь вычислительными неточностями, а также можете считать, что изображение окружено рамкой из необходимого числа нулей.

(б) Будет ли результат (а) тем же, если используется корреляционный фильтр, как определено в уравнении (3.4-1)?

3.17 Исследуйте предельные эффекты многократного применения низкочастотного сглаживающего фильтра размерами 3×3 к дискретному изображению. Можете игнорировать влияние границ изображения. Будет ли результат другим, если применять фильтр размерами 5×5?

3.18 ★ **(а)** В разделе 3.5.2 утверждалось, что изолированные темные или светлые (по отношению к фону) кластеры с числом пикселей, меньшим половины площади медианного фильтра, будут при фильтрации удалены (заменены на значение медианы по окружающему фону). Объясните, почему это так, предполагая, что фильтр имеет размеры $n \times n$, где n — нечетное число.

(б) Рассмотрим изображение, содержащее различные наборы кластеров. Предположим, что все точки кластера или светлее, или темнее, чем фон (но не в одном и том же кластере), а также что площадь каждого из кластеров меньше, чем $n^2/2$. При каких условиях относительно n один или больше кластеров перестанут быть изолированными в смысле, указанном в части (а)?

★3.19 **(а)** Разработайте алгоритм для вычисления медианы по окрестности размерами $n \times n$.

(б) Предложите способ обновления значения медианы при передвижении центра окрестности от точки к точке.

3.20 **(а)** В задаче распознавания символов страницы текста редуцируются в двоичную форму с помощью функции порогового преобразования, показанной на рис. 3.2(*б*). Затем применяется процедура утончения линий до размера в один пиксель, которые можно рассматривать как двоичные единицы на нулевом фоне. Ввиду наличия шума процесс бинаризации и утончения приводит к возникновению разрывов в линиях размерами от 1 до 3 элементов. Один из способов «реставрирования» разрывов состоит в сглаживании бинарного изображения усредняющей маской, что позволяет с помощью ненулевых пикселей построить «мостики» через разрывы. Определите минимальный (не-

четный) размер усредняющей маски, при котором решается данная задача.

(б) После устранения разрывов желательно преобразовать изображение пороговой операцией обратно в бинарную форму. Продолжая ответ к задаче **(а)**, скажите, каково должно быть максимальное значение порога, которое не приведет вновь к возникновению разрывов между линиями.

★3.21 Три приведенных изображения были сглажены квадратной усредняющей маской размерами $n = 23$, 25 и 45 соответственно. Вертикальные полоски в левой нижней части изображений **(а)** и **(в)** расфокусированы, но между ними видно четкое разделение. Однако на изображении **(б)** полоски слились, несмотря на то, что размер усредняющей маски значительно меньше, чем на изображении **(в)**. Объясните причину этого.

(а) **(б)** **(в)**

3.22 Рассмотрим задачу, подобную показанной на рис. 3.34, в которой требовалось удалить объекты, укладывающиеся в квадрат размерами $q \times q$ пикселей. Предположим, желательно уменьшить среднюю яркость таких объектов до уровня одной десятой (или более) от первоначального уровня их яркости. Тогда подобные объекты по яркости будут близки к фону и могут быть удалены пороговым преобразованием. Укажите минимальной (нечетный) размер усредняющей маски, которая будет обеспечивать желаемое снижение уровня яркости за один проход.

3.23 К исходному изображению применяется усредняющая маска для подавления шума, а затем маска лапласиана для улучшения мелких деталей. Изменится ли результат, если поменять очередность этих операций?

★3.24 Покажите, что операция лапласиана, заданная уравнением (3.6-3), является изотропной (инвариантной к повороту). Вам потребуются следующие уравнения, связывающие координаты точки при повороте осей на угол θ:

$$x = x' \cos\theta - y' \sin\theta$$
$$y = x' \sin\theta + y' \cos\theta,$$

где (x, y) — значения координат до поворота, а (x', y') — после поворота.

★3.25 На рис. 3.38 видно, что лапласиан с коэффициентом в центре, равным -8, дает более резкий результат, чем с коэффициентом -4. Объясните детально причину.

3.26 В продолжение задачи 3.25.

(а) Получим ли мы более резкий результат, если используем «лапласианоподобную» маску большего размера, скажем, 5×5 элементов с коэффициентом -24 в центре? Объясните в деталях.

(б) Что произойдет, когда размер маски сравняется с размером изображения?

3.27 Задайте маску размерами 5×5 для выполнения за один проход операции нерезкого маскирования. Предположите, что среднее значение получается при помощи гауссова фильтра.

★3.28 Покажите, что результат вычитания лапласиана из изображения пропорционален результату нерезкого маскирования. Используйте определение лапласиана, заданное уравнением (3.6-6).

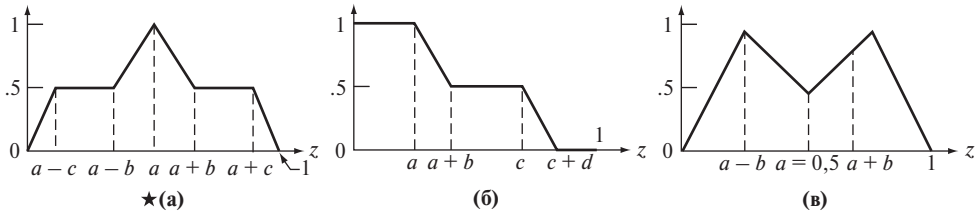
3.29 **(а)** Покажите, что вычисление модуля градиента согласно уравнению (3.6-11) является изотропной операцией (см. задачу 3.24).

(б) Покажите, что свойство изотропности в общем случае нарушается, если вычислять градиент согласно уравнению (3.6-12).

3.30 Для долговременного (круглосуточно в течение 30 дней) контроля одной и той же сцены используется ПЗС-камера. Цифровые изображения фиксируются и передаются в центр обработки каждые 5 минут. Освещение сцены меняется с дневного на искусственное, но так, что сцена всегда освещена и всегда можно сделать снимок. Поскольку уровень освещения достаточен настолько, что яркости объектов сцены всегда остаются на линейном участке характеристики камеры, было принято решение отказаться от использования какого-либо механизма компенсации освещенности в самой камере. Наоборот, решили использовать последующие преобразования при помощи методов обработки изображений и таким путем осуществить нормализацию — приведение изображений к некоторому постоянному уровню освещения. Предложите способ решения этой задачи. Вы свободны в выборе любых желаемых методов, но ясно формулируйте все предположения, которые Вам потребуются для достижения цели.

3.31 Покажите, что точка перегиба на рис. 3.46(*г*) задается координатой $b = (a + c)/2$.

3.32 Используя определения нечетких множеств в разделе 3.8.2 и основные функции принадлежности, представленные на рис. 3.46, сформируйте функции принадлежности, показанные ниже.



★3.33 Каков будет эффект, если увеличить размер окрестности в подходе, использующем нечеткую фильтрацию, который рассмотрен в разделе 3.8.5? Объясните умозаключения, на основе которых строится ответ (для объяснения ответа можно использовать пример).

3.34 Построим систему на основе правил нечеткой логики, предназначенную для уменьшения эффекта импульсных шумов на зашумленном изображении с диапазоном яркостей $[0, L - 1]$. Так же, как и в разделе 3.8.5, для упрощения задачи используем лишь разности d_2, d_4, d_6 и d_8 в окрестности 3×3 элементов. Пусть z_5 — значение яркости в центре окрестности. Соответствующее выход-

ное значение яркости будет равно $z' = z_5 + v$, где v — выход нечеткой системы. Т. е. выход нечеткой системы есть поправочный фактор, использующийся для *уменьшения* эффекта импульсного выброса, который может оказаться в центре окрестности 3×3. Предположим, что импульсные выбросы появляются достаточно далеко друг от друга, поэтому нет необходимости рассматривать случаи попадания в окрестность одновременно нескольких выбросов. Выбросы могут быть как темными, так и светлыми. Везде используется треугольная функция принадлежности.

★ **(а)** Задайте нечеткую формулировку для данной задачи.

★ **(б)** Сформулируйте правила ЕСЛИ—ТО.

(в) Изобразите функции принадлежности графически, как на рис. 3.57.

(г) Изобразите набор правил схематически, как на рис. 3.58.

(д) Постройте упрощенную схему системы с нечеткой логикой по аналогии с показанной на рис. 3.52.

ГЛАВА 4

ФИЛЬТРАЦИЯ В ЧАСТОТНОЙ ОБЛАСТИ

Фильтр — устройство или вещество для подавления или минимизации амплитуды волн или колебаний определенной частоты.

Частота — величина, выражающая число повторений одной и той же последовательности значений периодической функции в единицу времени.

Толковый словарь Вебстера

Введение

Несмотря на значительные усилия, затраченные в предыдущей главе на изучение методов пространственной фильтрации, для всестороннего понимания предмета необходимо практическое представление того, как для фильтрации изображений могут быть использованы преобразование Фурье и частотная область. Для глубокого понимания этих вопросов не обязательно становиться специалистом в области обработки сигналов. Ключ к успеху в том, чтобы сосредоточиться на основных принципах и их связи с цифровой обработкой изображений. Для того чтобы сделать более понятной систему обозначений, являющуюся обычно источником трудностей для начинающих, мы в этой главе обращаем особое внимание на связь характеристик изображения с используемыми при их определении математическими инструментами. Первоочередная цель этой главы — помочь читателю овладеть базовыми знаниями в области Фурье-анализа и его применении для фильтрации изображений. Позднее, в главах 5, 8, 10 и 11, мы обсудим другие применения преобразования Фурье. Мы начнем обсуждение с краткого обзора истории возникновения Фурье-анализа и того воздействия, которое он оказал на математику, естественные науки и технику. Далее мы рассмотрим основы дискретизации функций и шаг за шагом перейдем к выводу одномерного и двумерного преобразования Фурье — основного стержня обработки данных в частотной области. Во время этого обсуждения мы коснемся различных важных аспектов дискретизации, в частности вопроса наложения спектров. Рассмотрение данных вопросов требует понимания частотной области и поэтому глубоко изучается в настоящей главе. Вслед за этим материалом следует постановка задачи фильтрации в частотной области, а также изложение

методов сглаживания и повышения резкости, параллельных методам фильтрации в пространственной области, рассмотренным в главе 3. Завершим главу обсуждением проблем, связанных с применением преобразования Фурье для обработки изображений. Поскольку материалы, изложенные в разделах 4.2—4.4, являются основами, то читатели, хорошо знакомые с вопросами обработки одномерных сигналов, включая преобразование Фурье, дискретизацию, наложение спектров и теорему о свертке, могут перейти к разделу 4.5, где начинается рассмотрение двумерного преобразования Фурье и его применения для цифровой обработки изображений.

4.1. Основы

4.1.1. Краткая история ряда и преобразования Фурье

Французский математик Жан Батист Жозеф Фурье родился в 1768 г. в городе Осер на полпути из Парижа в Дижон. Главное научное достижение Фурье, благодаря которому он остался в памяти потомков, было схематически изложено им в мемуарах 1807 г. и полностью опубликовано в 1822 г. в его книге «*Аналитическая теория тепла*» (*La Théorie Analitique de la Chaleur*). Спустя 55 лет эта книга была переведена на английский Фрименом (см. [Freeman, 1878]). Результат Фурье, относящийся к предмету рассмотрения настоящей главы, состоит, по существу, в том, что любая периодическая функция может быть представлена в виде суммы синусов и/или косинусов различных частот, умноженных на некоторые коэффициенты (теперь эта сумма носит название *ряд Фурье*). Не имеет значения, насколько сложной является функция; если только она периодическая и удовлетворяет некоторым достаточно слабым математическим условиям, эта функция может быть представлена в виде вышеуказанной суммы. В настоящее время это утверждение является общепризнанным, однако в момент своего появления концепция, что сложные функции могли бы быть представлены в виде суммы простых синусов и косинусов, казалась далеко не очевидной (рис. 4.1). Поэтому не удивительно, что идеи Фурье в этом отношении были встречены скептически.

Даже функции, не являющиеся периодическими (но имеющие конечную площадь под графиком[1]), могут быть выражены в виде интеграла от синусов и/или косинусов, умноженных на некоторую весовую функцию. В таком случае мы имеем дело с *преобразованием Фурье*, польза которого во многих теоретических и прикладных задачах оказывается даже большей, чем ряда Фурье. Оба представления обладают важной характерной особенностью. Функция, заданная как рядом, так и преобразованием Фурье, может быть полностью, без потери информации, восстановлена (реконструирована) при помощи некоторой процедуры обращения. Это свойство является одним из наиболее важных свойств рассматриваемых представлений, поскольку оно позволяет работать в «Фурье-области», а затем вернуться в исходную область определения функции без потери какой-либо информации. В конечном счете именно эффективность применения аппарата рядов и преобразования Фурье для решения практических задач превратила его в широко используемый и изучаемый фундаментальный инструмент.

[1] Точнее, конечную площадь под графиком модуля функции. — *Прим. перев.*

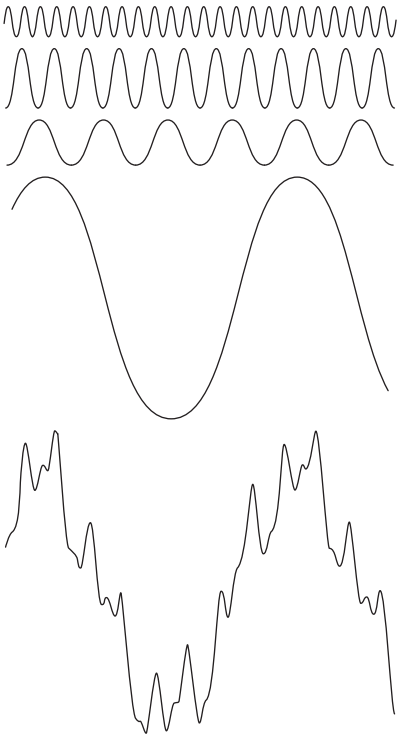


Рис. 4.1. Нижняя функция является суммой четырех расположенных над ней функций. Идея Фурье 1807 г. о том, что периодическая функция может быть представлена в виде взвешенной суммы синусов и косинусов, была встречена скептически

Первоначально идеи Фурье были применены для решения задачи о распространении тепла. Это дало возможность представить дифференциальные уравнения, описывающие тепловой поток, в таком виде, который позволил впервые получить их решения. На протяжении последнего века, и особенно в последние 50 лет, идеи Фурье привели к расцвету целых научных дисциплин и даже промышленных отраслей. Наступление эпохи ЭВМ и открытие алгоритма быстрого преобразования Фурье (БПФ) в начале 60-х годов[2] (или немного позднее) произвели революцию в области обработки сигналов. Эти две основные технологии впервые сделали возможным обработку огромной совокупности сигналов исключительной важности в разных сферах человеческой деятельности — от медицинской диагностики до новейших средств электронной связи.

Поскольку мы будем иметь дело только с функциями (изображениями) конечной протяженности, нас будет интересовать именно преобразование Фурье. Материал следующего раздела знакомит читателя с преобразованием Фурье и частотной областью. Показано, что методы Фурье-анализа дают ясные по смыслу и практичные способы изучения и реализации совокупности подходов для обработки изображений. В некоторых случаях эти подходы аналогичны подходам, развитым в главе 3.

4.1.2. О примерах, приводимых в данной главе

Также как и в главе 3, большинство примеров фильтрации изображений, включенных в настоящую главу, относятся к улучшению изображений. Например, сглаживание и повышение резкости традиционно ассоциируются с улучшением изображений как методы манипулирования контрастом. Как правило, новички в области обработки изображений находят приложения, в которых используется улучшение, интересными и сравнительно простыми для понимания. Таким образом, использование в настоящей главе примеров из области улучшения изображений не только освобождает лишнюю главу в книге, но, что более важно, является эффективным способом ознакомления новичков с методами фильтрации в частотной области. Использование частотных методов в других приложениях рассматривается в главах 5, 8, 10 и 11.

---

[2] См. прим. 26. — *Прим. перев.*

4.2. Предварительные понятия

Чтобы упростить продвижение идей, обсуждаемых в данной главе, прервемся ненадолго для введения нескольких базовых понятий, лежащих в основе материалов, которые излагаются в дальнейших разделах.

4.2.1. Комплексные числа

Комплексное число C определяется как

$$C = R + iI\,, \tag{4.2-1}$$

где R и I являются действительными числами, а i — мнимое число, равное квадратному корню из –1; т.е. $i = \sqrt{-1}$. Здесь R обозначает *действительную* (*real*) *часть* комплексного числа, а I — его *мнимую* (*imaginary*) *часть*. Число, *комплексно сопряженное* к C и обозначаемое C^*, определяется как

$$C^* = R - iI\,. \tag{4.2-2}$$

Комплексные числа могут быть геометрически представлены как точки на *комплексной плоскости*, абсцисса которой является *действительной осью* (ось действительных чисел, R), а ордината — *мнимой осью* (ось мнимых чисел, I). Таким образом комплексное число $R + iI$ является точкой (R, I) в прямоугольной системе координат комплексной плоскости.

Иногда полезно представлять комплексные числа в полярной системе координат:

$$C = |C|(\cos\theta + i\sin\theta)\,, \tag{4.2-3}$$

где $|C| = \sqrt{R^2 + I^2}$ — длина вектора из начала координат в точку (R, I), а θ — угол между этим вектором и осью действительных чисел. Начертив простую диаграмму из действительной и мнимой осей с вектором в точку (R, I) в первом квадранте, легко видеть, что $\operatorname{tg}\theta = (I/R)$, т.е. $\theta = \operatorname{arctg}(I/R)$. Функция arctg возвращает значение углов в диапазоне $[-\pi/2, \pi/2]$, но поскольку I и R могут независимо быть как положительными, так и отрицательными, нам необходим полный диапазон углов $[-\pi, \pi]$. Это достигается простым учетом знаков I и R перед вычислением θ. Многие из языков программирования делают это автоматически при помощи так называемых *четырехквадрантных функций* арктангенса. Так, например, MATLAB имеет для этого функцию atan2(*Imag*, *Real*).

Использование формулы Эйлера

$$e^{i\theta} = \cos\theta + i\sin\theta\,, \tag{4.2-4}$$

где $e = 2{,}71828...$, дает следующее привычное представление комплексного числа в полярных координатах:

$$C = |C|e^{i\theta}\,, \tag{4.2-5}$$

где C и θ те же самые, что и выше. Например, полярное представление комплексного числа $1 + i2$ будет $\sqrt{5}e^{i\theta}$, где $\theta = 64{,}4°$ или 1,1 радиана. Предыдущие выражения применимы также и к комплексным функциям. Так, комплексная

функция $F(u)$ переменной u может быть выражена как сумма $F(u) = R(u) + iI(u)$, где $R(u)$ и $I(u)$ — действительная и мнимая компоненты функции. Как и ранее, комплексно сопряженной функцией будет $F(u) = R(u) - iI(u)$, амплитуда равна $|F(u)| = \sqrt{R(u)^2 + I(u)^2}$, а угол $\theta(u) = \arctg(I(u)/R(u))$. Мы вернемся несколько раз к комплексным числам на протяжении этой и следующей глав.

4.2.2. Ряды Фурье

Как отмечено в разделе 4.1.1, функция $f(t)$ непрерывной переменной t, имеющая период T, может быть выражена в виде суммы синусов и косинусов, умноженных на соответствующие коэффициенты. Данная сумма, известная как *ряд Фурье*, имеет вид

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{i\frac{2\pi n}{T}t}, \tag{4.2-6}$$

где коэффициенты c_n равны

$$c(n) = \frac{1}{T}\int_{-T/2}^{T/2} f(t)e^{-i\frac{2\pi n}{T}t}dt \quad \text{для } n = 0, \pm 1, \pm 2, ... \tag{4.2-7}$$

Тот факт, что выражение (4.2-6) есть разложение на синусы и косинусы, следует из формулы Эйлера (4.2-4). Мы еще вернемся к ряду Фурье позже в данном разделе.

4.2.3. Импульсы и их свойство отсеивания

Концепция импульса и его свойств отсеивания является центральной в изучении линейных систем и преобразования Фурье. *Единичный импульс* непрерывной переменной t, локализованный в точке $t = 0$ и обозначаемый $\delta(t)$, *определяется* как

$$\delta(t) = \begin{cases} \infty & \text{если } t = 0, \\ 0 & \text{если } t \neq 0, \end{cases} \tag{4.2-8а}$$

> Импульс не является функцией в обычном смысле. Более точным названием для него является распределение или обобщенная функция. Однако часто можно встретить в литературе названия «импульсная функция», «дельта-функция» и «дельта-функция Дирака», несмотря на неправильное употребление.

удовлетворяющий также тождеству

$$\int_{-\infty}^{\infty} \delta(t)dt = 1. \tag{4.2-8б}$$

Физически, если интерпретировать t как время, импульс может быть представлен как всплеск бесконечной амплитуды и нулевой длительности, имеющий единичную площадь. В связи с интегрированием импульс имеет так называемое *свойство отсеивания*:

> «Отсеивать» буквально означает «разделять или выделять путем пропускания через сито».

$$\int_{-\infty}^{\infty} f(t)\delta(t)dt = f(0), \tag{4.2-9}$$

при условии, что $f(t)$ непрерывна в точке $t = 0$, — условии, обычно удовлетворяемом на практике. Отсеивание просто дает на выходе *значение* функции $f(t)$ в *точке локализации* импульса (например в начале координат, согласно предыдущему равенству). Более общее выражение свойства отсеивания касается импульса, локализованного в произвольной точке t_0, обозначаемого $\delta(t - t_0)$. В этом случае свойство отсеивания приобретает вид

$$\int_{-\infty}^{\infty} f(t)\delta(t-t_0)dt = f(t_0), \tag{4.2-10}$$

что дает на выходе значение функции в точке локализации импульса t_0. Например, если $f(t) = \cos(t)$, используя импульс $\delta(t - \pi)$, получим результат: $f(\pi) = \cos(\pi) = -1$. Сила концепции отсеивания скоро станет совершенно очевидной.

Пусть x представляет собой *дискретную* переменную. *Единичный дискретный импульс* $\delta(t)$ в контексте дискретных систем служит достижению тех же целей, что и импульс $f(t)$ при работе с непрерывными переменными. $\delta(t)$ определяется следующим образом:

$$\delta(x) = \begin{cases} 1 & \text{если } x = 0, \\ 0 & \text{если } x \neq 0. \end{cases} \tag{4.2-11а}$$

Очевидно, это определение также удовлетворяет дискретному аналогу выражения (4.2-8б):

$$\sum_{x=-\infty}^{\infty} \delta(x) = 1. \tag{4.2-11б}$$

Свойство отсеивания для дискретной переменной имеет вид

$$\sum_{x=-\infty}^{\infty} f(x)\delta(x) = f(0), \tag{4.2-12}$$

или в более общем виде, с использованием дискретного импульса, локализованного в точке $x = x_0$:

$$\sum_{x=-\infty}^{\infty} f(x)\delta(x-x_0) = f(x_0). \tag{4.2-13}$$

Как и ранее, видно, что в дискретном случае свойство отсеивания просто возвращает значение функции в точке импульса. На рис. 4.2 представлена диаграмма дискретного единичного импульса. В отличие от своего непрерывного аналога, дискретный импульс является обычной функцией.

Особенный интерес представляет *последовательность импульсов* $s_{\Delta T}(t)$, определяемая как сумма бесконечно большого числа *периодически* расположенных импульсов, отстоящих друг от друга на ΔT:

$$s_{\Delta T}(t) = \sum_{n=-\infty}^{\infty} \delta(t - n\Delta T). \tag{4.2-14}$$

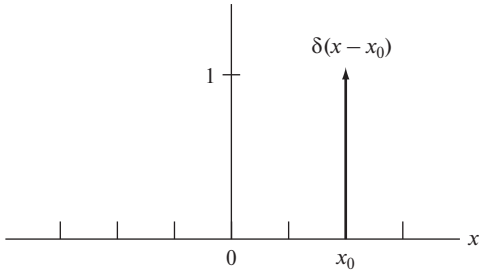


Рис. 4.2. Единичный дискретный импульс, локализованный в точке $x = x_0$. Переменная x является дискретной и δ равна 0 везде, кроме $x = x_0$

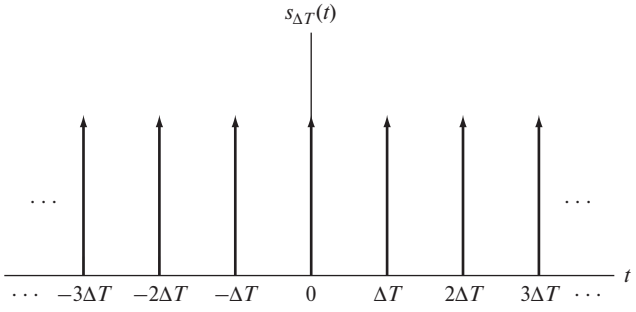


Рис. 4.3. Последовательность импульсов

Последовательность импульсов представлена на рис. 4.3. Импульсы могут быть как непрерывными, так и дискретными.

4.2.4. Преобразование Фурье функции одной непрерывной переменной

Преобразование Фурье (*Фурье-образ*) непрерывной функции $f(t)$ одной непрерывной переменной t, обозначаемое $\mathfrak{F}$, *определяется* выражением[3]

$$\mathfrak{F}\{f(t)\} = \int_{-\infty}^{\infty} f(t)e^{-i2\pi\mu t}dt\,, \tag{4.2-15}$$

где μ также является непрерывной переменной. Поскольку при интегрировании переменная t исключается, $\mathfrak{F}\{f(t)\}$ оказывается функцией только от μ. Мы явно указываем на этот факт, записывая преобразование Фурье как $\mathfrak{F}\{f(t)\} = F(\mu)$;

[3] Условия существования преобразования Фурье в общем случае сформулировать сложно [Champeney, 1987], но достаточным условием его существования является конечность интеграла абсолютной величины $f(t)$ либо интеграла квадрата $f(t)$. На практике вопрос существования редко является проблемой, за исключением идеализированных сигналов, таких как синусоиды, которые продолжаются в бесконечность. Для их рассмотрения используют обобщенные импульсные функции. Нас в основном интересует пара дискретных преобразований Фурье, которая, как будет вскоре видно, гарантированно существует для всех финитных функций.

а значит, преобразование Фурье от $f(t)$ может быть для удобства представлено в виде[4]

$$F(\mu) = \int_{-\infty}^{\infty} f(t) e^{-i2\pi\mu t} dt . \qquad (4.2\text{-}16)$$

И наоборот, зная $F(\mu)$, можно восстановить $f(t)$, используя обратное преобразование Фурье $f(t) = \Im^{-1}\{F(\mu)\}$, записываемое как

$$f(t) = \int_{-\infty}^{\infty} F(\mu) e^{i2\pi\mu t} d\mu , \qquad (4.2\text{-}17)$$

где мы использовали тот факт, что переменная μ исключается при обратном преобразовании, и записали просто $f(t)$ вместо более громоздкого $f(t) = \Im^{-1}\{F(\mu)\}$. Выражения (4.2-16) и (4.2-17) составляют так называемую *пару преобразований Фурье*, а входящие в них функции образуют *Фурье-пару*. Они демонстрируют важный факт, указанный в разделе 4.1, что, зная Фурье-образ, можно получить исходную функцию.

Используя формулу Эйлера, можно записать (4.2-16) в виде

$$F(\mu) = \int_{-\infty}^{\infty} f(t)\left[\cos(2\pi\mu t) - i\sin(2\pi\mu t)\right] dt . \qquad (4.2\text{-}18)$$

Видно, что если $f(t)$ является действительной функцией, то ее Фурье-образ вообще говоря, будет комплексным. Отметим, что преобразование Фурье является разложением $f(t)$, умноженным на синусоидальные члены, частоты которых определяются значениями μ (как уже говорилось, переменная t исключается при интегрировании). Поскольку единственной переменной, остающейся после интегрирования, является частота, то область определения Фурье-образа называют *частотной областью*. Более детально частотная область и ее свойства будут рассматриваться в этой главе позже. В наших обсуждениях t может представлять любую непрерывную переменную, а единица измерения μ зависит от единицы измерения t. Например, если t представляет время в секундах, то μ будет измеряться в числе периодов в секунду, т. е. в герцах. Если t представляет расстояние в метрах, то μ будет измеряться в числе периодов в метре (величине, обратной длине волны) и так далее. Другими словами, единицей измерения в частотной области является число колебаний на единицу независимой переменной исходной функции.

> Для согласованности терминологии, используемой как в предыдущих двух главах, так и в дальнейшем в связи с изображениями, будем называть область определения переменной t в общем случае пространственной областью.

Пример 4.1. Получение Фурье-преобразования прямоугольной функции.

■ Преобразование Фурье прямоугольной функции, представленной на рис. 4.4(*a*), легко получить из выражения (4.2-16)

[4] Наряду с приведенной формой распространены другие определения преобразования Фурье, отличающиеся выбором коэффициентов в показателе экспоненты, а также перед интегралом. Все свойства при этом сохраняются, но вид каких-то формул может измениться. — *Прим. перев.*

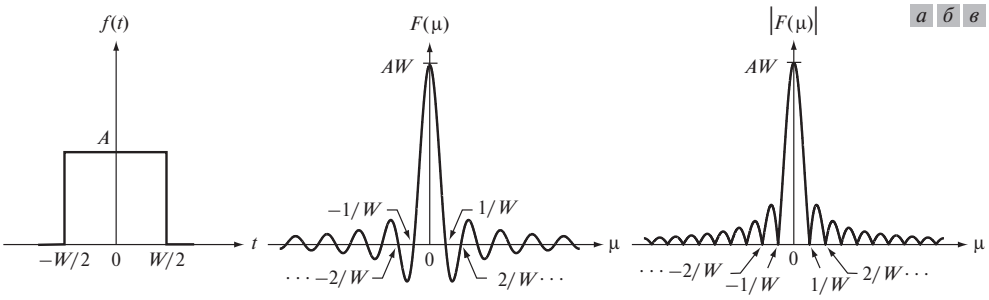


Рис. 4.4. (*а*) Простая прямоугольная функция; (*б*) ее Фурье-образ; (*в*) ее спектр. Все функции продолжаются в бесконечность в обоих направлениях

$$F(\mu)=\int_{-\infty}^{\infty} f(t)e^{-i2\pi\mu t}dt=\int_{-W/2}^{W/2} Ae^{-i2\pi\mu t}dt=\frac{-A}{i2\pi\mu}\left[e^{-i2\pi\mu t}\right]_{-W/2}^{W/2}=$$

$$=\frac{-A}{i2\pi\mu}\left[e^{-i\pi\mu W}-e^{i\pi\mu W}\right]=\frac{A}{i2\pi\mu}\left[e^{i\pi\mu W}-e^{-i\pi\mu W}\right]=AW\frac{\sin(\pi\mu W)}{(\pi\mu W)},$$

где использовалось тригонометрическое тождество $\sin\theta=(e^{i\theta}-e^{-i\theta})/2i$. В этом случае комплексные члены преобразования Фурье прекрасно преобразуются в функцию действительного синуса. Результат, полученный на последнем шаге выражения, известен как функция *sinc* (кардинальный синус):

$$\mathrm{sinc}(m)=\frac{\sin(\pi m)}{(\pi m)}, \tag{4.2-19}$$

где $\mathrm{sinc}(0)=1$ и $\mathrm{sinc}(m)=0$ для всех остальных *целых* значений m. На рис. 4.4(*б*) показан график $F(\mu)$.

В общем случае преобразование Фурье содержит комплексные члены, поэтому для визуализации обычно используют амплитуду преобразования (являющуюся действительной величиной), называемую Фурье-спектром или частотным спектром:

$$|F(\mu)|=AW\left|\frac{\sin(\pi\mu W)}{(\pi\mu W)}\right|.$$

График $|F(\mu)|$ как функции частоты показан на рис. 4.4(*в*). Важнейшие свойства, которые необходимо отметить, состоят в том, что расстояние между нулями как функции $F(\mu)$, так и $|F(\mu)|$, обратно пропорциональны W — ширине прямоугольной функции; что высота лепестков уменьшается как функция расстояния от начала координат; а также что функция продолжается в бесконечность как для положительных, так и для отрицательных значений μ. Как будет видно позже, эти свойства весьма помогают в интерпретации спектров двумерных Фурье-образов изображений. ■

Пример 4.2. Фурье-преобразования импульса и последовательности импульсов.

■ Фурье-преобразование единичного импульса, расположенного в начале координат, вытекает из выражения (4.2-16)

$$F(\mu)=\int_{-\infty}^{\infty}\delta(t)e^{-i2\pi\mu t}dt=\int_{-\infty}^{\infty}e^{-i2\pi\mu t}\delta(t)dt=e^{-i2\pi\mu\cdot 0}=e^{0}=1,$$

где третий шаг следует из свойства отсеивания в выражении (4.2-9). Таким образом, Фурье-образ импульса, расположенного в начале координат пространственной области, является константой в частотной области. Подобным образом Фурье-образ импульса, расположенного в точке $t = t_0$, будет

$$F(\mu)=\int_{-\infty}^{\infty}\delta(t-t_0)e^{-i2\pi\mu t}dt=\int_{-\infty}^{\infty}e^{-i2\pi\mu t}\delta(t-t_0)dt=e^{-i2\pi\mu t_0}=\cos(2\pi\mu t_0)-i\sin(2\pi\mu t_0),$$

где третий шаг следует из свойства отсеивания в выражении (4.2-10), а последний — из формулы Эйлера. Последние два выражения являются эквивалентными представлениями единичной окружности с центром в начале координат комплексной плоскости.

В разделе 4.3 нам потребуется воспользоваться Фурье-преобразованием периодической последовательности импульсов. Выполнение этого преобразования не настолько очевидно, как только что было показано для одиночных импульсов. Однако понимание того, как осуществить преобразование последовательности импульсов, является весьма важным, поэтому мы потратим сейчас время на детальное рассмотрение этой процедуры.

Заметим, что единственная разница в *форме* выражений (4.2-16) и (4.2-17) состоит в знаке показателя экспоненты. Тем самым, если функция $f(t)$ имеет преобразование Фурье $F(\mu)$, то, значит, выразив последнюю относительно t, т. е. в виде $F(t)$, получим, что ее преобразование Фурье будет равно $f(-\mu)$. Из данного свойства *симметрии*, а также (как было показано выше) того, что Фурье-преобразование импульса $\delta(t - t_0)$ равно $e^{-i2\pi\mu t_0}$, вытекает, что функция $e^{-i2\pi t_0 t}$ имеет преобразование $\delta(-\mu - t_0)$. Делая замену $-t_0 = a$, получим, что Фурье-преобразование от $e^{i2\pi at}$ будет $\delta(-\mu + a) = \delta(a - \mu)$. Последнее действие справедливо, поскольку $\delta(t)$ — функция симметричная, т. е. $\delta(t) = \delta(-t)$.

Последовательность импульсов $s_{\Delta T}(t)$ в выражении (4.2-14) является периодической с периодом ΔT; как уже известно из раздела 4.2.2, она может быть выражена рядом Фурье:

$$s_{\Delta T}(t)=\sum_{n=-\infty}^{\infty}c_n e^{i\frac{2\pi n}{\Delta T}t},$$

где

$$c_n=\frac{1}{T}\int_{-\Delta T/2}^{\Delta T/2}s_{\Delta T}(t)e^{-i\frac{2\pi n}{\Delta T}t}dt.$$

Взглянув на рис. 4.3, видно, что в интервал $[-\Delta T/2, \Delta T/2]$, по которому берется интеграл, попадает лишь один импульс $s_{\Delta T}(t)$, находящийся в начале координат. Таким образом, предыдущее выражение равно

$$c_n=\frac{1}{T}\int_{-\Delta T/2}^{\Delta T/2}s_{\Delta T}(t)e^{-i\frac{2\pi n}{\Delta T}t}dt=\frac{1}{\Delta T}e^{0}=\frac{1}{\Delta T}.$$

Разложение в ряд Фурье тогда принимает вид

$$s_{\Delta T}(t) = \frac{1}{\Delta T}\sum_{n=-\infty}^{\infty} e^{i\frac{2\pi n}{\Delta T}t}.$$

Цель состоит в получении Фурье-преобразования этого выражения. Поскольку суммирование есть линейный процесс, то Фурье-преобразование суммы равно сумме преобразований отдельных слагаемых. Эти слагаемые являются экспоненциальными функциями, и ранее было показано, что

$$\Im\{e^{i\frac{2\pi n}{\Delta T}t}\} = \delta\left(\mu - \frac{n}{\Delta T}\right).$$

Таким образом $S(\mu)$, Фурье-преобразование периодической последовательности импульсов $s_{\Delta T}(t)$ будет равно

$$S(\mu) = \Im\{s_{\Delta T}(t)\} = \Im\left\{\frac{1}{\Delta T}\sum_{n=-\infty}^{\infty} e^{i\frac{2\pi n}{\Delta T}t}\right\} = \frac{1}{\Delta T}\Im\left\{\sum_{n=-\infty}^{\infty} e^{i\frac{2\pi n}{\Delta T}t}\right\} = \frac{1}{\Delta T}\sum_{n=-\infty}^{\infty}\delta\left(\mu - \frac{n}{\Delta T}\right).$$

Этот фундаментальный результат показывает, что Фурье-преобразование последовательности импульсов с периодом ΔT также является последовательностью импульсов, период которой $1/\Delta T$. Эта обратная пропорциональность между периодами $s\Delta T(t)$ и $S(\mu)$ аналогична тому, что было показано на рис. 4.4 в связи с прямоугольной функцией и ее преобразованием. Данное свойство играет фундаментальную роль в оставшейся части настоящей главы. ■

4.2.5. Свертка

Для дальнейшего построения нужен еще один блок. Общее представление о свертке было дано в разделе 3.4.2, где мы показали, что при свертке двух функций одна из них переворачивается (поворотом на 180° относительно ее начала координат) и затем скользит (постепенно смещается) по другой. Для каждого значения смещения выполняются вычисления, которые в случае, рассматриваемом в главе 3, являлись суммой произведений двух функций. В данный момент нас интересует свертка двух *непрерывных* функций $f(t)$ и $h(t)$ одной непрерывной переменной t, поэтому вместо суммирования необходимо использовать интегрирование. Свертка двух таких функций, обозначаемая, как и ранее, оператором ★, *определяется* следующим образом:

$$f(t)\star h(t) = \int_{-\infty}^{\infty} f(\tau)h(t-\tau)d\tau, \qquad (4.2\text{-}20)$$

где знак минус обеспечивает только что упомянутый переворот, t есть параметр смещения, необходимый для определения положения одной функции относительно другой, а τ — переменная интегрирования. Пока здесь предполагается, что функции определены на интервале от $-\infty$ до ∞.

Механизм выполнения свертки излагался в разделе 3.4.2, пояснения будут продолжены позже в настоящей главе, а также в главе 5. В данный момент нас интересует нахождение преобразования Фурье уравнения (4.2-20). Начнем с формулы (4.2-15):

$$\mathfrak{F}\{f(t)\star h(t)\} = \int_{-\infty}^{\infty}\left[\int_{-\infty}^{\infty} f(\tau)h(t-\tau)d\tau\right]e^{-i2\pi\mu t}dt = \int_{-\infty}^{\infty} f(\tau)\left[\int_{-\infty}^{\infty} h(t-\tau)e^{-i2\pi\mu t}dt\right]d\tau\,.$$

Член в квадратных скобках — Фурье-преобразование от $h(t - \tau)$. Ниже в настоящей главе будет показано, что $\mathfrak{F}\{h(t - \tau)\} = H(\mu)e^{-i2\pi\mu\tau}$, где $H(\mu)$ — Фурье-преобразование от $h(t)$. Используя этот факт, последнее выражение может быть преобразовано:

$$\mathfrak{F}\{f(t)\star h(t)\} = \int_{-\infty}^{\infty} f(\tau)\left[H(\mu)e^{-i2\pi\mu\tau}\right]d\tau = H(\mu)\int_{-\infty}^{\infty} f(\tau)e^{-i2\pi\mu\tau}d\tau = H(\mu)F(\mu)\,.$$

Тот же самый результат можно получить, если изменить порядок $f(t)$ и $h(t)$ на обратный; т. е. свертка является коммутативной операцией.

В разделе 4.2.4 мы назвали область определения t пространственной областью, а область определения μ — частотной областью. Тем самым последнее выражение говорит, что Фурье-образ свертки двух функций в пространственной области эквивалентен произведению в частотной области Фурье-образов этих двух функций. И наоборот, если имеется произведение двух Фурье-образов, можно получить свертку в пространственной области путем вычисления обратного преобразования Фурье. Другими словами, $f(t)\star h(t)$ и $H(\mu)F(\mu)$ образуют *Фурье-пару*. Этот результат составляет первую половину *теоремы о свертке* и записывается в виде

$$f(t)\star h(t) \Leftrightarrow H(\mu)F(\mu)\,. \tag{4.2-21}$$

Двойная стрелка используется для указания, что выражение в правой части получается Фурье-преобразованием выражения в левой, и наоборот, выражение в левой части получается *обратным* Фурье-преобразованием выражения в правой части.

Выполняя аналогичные преобразования, можно получить вторую половину теоремы о свертке:

$$f(t)h(t) \Leftrightarrow H(\mu)\star F(\mu)\,, \tag{4.2-22}$$

которая утверждает, что свертка в частотной области аналогична умножению в пространственной области; обе части связаны прямым и обратным преобразованиями Фурье соответственно. Как будет показано ниже в данной главе, теорема о свертке является основой для изучения вопросов фильтрации в частотной области.

4.3. Дискретизация и преобразование Фурье дискретных функций

Чтобы сформулировать основу для математического выражения дискретизации, в данном разделе будет использоваться концепция из раздела 4.2. Начиная

с основных принципов, это приведет нас к преобразованию Фурье дискретных функций.

4.3.1. Дискретизация

Для обработки на компьютере непрерывные функции должны быть преобразованы в последовательность дискретных значений. Как пояснялось в разделе 2.4, это осуществляется при помощи операций дискретизации и квантования. В последующем обсуждении дискретизация будет рассмотрена более подробно.

> Взятие отсчетов на расстоянии ΔT друг от друга задает *частоту дискретизации* равной $1/\Delta T$. Если ΔT определяется в секундах, то частота дискретизации будет измеряться в числе отсчетов в секунду. Если ΔT определяется в метрах, то частота дискретизации будет измеряться в метрах в секунду и т. д.

Используя рис. 4.5 в качестве иллюстрации, рассмотрим непрерывную функцию $f(t)$, которую требуется дискретизовать по непрерывной переменной t с интервалом (шагом) дискретизации ΔT. Предполагается, что функции определены для аргумента t от $-\infty$ до ∞. Одним из способов осуществить дискретизацию является умножение $f(t)$ на *функцию дискретизации*, являющуюся последовательностью отдельных импульсов (4.2-14), отстоящих друг от друга на ΔT, как это обсуждалось в разделе 4.2.3, т. е.

$$\tilde{f}(t) = f(t)s_{\Delta T}(t) = \sum_{n=-\infty}^{\infty} f(t)\delta(t - n\Delta T), \qquad (4.3\text{-}1)$$

где $\tilde{f}(t)$ означает уже дискретизованную функцию. Каждый компонент суммирования является импульсом с весовым вкладом, равным значению $f(t)$ в точке импульса, как это показано на рис. 4.5(*в*). Значение каждого отсчета определяется «весом» импульса, который получается при интегрировании. Таким образом, значение f_k произвольного отсчета в последовательности дается формулой

$$f_k = \int_{-\infty}^{\infty} f(t)\delta(t - k\Delta T)dt = f(k\Delta T), \qquad (4.3\text{-}2)$$

где использовалось отсеивающее свойство δ в равенстве (4.2-10). Выражение (4.3-2) справедливо для всех допустимых целых значений $k = ..., -2, -1, 0, 1, 2, ...$. На рис. 4.5(*г*) показан результат, состоящий из множества равноотстоящих отсчетов исходной функции.

4.3.2. Преобразование Фурье дискретизованных функций

Пусть $F(\mu)$ означает Фурье-преобразование непрерывной функции $f(t)$. Как обсуждалось в предыдущем разделе, соответствующая дискретизованная функция $\tilde{f}(t)$ есть произведение $f(t)$ и последовательности импульсов. Из теоремы о свертке (раздел 4.2.5) известно, что Фурье-преобразование произведения двух функций в пространственной области есть свертка Фурье-образов обеих функций в частотной области. Таким образом, Фурье-преобразованием $\tilde{F}(\mu)$ дискретизованной функции $\tilde{f}(t)$ является выражение:

а
б
в
г

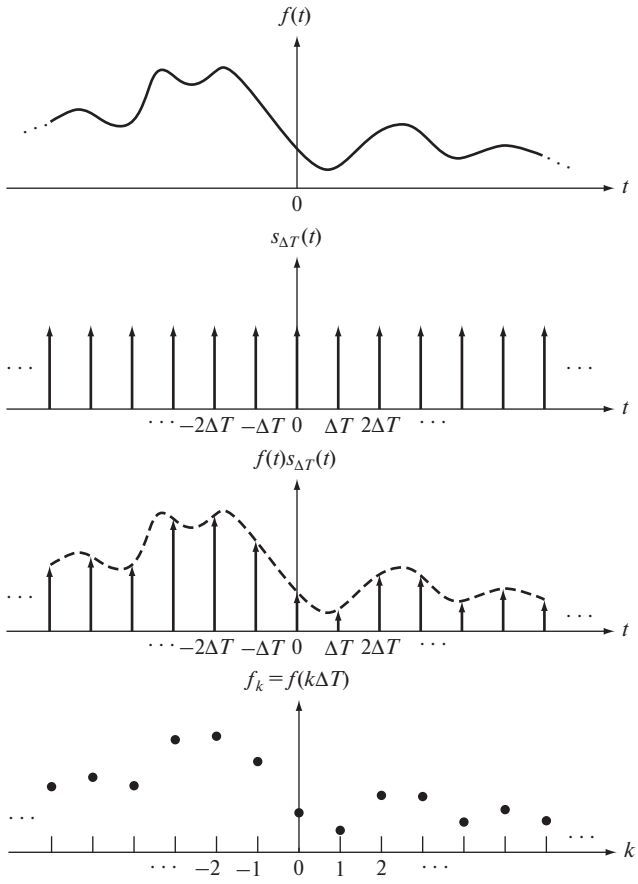


Рис. 4.5. (*а*) Непрерывная функция. (*б*) Последовательность импульсов, используемая для моделирования процесса дискретизации. (*в*) Дискретизованная функция, полученная как произведение (*а*) на (*б*). (*г*) Дискретизованные значения, полученные интегрированием и использованием отсеивающего свойства импульса. (Пунктирная линия на (*в*) приведена для иллюстрации; она не является частью данных)

$$\tilde{F}(\mu) = \Im\{\tilde{f}(t)\} = \Im\{f(t)s_{\Delta T}(t)\} = F(\mu) \bigstar S(\mu)\,, \tag{4.3-3}$$

где из примера 4.2

$$S(\mu) = \frac{1}{\Delta T}\sum_{n=-\infty}^{\infty}\delta\left(\mu - \frac{n}{\Delta T}\right) \tag{4.3-4}$$

является Фурье-преобразованием последовательности импульсов $s_{\Delta T}(t)$. Свертку $F(\mu)$ и $S(\mu)$ получим непосредственно из определения свертки (4.2-20):

$$\begin{aligned}\tilde{F}(\mu) = F(\mu)\bigstar S(\mu) &= \int_{-\infty}^{\infty} F(\tau)S(\mu-\tau)d\tau = \frac{1}{\Delta T}\int_{-\infty}^{\infty} F(\tau)\sum_{n=-\infty}^{\infty}\delta\left(\mu-\tau-\frac{n}{\Delta T}\right)d\tau = \\ &= \frac{1}{\Delta T}\sum_{n=-\infty}^{\infty}\int_{-\infty}^{\infty} F(\tau)\delta\left(\mu-\tau-\frac{n}{\Delta T}\right)d\tau = \frac{1}{\Delta T}\sum_{n=-\infty}^{\infty} F\left(\mu-\frac{n}{\Delta T}\right),\end{aligned} \tag{4.3-5}$$

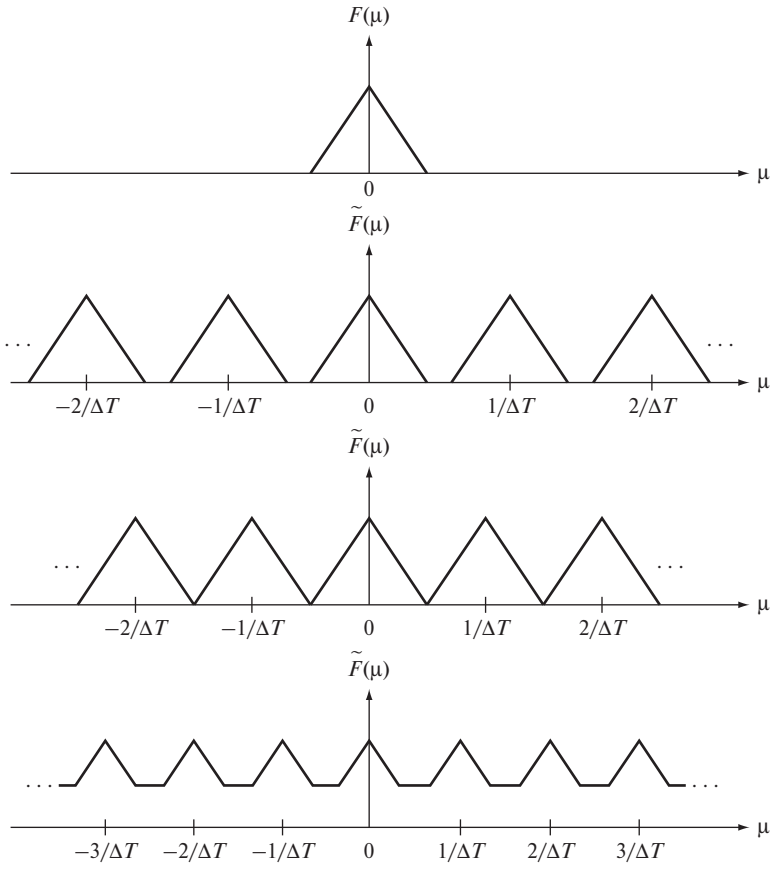


Рис. 4.6. (*а*) Фурье-преобразование непрерывной функции с ограниченной полосой частот. (*б*)—(*г*) Преобразования соответствующих дискретизованных функций при условии *избыточной дискретизации*, *точной дискретизации* и *субдискретизации* соответственно

где последний шаг следует из отсеивающего свойства импульса, как это показано равенством (4.2-10).

Суммирование в последнем шаге выражения (4.3-5) показывает, что Фурье-образ $\tilde{F}(\mu)$ дискретизованной функции $\tilde{f}(t)$ представляет собой *бесконечную периодическую* последовательностью *копий* функции $F(\mu)$, которая является Фурье-образом исходной, непрерывной функции $f(t)$. Интервал между копиями задается значением $1/\Delta T$. Заметим, что хотя $\tilde{f}(t)$ есть дискретизованная функция, ее Фурье-образ $\tilde{F}(\mu)$ будет *непрерывным*, поскольку состоит из копий $F(\mu)$, которая является непрерывной функцией.

Графическое объединение полученных результатов[5] показано на рис. 4.6. Рис. 4.6(*а*) представляет собой эскиз $F(\mu)$, Фурье-образа функции $f(t)$, а на рис. 4.6(*б*) показана $\tilde{F}(\mu)$ — Фурье-образ дискретизованной функции. Как упо-

[5] Для ясности иллюстраций схемы преобразований Фурье на рис. 4.6, а также другие аналогичные рисунки в настоящей главе игнорируют тот факт, что Фурье-образ, как правило, является комплексной функцией.

миналось в предыдущем разделе, значение $1/\Delta T$ является частотой дискретизации, используемой для получения дискретизованной функции. Как видно, на рис. 4.6(*б*) частота дискретизации выбрана достаточно высокой, что позволяет обеспечить уверенное разделение периодов и тем самым сохранить правильность $F(\mu)$. На рис. 4.6(*в*) частота дискретизации является минимально возможной для сохранения $F(\mu)$, но на рис. 4.6(*г*) частота дискретизации выбрана ниже минимума, необходимого для сохранения точных копий $F(\mu)$, и тем самым не позволяет сохранить оригинальный Фурье-образ. Рис. 4.6(*б*) — результат так называемой *передискретизации* или *избыточной дискретизации*, а рис. 4.6(*в*) и (*г*) — результаты *критической дискретизации* и *субдискретизации* сигнала соответственно. Эти понятия являются основой материала следующих разделов.

4.3.3. Теорема отсчетов

Идея дискретизации была наглядно представлена в разделе 2.4. Сейчас рассмотрим процесс дискретизации формально и установим условия, при которых непрерывная функция может быть *однозначно восстановлена* по множеству своих отсчетов.

Функция $f(t)$, Фурье-образ которой равняется нулю на частотах вне некоторого конечного интервала (полосы) $[-\mu_{max}, \mu_{max}]$ вокруг начала координат, называется *функцией с ограниченной полосой частот*. Рис. 4.7(*а*), который является увеличенной центральной частью рис. 4.6(*а*), представляет Фурье-образ такой функции. Аналогично на рис. 4.7(*б*) более детально показана центральная часть Фурье-образа функции с точной дискретизацией, показанного на рис. 4.6(*в*). Более низкие значения $1/\Delta T$ приведут к наложению точек $\tilde{F}(\mu)$, а более высокие значения — к появлению пустых промежутков между копиями $F(\mu)$.

а
б

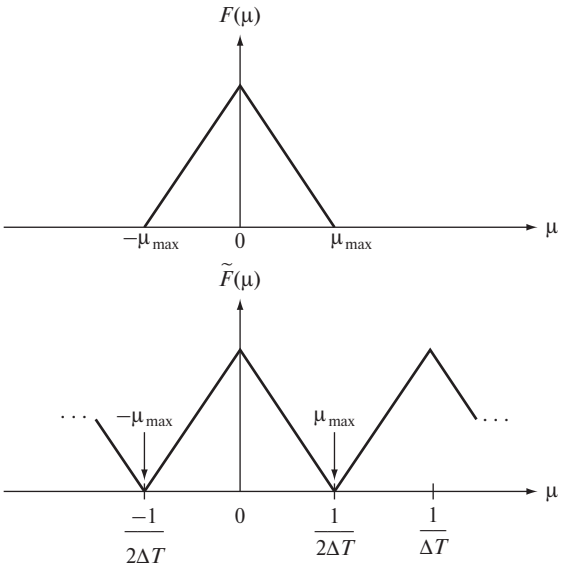


Рис. 4.7. (*а*) Фурье-образ непрерывной функции с ограниченной полосой частот. (*б*) Фурье-образ той же функции, дискретизованной с частотой точной дискретизации

Непрерывная функция $f(t)$ может быть восстановлена из своей дискретизованной версии, если можно выделить копию $F(\mu)$ из периодической последовательности копий этой функции, содержащейся в $\tilde{F}(\mu)$ — Фурье-образе дискретизованной функции $\tilde{f}(t)$. Напомним из обсуждения в предыдущем разделе, что $\tilde{F}(\mu)$ является *непрерывной периодической* функцией с периодом $1/\Delta T$. Таким образом, для описания полного преобразования достаточно знать один полный период. Из этого следует, что обратным преобразованием Фурье можно восстановить $f(t)$ из единственного периода.

Выделение из $\tilde{F}(\mu)$ единственного периода, эквивалентного $F(\mu)$, возможно в случае, если удается удовлетворительно провести границы между копиями (см. рис. 4.6). Согласно рис. 4.6(*б*), удовлетворительное разделение обеспечивается при $1/2\ \Delta T > 2\mu_{max}$ или

$$\frac{1}{\Delta T} > 2\mu_{max}. \tag{4.3-6}$$

Это неравенство показывает, что непрерывная функция с ограниченной полосой частот может быть точно восстановлена по множеству своих отсчетов, если эти отсчеты сделаны с частотой, превышающей удвоенную верхнюю (максимальную) частоту сигнала функции. Этот результат известен как *теорема отсчетов*[6]. Основываясь на данном результате, можно утверждать, что если непрерывная функция с ограниченной полосой частот представляется набором отсчетов, взятых с частотой, превышающей удвоенную верхнюю частоту сигнала функции, то потери информации не происходит[7]. И наоборот, можно утверждать, что максимальная частота, которая может быть «зафиксирована» при дискретизации сигнала с частотой дискретизации $1/\Delta T$, составляет $\mu_{max} = \Delta T/2$. Дискретизация с частотой Найквиста иногда является достаточной для точного восстановления сигнала, однако имеются случаи, когда это приводит к трудностям, что иллюстрируется ниже в примере 4.3. Таким образом, теорема отсчетов определяет, что дискретизация должна превышать частоту Найквиста.

Чтобы увидеть, как в принципе возможно восстановление $F(\mu)$ из $\tilde{F}(\mu)$, рассмотрим рис. 4.8, показывающий преобразование Фурье функции, дискретизованной с частотой немного выше частоты Найквиста. Функция на рис. 4.8(*б*) задана формулой:

$$H(\mu) = \begin{cases} \Delta T & -\mu_{max} \le \mu \le \mu_{max}, \\ 0 & \text{в остальных случаях.} \end{cases} \tag{4.3-7}$$

---

[6] Теорема отсчетов является краеугольным камнем теории цифровой обработки сигналов. Она была впервые сформулирована в 1928 г. Гарри Найквистом, ученым и инженером Bell Laboratories. Формально теорему доказал в 1949 г. Клод Шеннон, также из Bell Laboratories. Возобновление интереса к теореме отсчетов в конце 1940-х годов было вызвано появлением первых цифровых вычислительных систем и современных систем связи, что вызвало необходимость появления методов, оперирующих цифровыми данными (отсчетами). — *Прим. авт.*

В русскоязычной литературе теорема отсчетов хорошо известна как *теорема Котельникова*. Теорема была предложена и доказана В.А. Котельниковым в 1933 году в работе «О пропускной способности эфира и проволоки в электросвязи». — *Прим. перев.*

[7] Частота отсчетов, равная удвоенной величине максимальной частоты сигнала, называется частотой Найквиста.

а
б
в

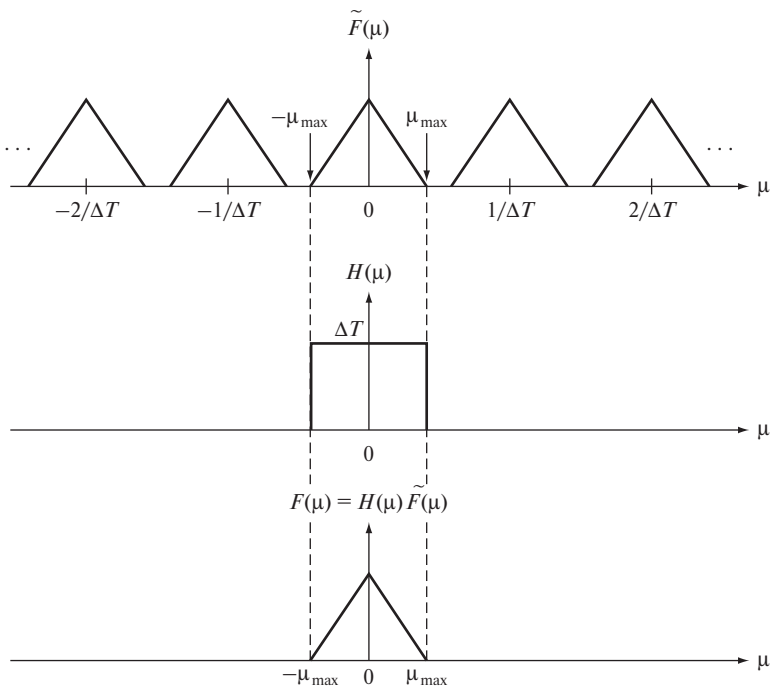


Рис. 4.8. Выделение одного периода Фурье-образа функции с ограниченной полосой частот при помощи идеального низкочастотного фильтра

При ΔT в (4.3-7) происходит сокращение $1/\Delta T$ в выражении (4.3-5).

При умножении на периодическую последовательность на рис. 4.8(*а*) эта функция выделяет период с центром в начале координат. Т. е., как показано на рис. 4.8(*в*), при умножении $\tilde{F}(\mu)$ на $H(\mu)$ в результате мы получаем $F(\mu)$:

$$F(\mu) = H(\mu)\tilde{F}(\mu). \tag{4.3-8}$$

Получив $F(\mu)$, можно восстановить $f(t)$ при помощи обратного преобразования Фурье:

$$f(t) = \int_{-\infty}^{\infty} F(\mu)e^{i2\pi\mu t}d\mu. \tag{4.3-9}$$

Формулы (4.3-7)—(4.3-9) доказывают, что теоретически возможно точно восстановить функцию с ограниченной полосой частот из отсчетов, взятых с частотой дискретизации, превышающей удвоенную верхнюю частоту функции. Как будет обсуждаться в следующем разделе, требование, чтобы $f(t)$ являлась функцией с ограниченной полосой частот, вообще говоря, означает, что $f(t)$ должна продолжаться от $-\infty$ до ∞ — условие, которое невозможно выполнить на практике. Как скоро будет показано, ограничение длительности функции препятствует точному восстановлению функции, за исключением некоторых особых случаев.

Функция $H(\mu)$ называется *низкочастотным фильтром*, поскольку она пропускает частоты нижней части диапазона частот и задерживает (отфильтровывает) все высокие частоты. Она также называется *идеальным* низкочастотным фильтром, поскольку содержит бесконечно крутые переходы по амплитуде (от 0 до ΔT в точке $-\mu_{max}$ и обратно в точке μ_{max}), — характеристика, недостижимая физическими электронными компонентами. Идеальный фильтр можно создать программно, но даже в этом случае возникают ограничения, как это объясняется в разделе 4.7.2. Позже в настоящей главе можно будет сказать о фильтрации значительно больше. Фильтры, используемые для только что рассмотренных целей восстановления (реконструкции) исходной функции по своим отсчетам, называются *фильтрами реконструкции*, и их можно считать базовым инструментарием для этих задач.

4.3.4. Наложение спектров

В этот момент логично задаться вопросом: «Что произойдет, если функция с ограниченной полосой частот дискретизована с частотой, которая ниже удвоенной верхней частоты сигнала?» Это соответствует случаю субдискретизации, рассмотренному в предыдущем разделе. Иллюстрирующий эту ситуацию рис. 4.9(*а*) повторяет рис. 4.6(*г*). В случае снижения частоты дискретизации ниже частоты Найквиста эффект состоит в том, что периоды перекрываются, и выделить единственный период становится невозможным независимо от используемого фильтра. Например, применение даже идеального низкочастотного фильтра на рис. 4.9(*б*) приведет к выделению Фурье-образа, который будет искажен на частотах, оказавшихся занятыми соседними периодами, как это видно на рис. 4.9(*в*). Обратное преобразование затем приведет к восстановлению искаженной функции от *t*. Этот эффект, вызываемый субдискретизацией финкции, известен как *наложение диапазонов частот* (*спектров*) или просто *наложение*[8] (*aliasing*). Образно говоря, наложение спектров есть процесс, в котором высокочастотные компоненты непрерывной функции в дискретизованной функции «притворяются» низкочастотными. Это согласуется с общим употреблением английского слова *alias* в значении «вымышленная личность, псевдоним».

К сожалению, за исключением некоторых особых случаев, о которых речь ниже, наложение спектров присутствует в дискретизованных сигналах всегда,

---

[8] Термин *наложение* или *совмещение* (aliasing), чаще употребляемый в русскоязычных текстах в виде транслитерации *элайсинг* (с некоторыми вариациями написания), объединяет достаточно большой диапазон искажений изображений, на первый взгляд, не связанных друг с другом, таких как периодические структуры и муар, ступенчатость гладких линий, «эффект колеса» — нарушение видимой скорости вращения объектов в видеопоследовательностях (вплоть до обратного) и другие. Несмотря на явные различия в проявлении, корень у всех этих искажений один — неправильная дискретизация исходного сигнала (или передискретизация при воспроизведении), когда нарушается условие теоремы Котельникова и в спектре сигнала (пространственном или временно́м) остаются частоты выше частоты Найквиста, что в результате приводит к *наложению спектров* (отсюда и название). Более подробно этот вопрос рассматривается в разделе 4.5.4. — *Прим. перев.*

а
б
в

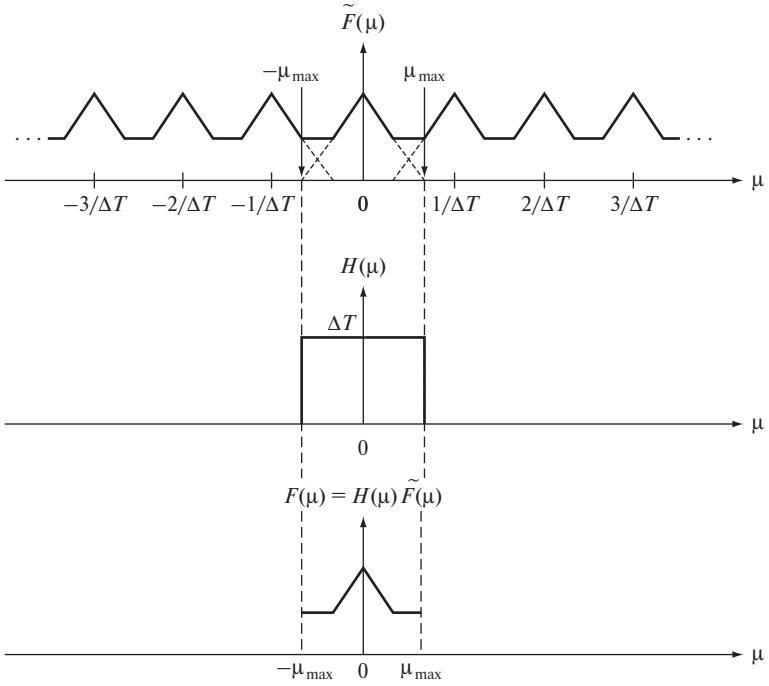


Рис. 4.9. (*а*) Преобразование Фурье субдискретизованной функции с ограниченной полосой частот. (Наложение спектров соседних периодов показано пунктиром.) (*б*) Идеальный низкочастотный фильтр; тот же самый, что и на рис. 4.8(*б*). (*в*) Результат произведения (*а*) и (*б*). Взаимовлияние соседних периодов приводит к наложению спектров, что препятствует точному восстановлению как $F(\mu)$, так и исходной непрерывной функции с ограниченной полосой частот. Сравните с рис. 4.8

поскольку даже если исходная дискретизуемая функция имела ограниченную полосу частот, в момент ограничения длительности функции, что мы всегда вынуждены делать на практике, возникают компоненты бесконечной частоты. Предположим, например, что мы хотим ограничить длительность функции $f(t)$ с ограниченной полосой частот интервалом, скажем, $[0, T]$. Это можно сделать умножением $f(t)$ на функцию

$$h(t) = \begin{cases} 1 & 0 \le t \le T, \\ 0 & \text{в остальных случаях.} \end{cases} \qquad (4.3\text{-}10)$$

Данная функция имеет ту же базовую форму, что и функция на рис. 4.4(*а*), чей Фурье-образ $H(\mu)$ имеет частотные компоненты, продолжающиеся в бесконечность, как показано на рис. 4.4(*б*). Из теоремы о свертке известно, что Фурье-образ произведения $h(t)f(t)$ равняется свертке Фурье-образов функций. Даже если Фурье-образ $f(t)$ ограничен по частоте, результат его свертки с $H(\mu)$ будет иметь частотные компоненты, продолжающиеся в бесконечность. Таким образом, ни одна из ограниченных по времени функций не может являться функ-

цией с ограниченной полосой частот. И наоборот, функция с ограниченной полосой частот обязана простираться от $-\infty$ до ∞.[9]

Исходя из вышеуказанных причин, можно сделать заключение, что наложение спектров является неизбежным фактом при работе с дискретизованными записями конечной длины. На практике эффект наложения спектров может быть *уменьшен* сглаживанием исходной функции путем ослабления самых верхних частот сигнала (например, в случае изображения, при помощи расфокусировки). Процесс *сглаживания* (*anti-aliasing*) должен выполняться *до* дискретизации функции, поскольку наложение спектров, являющееся результатом дискретизации, не может быть «отменено» при помощи вычислительных методов.

Пример 4.3. Наложение спектров.

■ Классическая иллюстрация эффекта наложения спектров представлена на рис. 4.10. Чисто синусоидальная функция, продолжающаяся бесконечно в обоих направлениях, имеет единственную частоту, т. е. имеет ограниченную полосу частот. Пусть функция на рисунке (в данный момент отвлечемся от больших точек) описывается закономерностью $\sin(\pi t)$, а горизонтальная ось соответствует времени t в секундах. Функция пересекает эту ось в точках $t = \ldots -1, 0, 1, 2, 3, \ldots$.

Период P функции $\sin(\pi t)$ составляет 2 с., а частота ее равна $1/P$, т. е. 1/2 цикла/с. В соответствии с теоремой отсчетов, данный сигнал может быть восстановлен по набору отсчетов, если их частота дискретизации $1/\Delta T$ вдвое превышает верхнюю частоту сигнала. Это означает, что для восстановления сигнала требуется частота дискретизации выше, чем 1 отсчет/с. ($1/2 \times 2 = 1$) или $\Delta T < 1$ с. Заметим, что дискретизация выбранного сигнала с частотой, *точно* равной его удвоенной (т. е. 1 отсчет/с.), и отсчетами, взятыми в точках $t = \ldots -1, 0, 1, 2, 3, \ldots$, даст в результате значения $\ldots \sin(-\pi), \sin(0), \sin(\pi), \sin(2\pi), \ldots$, которые все равны 0. Это является иллюстрацией причины, по которой теорема отсчетов требует, чтобы частота дискретизации обязательно превышала удвоенную верхнюю частоту сигнала, как и было указано выше.

Напомним, что частота 1 цикл/с есть по определению 1 Гц.

Большие точки на рис. 4.10 суть отсчеты, взятые равномерно со скоростью, меньшей, чем 1 отсчет/с (фактически расстояния между отсчетами превышают 2 с, что дает частоту дискретизации ниже 1/2 отсчета/с). Оцифрованный сигнал выглядит как синусоидальный, однако его частота — около одной десятой частоты оригинала. Этот дискретизованный сигнал, частота которого значительно ниже частоты исходной непрерывной функции, также является результатом наложения спектров. Имея в качестве результата фильтрации отсчеты, показанные черными точками на рис. 4.10 (являющиеся серьезнейшими погрешностя-

[9] Важный особый случай имеет место, когда функция, простирающаяся от $-\infty$ до ∞, имеет ограниченную полосу частот и является периодической. В такой ситуации функция может быть усечена по времени и все еще оставаться ограниченной по полосе частот, *при условии*, что остается *целое число* периодов. Единственный оставленный период (а значит, и вся функция) может быть представлен набором дискретных отсчетов, взятых на усеченном интервале и удовлетворяющих теореме отсчетов.

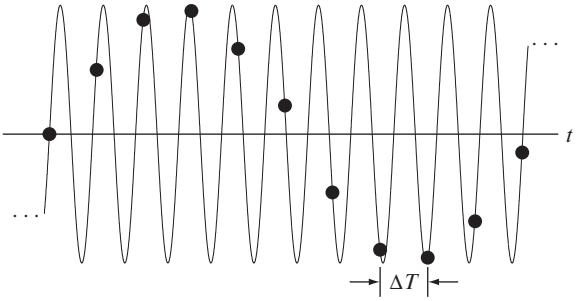


Рис. 4.10. Иллюстрация эффекта наложения спектров. Субдискретизованная функция (черные точки) выглядит как синус, имеющий частоту много ниже, чем частота исходного непрерывного синусоидального сигнала с периодом 2 с, пересечения которого с горизонтальной осью происходят каждую секунду. ΔT — период между отсчетами

ми, вызванными наложением спектров), невозможно узнать, что полученные отсчеты не дают правильного представления об исходной функции. Как будет видно ниже в данной главе, в случае изображений наложение спектров может привести к аналогичным обманчивым результатам. ■

4.3.5. Реконструкция (восстановление) функции из отсчетов

Реконструкция функции по набору ее отсчетов сводится к интерполяции между отсчетами. Даже простой вывод изображения на дисплей требует восстановления его из отсчетов устройством воспроизведения. Таким образом, понимание основ восстановления дискретизованных данных весьма важно. Центральной для понимания этого вопроса является свертка, что еще раз показывает актуальность этой концепции.

Основные принципы процедуры точного восстановления функции с ограниченной полосой частот из отсчетов при помощи частотных методов были изложены при обсуждении рис. 4.8 и выражения (4.3-8). Используя теорему о свертке, можно получить эквивалентный результат в пространственной области. Согласно (4.3-8), $F(\mu) = H(\mu)\tilde{F}(\mu)$ и, следовательно,

$$f(t) = \Im^{-1}\{F(\mu)\} = \Im^{-1}\{H(\mu)\tilde{F}(\mu)\} = h(t) \star \tilde{f}(t), \tag{4.3-11}$$

где последний шаг следует из соотношения (4.2-21) теоремы о свертке. Можно показать (задача 4.6), что подстановка (4.3-1) для $\tilde{f}(t)$ в выражение (4.3-11) и использование затем уравнения (4.2-20) приводит к следующему выражению для $f(t)$ в *пространственной области*:

$$f(t) = \sum_{n=-\infty}^{\infty} f(n\Delta T)\,\mathrm{sinc}[(t - n\Delta T)/\Delta T], \tag{4.3-12}$$

где функция sinc определена в (4.2-19). Данный результат не является неожиданным, поскольку обратное преобразование Фурье от прямоугольного фильтра $H(\mu)$ также является sinc-функцией (см. пример 4.1). Выражение (4.3-12) показывает, что точно восстанавливаемая функция является бесконечной суммой

sinc-функций, взвешенных значениями отсчетов, и ее важным свойством является то, что значения такой функции точно совпадают со значениями отсчетов, расположенных в целочисленном ряду периодов ΔT. То есть для любого $t = k\Delta T$, где k — целое, $f(t)$ равна k-му отсчету $f(k\Delta T)$. Это следует из выражения (4.3-12), поскольку $\mathrm{sinc}(0) = 1$ и $\mathrm{sinc}(m) = 0$ для любого другого значения m. Значения $f(t)$ между точками отсчетов даются интерполяциями, формирующимися суммами sinc-функций.

Для интерполяции между отсчетами выражение (4.3-12) требует бесконечного числа членов. На практике это означает, что необходимо искать приближения в виде конечных интерполяций между отсчетами. Как обсуждалось в разделе 2.4.4, основными методами интерполяции, используемыми в обработке изображений, являются выбор ближайшего соседа, билинейная и бикубическая интерполяции. Эффекты интерполяции будут обсуждаться в разделе 4.5.4.

4.4. Дискретное преобразование Фурье (ДПФ) одной переменной

Одной из основных целей данной главы является вывод *дискретного преобразования Фурье* (ДПФ), начиная с основных принципов. Сведения, изложенные к данному моменту, можно рассматривать как фундамент этих основных принципов, так что теперь у нас есть все необходимое для вывода ДПФ.

4.4.1. Получение ДПФ из непрерывного преобразования дискретизованных функций

Согласно изложенному в разделе 4.3.2, преобразование Фурье дискретизованной функции, определенной от $-\infty$ до ∞ и имеющей ограниченную полосу частот, является непрерывной периодической функцией, которая также задана от $-\infty$ до ∞. На практике приходится иметь дело с конечным числом отсчетов, поэтому целью настоящего раздела является вывод ДПФ, соответствующего такому набору значений.

Выражение (4.3-5) дает Фурье-преобразование $\tilde{F}(\mu)$ дискретизованной функции $\tilde{f}(t)$ через преобразование исходной функции $F(\mu)$, но не через значения самой дискретизованной функции $\tilde{f}(t)$. Выведем это выражение напрямую из определения Фурье-преобразования, определяемого (4.2-16):

$$\tilde{F}(\mu) = \int_{-\infty}^{\infty} \tilde{f}(t) e^{-i2\pi\mu t} dt \,. \tag{4.4-1}$$

Подставляя $\tilde{f}(t)$ из (4.3-1), получим

$$\begin{aligned} \tilde{F}(\mu) &= \int_{-\infty}^{\infty} \tilde{f}(t) e^{-i2\pi\mu t} dt = \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f(t)\delta(t - n\Delta T) e^{-i2\pi\mu t} dt = \\ &= \sum_{n=-\infty}^{\infty} \int_{-\infty}^{\infty} f(t)\delta(t - n\Delta T) e^{-i2\pi\mu t} dt = \sum_{n=-\infty}^{\infty} f_n e^{-i2\pi\mu n\Delta T}, \end{aligned} \tag{4.4-2}$$

где последний шаг следует из (4.3-2). Хотя f_n является дискретной функцией, ее Фурье-преобразование $\tilde{F}(\mu)$ — непрерывная периодическая функция с периодом $1/\Delta T$, как видно из (4.3-5). Таким образом, для описания всей функции $\tilde{F}(\mu)$ требуется лишь один период, и дискретизация одного периода является основой для ДПФ.

Предположим, нам требуется получить M равноотстоящих отсчетов $\tilde{F}(\mu)$, взятых на отрезке от $\mu = 0$ до $\mu = 1/\Delta T$. Для этого берутся отсчеты на следующих частотах:

$$\mu = \frac{m}{M\Delta T} \quad m = 0, 1, 2, \ldots, M-1. \tag{4.4-3}$$

Подставляя этот результат для μ в выражение (4.4-2) и обозначая результат через F_m, получим:

$$F_m = \sum_{n=0}^{M-1} f_n e^{-i2\pi mn/M} \quad m = 0, 1, 2, \ldots, M-1. \tag{4.4-4}$$

Это выражение и является искомым дискретным преобразованием Фурье[10]. Для заданного набора $\{f_n\}$, состоящего из M отсчетов, выражение (4.4-4) позволяет получить набор $\{F_m\}$ также из M комплексных дискретных величин, соответствующих дискретному преобразованию Фурье исходного набора отсчетов. И наоборот, используя *обратное дискретное преобразование Фурье* (обратное ДПФ), из заданного набора $\{F_m\}$ можно восстановить набор отсчетов $\{f_n\}$:

$$f_n = \frac{1}{M} \sum_{m=0}^{M-1} F_m e^{i2\pi mn/M} \quad n = 0, 1, 2, \ldots, M-1. \tag{4.4-5}$$

Нетрудно показать (задача 4.8), что подставляя (4.4-5) вместо f_n в (4.4-4), получим тождественный результат: $F_m \equiv F_m$. И наоборот, подставляя (4.4-4) в (4.4-5), получим аналогичное тождество $f_n \equiv f_n$. Это означает, что выражения (4.4-4) и (4.4-5) составляют *пару дискретных преобразований Фурье* (ДПФ-пару). Более того, эти тождества показывают, что прямое и обратное преобразования Фурье существуют для любых наборов данных с конечными значениями. Заметим, что ни одно из выражений явно не зависит ни от шага дискретизации ΔT, ни от частотного интервала в выражении (4.4-3). Таким образом, ДПФ-пара применима к *любому* равномерно взятому набору отсчетов.

Мы использовали символы m и n в предыдущих операциях для обозначения дискретных переменных, поскольку эти обозначения обычны для выкладок. Однако более наглядно, особенно при работе с двумя переменными, использовать символы x и y для обозначения координат на изображении, а для частотных

---

[10] Согласно рис. 4.6(*б*), отрезок [0, $1/\Delta T$] покрывает два *полупериода* соседних периодов преобразования, и высокие частоты в нем расположены посередине. Это означает, что данные в наборе F_m требуют переупорядочения отсчетов от нижней частоты к верхней. Это цена, которую приходится платить за удобство обозначений, беря отсчеты в точках $m = 0, 1, 2, \ldots, M-1$ вместо использования отсчетов по обе стороны от начала координат, что потребовало бы отрицательной индексации. Процедура исправления порядка данных рассмотрена в разделе 4.6.3.

переменных — u и v; все эти переменные предполагаются *целыми*[11]. Тогда выражения (4.4-4) и (4.4-5) примут вид:

$$F(u)=\sum_{x=0}^{M-1} f(x)e^{-i2\pi mx/M} \quad u=0,1,2,...,M-1 \tag{4.4-6}$$

и

$$f(x)=\frac{1}{M}\sum_{u=0}^{M-1} F(u)e^{i2\pi ux/M} \quad x=0,1,2,...,M-1, \tag{4.4-7}$$

где мы для простоты вместо индексов использовали обозначения функций. Совершенно ясно, что $F(u) \equiv F_m$ и $f(x) \equiv f_n$. Начиная с этого места, мы будем использовать выражения (4.4-6) и (4.4-7) для обозначения одномерной ДПФ-пары. Иногда множитель $1/M$ вставляется в выражение (4.4-6) тогда, когда мы используем его в обратном преобразовании (4.4-7). Это не влияет на доказательство того, что данные два выражения составляют пару преобразований Фурье.

Можно показать (задача 4.9), что как прямое, так и обратное ДПФ являются периодическими с периодом M. То есть

$$F(u)=F(u+kM) \tag{4.4-8}$$

и

$$f(x)=f(x+kM), \tag{4.4-9}$$

где k является целым.

> Неочевидным является факт, почему дискретная функция $f(x)$ должна быть периодической, учитывая, что непрерывная функция, отсчеты которой она представляет, может таковой не являться. Одним нестрогим обоснованием является учет того, что сама дискретизация приводит к периодическому ДПФ. Поэтому для существования ДПФ-пары логично, чтобы $f(x)$, которая является обратным ДПФ, также была периодической.

Дискретным эквивалентом уравнения свертки (4.2-20) будет

$$f(x)\star h(x)=\sum_{m=0}^{M-1} f(m)h(x-m) \tag{4.4-10}$$

для $x = 0, 1, 2, ..., M - 1$. Поскольку, согласно предыдущим формулировкам, функции являются периодическими, их свертка также должна быть периодической. Уравнение (4.4-10) задает один период такой периодической свертки. По этой причине внутренний процесс, который описывается данным уравнением и является прямым следствием периодичности прямого и обратного ДПФ, часто называют *круговой сверткой*. Это является заметным отличием

---

[11] Мы из осторожности использовали t для *непрерывной* пространственной переменной и μ для соответствующей *непрерывной* частотной переменной. Начиная с этого места, мы будем использовать x и u для обозначения одномерных *дискретных* пространственной и частотной переменных соответственно. При работе с двумерными функциями мы будем использовать (t, z) и (μ, v) для обозначения *непрерывных* переменных пространственной и частотной областей соответственно. Аналогично их *дискретные* эквиваленты будут обозначаться (x, y) и (u, v).

от свертки, рассматриваемой в разделе 3.4.2, где значения смещения x полностью определялись условиями полного смещения одной функции по другой и не были ограничены диапазоном $[0, M-1]$, как в случае круговой свертки. Это различие и его значимость будут рассматриваться в разделе 4.6.3 и при обсуждении рис. 4.28.

Наконец, укажем, что теорема о свертке, задаваемая соотношениями (4.2-21) и (4.2-22), также применима и к дискретным переменным (задача 4.10).

4.4.2. Взаимосвязь между шагом дискретизации и частотными интервалами

Если $f(x)$ состоит из M отсчетов функции $f(t)$, взятых с шагом ΔT, общая длительность записи, содержащей набор $\{f(x)\}$, $x = 0, 2, ..., M-1$, будет

$$T = M\Delta T . \tag{4.4-11}$$

Соответствующий дискретный интервал Δu в частотной области определяется выражением (4.4-3)

$$\Delta u = \frac{1}{M\Delta T} = \frac{1}{T} . \tag{4.4-12}$$

Весь частотный диапазон, покрываемый M компонентами ДПФ, будет

$$\Omega = M\Delta u = \frac{1}{\Delta T} . \tag{4.4-13}$$

Таким образом, из выражений (4.4-12) и (4.4-13) видно, что частотное разрешение ДПФ, равное Δu, зависит от длительности T, в течение которого производилась дискретизация непрерывной функции $f(t)$, а диапазон частот, охваченный ДПФ, зависит от шага дискретизации ΔT. Заметим, что оба выражения демонстрируют *обратную* зависимость по отношению к T и ΔT.

Пример 4.4. Техника вычисления ДПФ.

■ На рис. 4.11(*а*) показаны четыре отсчета непрерывной функции $f(t)$, взятые с шагом ΔT. На рис. 4.11(*б*) — те же четыре значения в дискретном пространстве

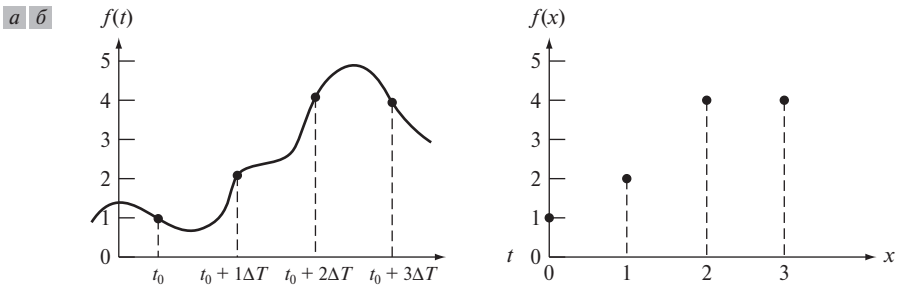


Рис. 4.11. (*а*) Непрерывная функция и (*б*) значения ее отсчетов в дискретном пространстве x. На (*а*) t является непрерывной переменной; на (*б*) x принимает целые значения

x. Заметим, что значения x составляют 0, 1, 2 и 3, показывая тем самым, что мы можем ссылаться на любую четверку отсчетов $f(t)$.

Из выражения (4.4-6) найдем значения $F(u)$, $u = 0, 1, 2, 3$:

$$F(0)=\sum_{x=0}^{3} f(x)=\left[f(0)+f(1)+f(2)+f(3)\right]=1+2+4+4=11,$$

$$F(1)=\sum_{x=0}^{3} f(x)e^{-i2\pi(1)x/4}=1e^{0}+2e^{-i\pi/2}+4e^{-i\pi}+4e^{-i(3/2)\pi}=-3+2i.$$

Аналогично $F(2) = -(1 + 0i)$ и $F(3) = -(3 + 2i)$. Заметим, что для вычисления *каждого* из членов $F(u)$ используются *все* значения $f(x)$.

Наоборот, если имеются значения $F(u)$ и необходимо вычислить значения $f(x)$, то их можно получить аналогичным способом, используя обратное преобразование. Например,

$$f(0)=\frac{1}{4}\sum_{u=0}^{3} F(u)e^{i2\pi u\cdot 0}=\frac{1}{4}\sum_{u=0}^{3} F(u)=\frac{1}{4}\left(11-3+2i-1-3-2i\right)=\frac{1}{4}(4)=1,$$

что совпадает с рис. 4.11(*б*). Остальные значения получаются аналогичным образом. ■

4.5. Расширение на функции двух переменных

В данном разделе мы расширяем концепции, введенные в разделах 4.2—4.4, на множества двух переменных.

4.5.1. Двумерный импульс и его свойство отсеивания

Импульс $\delta(t, z)$ двух непрерывных переменных t и z определяется так же, как и импульс одной переменной (4.2-8):

$$\delta(t)=\begin{cases}\infty & \text{если } t=z=0\\ 0 & \text{в остальных случаях}\end{cases} \tag{4.5-1а}$$

и

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\delta(t,z)\,dt\,dz=1. \tag{4.5-1б}$$

Как и в одномерном случае, двумерный импульс при интегрировании обладает *свойством отсеивания*:

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(t,z)\delta(t,z)\,dt\,dz=f(0,0), \tag{4.5-2}$$

или более общий случай для импульса в точке (t_0, z_0):

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(t,z)\delta(t-t_0,z-z_0)\,dt\,dz=f(t_0,z_0). \tag{4.5-3}$$

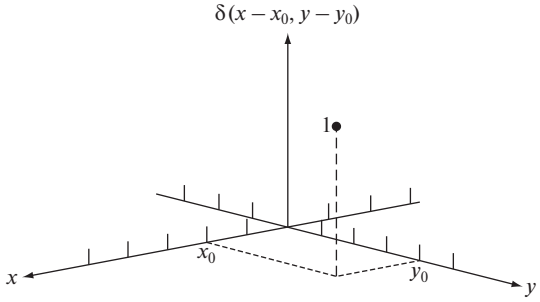


Рис. 4.12. Двумерный единичный дискретный импульс. Переменные x и y являются дискретными, а δ равна нулю везде, кроме точки (x_0, y_0)

Видно, что как и ранее, свойство отсеивания дает на выходе значение функции $f(t, z)$ в точке импульса.

Для дискретных переменных x и y двумерный импульс определяется как

$$\delta(x,y)=\begin{cases}1 & \text{если } x=y=0,\\ 0 & \text{в остальных случаях,}\end{cases} \tag{4.5-4}$$

а его свойство отсеивания — как

$$\sum_{x=-\infty}^{\infty}\sum_{y=-\infty}^{\infty} f(x,y)\delta(x,y)=f(0,0), \tag{4.5-5}$$

где $f(x,y)$ — функция двух дискретных переменных x и y. Для импульса в точке (x_0, y_0) (см. рис. 4.12) свойство отсеивания запишется как

$$\sum_{x=-\infty}^{\infty}\sum_{y=-\infty}^{\infty} f(x,y)\delta(x-x_0,y-y_0)=f(x_0,y_0). \tag{4.5-6}$$

Как и ранее, свойство отсеивания дает на выходе значение дискретной функции $f(x,y)$ в точке импульса.

4.5.2. Пара двумерных непрерывных преобразований Фурье

Пусть $f(t, z)$ — непрерывная функция двух переменных t и z. Пара двумерных непрерывных преобразований Фурье задается следующими выражениями:

$$F(\mu,v)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(t,z)e^{-i2\pi(\mu t+vz)}dt\,dz \tag{4.5-7}$$

и

$$f(t,z)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(\mu,v)e^{i2\pi(\mu t+vz)}d\mu\,dz, \tag{4.5-8}$$

где μ и v — частотные переменные. По отношению к изображениям t и z интерпретируются как непрерывные *пространственные* переменные. Как и в одномерном случае, область изменения переменных μ и v — *непрерывная частотная область*.

Пример 4.5. Вывод двумерного Фурье-преобразования прямоугольной функции.

■ На рис. 4.13 представлена двумерная прямоугольная функция, аналогичная одномерной в примере 4.1. Выполнение процедуры, подобной использованной в том примере, дает в результате

$$F(\mu,v)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(t,z)e^{-i2\pi(\mu t+vz)}dt\,dz=\int_{-T/2}^{T/2}\int_{-Z/2}^{Z/2} Ae^{-i2\pi(\mu t+vz)}dt\,dz=ATZ\left[\frac{\sin(\pi\mu T)}{(\pi\mu T)}\right]\left[\frac{\sin(\pi vZ)}{(\pi vZ)}\right].$$

Амплитуда (спектр) определяется выражением

$$|F(\mu,v)|=ATZ\left|\frac{\sin(\pi\mu T)}{(\pi\mu T)}\right|\left|\frac{\sin(\pi vZ)}{(\pi vZ)}\right|.$$

На рис. 4.13(*б*) изображена центральная часть спектра. Как и в одномерном случае, положения нулей спектра обратно пропорциональны значениям T и Z. Так, чем больше значения T и Z, тем более «сжатым» будет выглядеть спектр, и наоборот. ■

4.5.3. Двумерная дискретизация и двумерная теорема отсчетов

Аналогично одномерному случаю двумерная дискретизация может быть описана с помощью следующей функции дискретизации (двумерной последовательности импульсов):

$$s_{\Delta T\Delta Z}(t,z)=\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty}\delta(t-m\Delta T,z-n\Delta Z), \qquad (4.5\text{-}9)$$

где ΔT и ΔZ — промежутки (шаги) между отсчетами по осям t и z непрерывной функции $f(t, z)$. Выражение (4.5-9) описывает набор периодически расположенных импульсов, продолжающихся в бесконечность по обеим осям (рис. 4.14). Как и в одномерном случае, иллюстрированном рис. 4.5, умножение $f(t, z)$ на $s_{\Delta T\Delta Z}(t, z)$ дает в результате дискретизованную функцию.

Говорят, что $f(t, z)$ является функцией *с ограниченной полосой частот*, если ее Фурье-образ равен 0 вне некоторого прямоугольника, заданного интервалами $[-\mu_{max}, \mu_{max}]$ и $[-v_{max}, v_{max}]$, т. е.

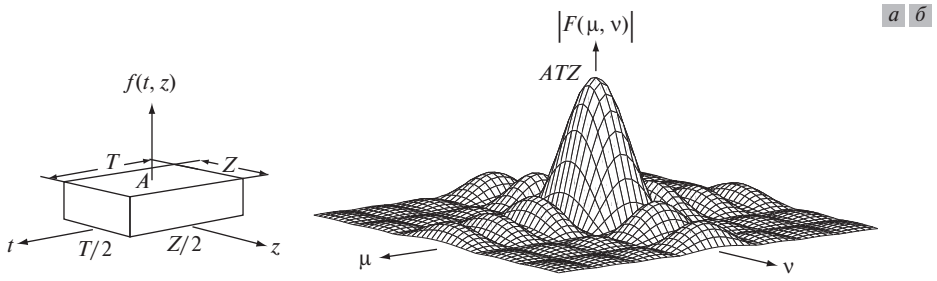


Рис. 4.13. (*а*) Двумерная функция и (*б*) центральная часть спектра (не в масштабе). Параллелепипед более вытянут вдоль оси t, поэтому его спектр более сжат вдоль оси μ. Сравните с рис. 4.4

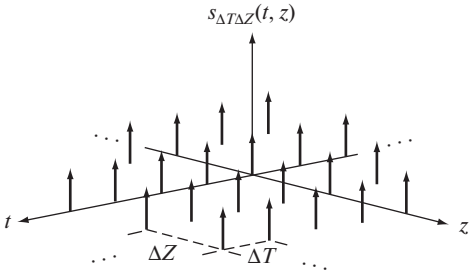


Рис. 4.14. Двумерная бесконечная последовательность импульсов

$$F(\mu, v) = 0 \quad \text{для } |\mu| \geq \mu_{\max} \text{ и } |v| \geq v_{\max}. \tag{4.5-10}$$

Двумерная теорема отсчетов утверждает, что непрерывная функция с ограниченной полосой частот $f(t, z)$ может быть точно восстановлена из набора своих отсчетов, если шаги дискретизации удовлетворяют следующим неравенствам:

$$\Delta T < \frac{1}{2\mu_{\max}} \tag{4.5-11}$$

и

$$\Delta Z < \frac{1}{2v_{\max}}, \tag{4.5-12}$$

или, выражая результат в терминах частоты дискретизации, если

$$\frac{1}{\Delta T} > 2\mu_{\max} \tag{4.5-13}$$

и

$$\frac{1}{\Delta Z} > 2v_{\max}. \tag{4.5-14}$$

Другими словами, если двумерная непрерывная функция с ограниченной полосой частот представляется набором своих отсчетов, сделанных с частотой, превышающей удвоенную верхнюю частоту сигнала функции по каждому из направлений μ и v, то потери информации не происходит.

а *б*

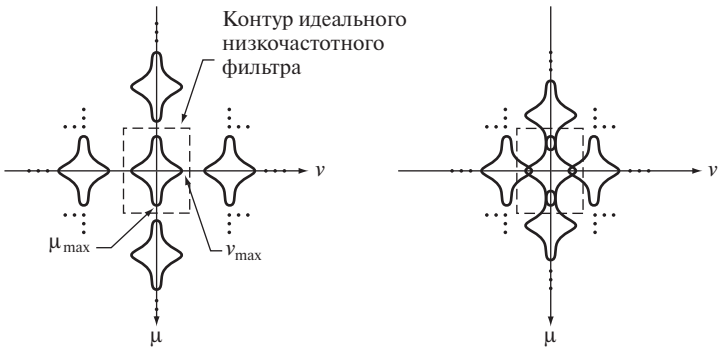


Рис. 4.15. Двумерное преобразование Фурье функции с ограниченной полосой частот: (*а*) передискретизация, (*б*) субдискретизация

На рис. 4.15 представлен двумерный эквивалент рис. 4.6(*б*) и (*г*). Идеальный двумерный прямоугольный фильтр имеет вид, показанный на рис. 4.13(*а*). Обведенный пунктиром прямоугольник на рис. 4.15(*а*) показывает границы фильтра, требуемые для выделения одного периода преобразования, чтобы восстановить функцию с ограниченной полосой частот по своим отсчетам (как и в разделе 4.3.3). Из раздела 4.3.4 известно, что если функция является субдискретизованной, то периоды ее Фурье-образа накладываются и выделение одного периода становится невозможным, что иллюстрируется на рис. 4.15(*б*). Результатом таких условий будет наложение спектров.

4.5.4. Наложение спектров при преобразовании изображений

В данном разделе мы расширим концепции наложения (спектров) применительно к задаче обработки изображений и рассмотрим некоторые аспекты, связанные с дискретизацией и передискретизацией (повторной дискретизацией).

Расширение по сравнению с одномерным наложением

Как и в одномерном случае, непрерывная функция $f(t, z)$ двух переменных t и z может иметь ограниченную полосу частот, вообще говоря, только в случае, если она продолжается бесконечно в обоих координатных направлениях. Само ограничение длительности функции вызывает искажение частотных компонент, продолжающееся в частотной области в бесконечность, как это объяснялось в разделе 4.3.4. Поскольку мы не можем дискретизовать функцию на бесконечности, наложение спектров всегда присутствует в цифровых изображениях, как и в дискретизованных одномерных функциях.

Имеются два принципиальных проявления наложения спектров в изображениях: пространственное и временно́е. *Пространственное наложение спектров*, как обсуждалось в разделе 4.3.4, возникает из-за субдискретизации. *Временно́е наложение спектров* связано с интервалом времени между кадрами изображений в видеопоследовательности. Одним из наиболее известных примеров временно́го наложения является эффект «колеса повозки», при котором колесо со спицами на видеопоследовательности (например в кино) выглядит как бы вращающимся назад. Это вызывается тем, что частота кадров слишком низка по сравнению со скоростью вращения колеса в видеопоследовательности.

В данной главе наше внимание будет уделено пространственному наложению. Основной интерес к пространственному наложению спектров на изображениях вызывается появлением таких артефактов, как рубчатость линейных деталей, ложные блики, муар, которые не имелись на исходном изображении. Артефакты, вызываемые наложением при работе с изображениями, иллюстрируются следующим примером.

Пример 4.6. Наложение спектров при работе с изображениями.

■ Вообразим систему регистрации изображений, являющуюся идеальной в том смысле, что она не добавляет шумов и возвращает изображение, точно повторяющее то, которое она «видит», однако допустимый размер изображения в данной системе фиксирован и составляет 96×96 пикселей. Если подать ей на вход изображение шахматной доски, то она сможет разрешить рисунки размерами до 96×96 клеток, когда каждая клетка занимает один пиксель; в этом случае

каждый пиксель результирующего изображения соответствует одному квадрату исходного рисунка. Интересно проверить, что произойдет, если детали рисунка (клетки шахматной доски) составят размер меньше одного пикселя, т. е. когда на вход системы подается изображение, в котором больше, чем 96×96 клеток.

> Данный пример не следует интерпретировать как нереальный. Дискретизация «идеальной» сцены в условиях отсутствия шумов и искажений является обычным случаем конвертирования компьютерных моделей и векторных рисунков в дискретное изображение.

На рис. 4.16(*а*) и (*б*) показаны результаты дискретизации шахматной доски, клетки которой отображаются квадратами размерами 16×16 и 6×6 пикселей соответственно. Данные результаты выглядят, как и можно было ожидать. Однако если размеры одной клетки оказываются слегка меньше одного пикселя, возникают сильные артефакты наложения, как это показано на рис. 4.16(*в*). Наконец, уменьшая размер клетки слегка меньше 0,5×0,5 пикселя, получим изображение, представленное на рис. 4.16(*г*). В этом случае результат с наложением выглядит как нормальный рисунок шахматной доски. Фактически результат оказывается таким же, как если дискретизовать шахматную доску с клетками размерами 12×12 пикселей. Последнее изображение является хорошим примером того, что наложение спектров может дать результат, который окажется весьма обманчивым. ■

Артефакты наложения могут быть *уменьшены* при помощи небольшой расфокусировки дискретизуемой сцены так, чтобы ослабить высокие частоты. Как объяснялось в разделе 4.3.4, сглаживающая фильтрация должна выполняться самой первой операцией, *до* дискретизации изображения. После дискретизации уже не существует способа или программного алгоритма, позволяюще-

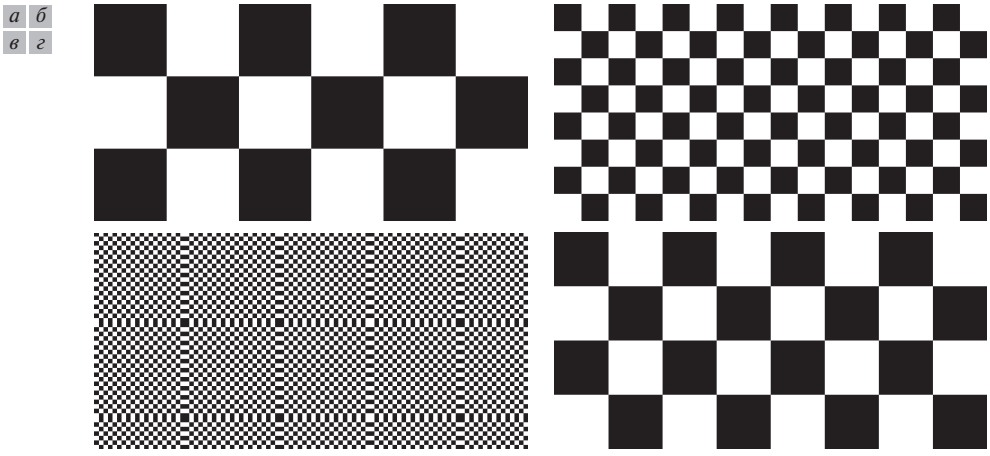


Рис. 4.16. Наложение спектров при работе с изображениями. На (*а*) и (*б*) размер стороны клетки 16 и 6 пикселей и артефакты наложения неощутимы. На (*в*) и (*г*) размеры клеток составляют 0,9174 и 0,4798 пикселя соответственно и на результате видны значительные артефакты наложения. Заметим, что (*г*) «имитирует» нормальное изображение

го уменьшить артефакты наложения, вызванные нарушением условия теоремы отсчетов. Большинство коммерческих пакетов и программ обработки изображений включают операцию, называемую «устранение артефактов наложения» (в оригинале «*anti-aliasing*» — *прим. перев.*). Однако, как показано в примерах 4.7 и 4.8, данный термин имеет отношение к сглаживанию *цифрового* изображения с целью уменьшения дополнительных искажений, вызываемых передискретизацией; он не имеет отношения к уменьшению артефактов наложения, уже имеющихся на исходном дискретизованном изображении. Значительное число промышленных цифровых камер имеют надлежащий сглаживающий фильтр, либо встроенный в объектив, либо находящийся на самой поверхности сенсоров. По этой причине затруднительно иллюстрировать артефакты наложения на изображениях, полученных при помощи таких камер.

Интерполяция и передискретизация изображения

Как и в одномерном случае, идеальное восстановление по отсчетам функции изображения с ограниченной полосой частот требует двумерной свертки (в пространственной области) с функцией sinc. Как пояснялось в разделе 4.3.5, такое теоретически идеальное восстановление требует интерполяции, в которой используется бесконечный ряд суммирования, что на практике заставляет прибегать к приближениям. Одним из наиболее общих приложений двумерной интерполяции является задача изменения размеров изображения (увеличения и уменьшения). Увеличение предполагает взятие более частых отсчетов, поэтому может рассматриваться как избыточная дискретизация, а уменьшение (более редкие отсчеты) — как субдискретизация. Основная разница между этими двумя операциями и концепцией дискретизации, обсуждавшейся в предыдущем разделе, заключается в том, что увеличение и уменьшение применяются по отношению к уже *дискретизованным* изображениям.

Интерполяция рассматривалась в разделе 2.4.4. Там наш интерес сводился к иллюстрации работы различных видов интерполяции — билинейной, бикубической и по ближайшему соседу. В настоящем разделе приводятся некоторые дополнительные примеры, в которых акцент сделан на результатах дискретизации и сглаживания. Особым случаем интерполяции по ближайшему соседу, который тесно связан с избыточной дискретизацией, является увеличение изображения путем *дублирования пикселей*, применимое в случае изменения размеров в целое число раз. Например, дублируя каждый столбец, мы вдвое увеличим размер изображения по горизонтали. Если затем продублировать каждую строку уже увеличенного по горизонтали изображения, его вертикальный размер возрастет вдвое. Та же самая процедура может быть использована для увеличения изображения в любое целое число раз. Значение яркости каждого пикселя определяется тем, что новый элемент точно дублирует значение старого.

Для сокращения размеров изображения в целое число раз используется симметричный увеличению способ. Прореживание осуществляется стиранием лишних строк и столбцов (т. е. чтобы вполовину сократить изображение, убираются все четные строки и столбцы). Для иллюстрации концепции сжатия с нецелым коэффициентом можно воспользоваться аналогией с увеличением сетки, рассмотренную в разделе 2.4.4, за исключением того, что теперь сначала сетка увеличивается до размеров, превышающих исходное изображение, затем происходит интерполяция уровней яркости, и наконец производится обратное

сжатие сетки до требуемого размера. Для уменьшения эффекта наложения перед сжатием рекомендуется осуществить небольшое размывание изображения (размывание в частотной области обсуждается в разделе 4.8). Альтернативный способ состоит в первоначальном осуществлении *сверхдискретизации* исходной сцены и последующем уменьшении ее размеров (передискретизации) путем стирания строк и столбцов. Такой способ позволяет достичь более резких результатов, чем со сглаживанием, но он требует доступа к исходной сцене. Очевидно, что если такой возможности не имеется (как обычно и бывает на практике), то сверхдискретизация использоваться не может.

Процесс передискретизации изображения без применения размывания, ограничивающего полосу частот, называется прореживанием.

Пример 4.7. Иллюстрация наложения при передискретизации изображения. ■ Артефакты наложения обычно усиливаются при уменьшении размера дискретизованного изображения. На рис. 4.17(*а*) показан снимок, специально сделанный для иллюстрации артефактов наложения (следует обратить внимание на близкорасположенные параллельные линии на всех элементах одежды модели). На исходном изображении незаметны никакие нежелательные артефакты, что говорит о том, что изначальная частота дискретизации была достаточной для предотвращения видимых артефактов наложения. Изображение на рис. 4.17(*б*) было получено двукратным уменьшением размера оригинала путем стирания строк и столбцов. На изображении стали весьма хорошо видны артефакты на-

а *б* *в*

Рис. 4.17. Иллюстрация артефактов наложения при передискретизации изображения. (*а*) Исходное цифровое изображение без видимых артефактов. (*б*) Результат двукратного уменьшения размеров путем стирания строк и столбцов. (*в*) Результат размывания изображения (*а*) усредняющим фильтром по окрестности 3×3 перед уменьшением размеров. Изображение размыто слегка больше, чем (*б*), но артефактов наложения больше не видно. (Исходное изображение предоставлено лабораторией сжатия сигналов Калифорнийского университета, Санта Барбара)

ложения (см., например, области коленей модели). Цифровым «эквивалентом» фильтра размывания, позволяющего избежать наложения спектров, является подавление высоких частот цифрового изображения путем сглаживания перед передискретизацией. На рис. 4.17(*в*) показан именно такой пример, когда изображение (*а*) перед уменьшением размера было сглажено усредняющим фильтром по окрестности 3×3 (см. раздел 3.5). Улучшение результата по сравнению с рис. 4.17(*б*) очевидно. Для упрощения сравнения изображения (*б*) и (*в*) были увеличены до исходных размеров путем дублирования строк и столбцов. ■

При работе с изображениями, имеющими резкие контуры, артефакты наложения проявляются в виде *ступенек* или *зубцов* (в оригинале «*jaggies*» — *прим. перев.*). Это явление иллюстрируется следующим примером.

Пример 4.8. Иллюстрация возникновения ступенчатости при уменьшении размеров изображения.

■ На рис. 4.18(*а*) представлено сгенерированное на компьютере изображение модельной сцены, имеющее размеры 1024×1024 элемента; артефакты наложения на этом изображении незаметны. Рис. 4.18(*б*) — результат уменьшения размеров (*а*) на 75%, т.е. до размеров 256×256 пикселей. Для этого использовались билинейная интерполяция[12] и последующее дублирование пикселей для приведения изображения к первоначальному размеру, чтобы сделать артефакты наложения (в данном случае ступеньки на контурных линиях) более наглядными. Как и в примере 4.7, артефакты наложения могут быть сделаны значительно менее заметными при помощи размывания изображения до пере-

а б в

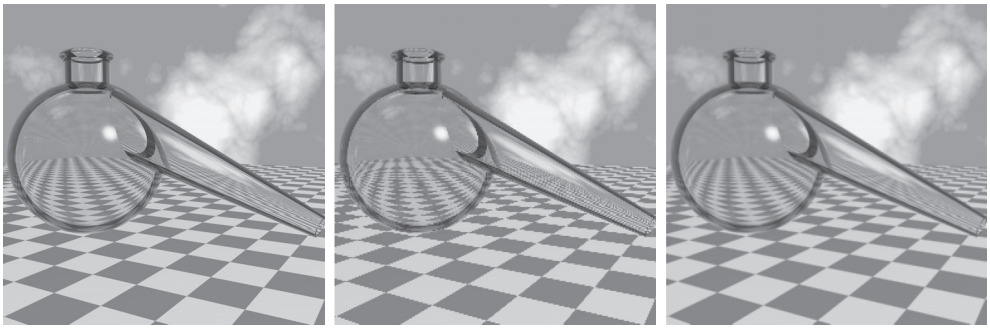


Рис. 4.18. Иллюстрация возникновения ступенчатости. (*а*) Сгенерированное на компьютере цифровое изображение размерами 1024×1024 пикселя без видимых артефактов наложения. (*б*) Результат уменьшения размеров изображения (*а*) до 25% исходных размеров при помощи билинейной интерполяции. (*в*) Результат размывания изображения (*а*) усредняющим фильтром по окрестности 5×5 перед уменьшением размеров. Изображение размыто слегка больше, чем (*б*), но артефактов наложения больше не видно. (Исходное изображение предоставлено Д. П. Митчеллом, Mental Landscape, LLC)

[12] В данном случае эта операция сводится к прореживанию — стиранию трех строк и столбцов из каждых четырех. — *Прим. перев.*

дискретизации. Рис. 4.18(*в*) — результат использования усредняющего фильтра по окрестности 5×5 перед уменьшением размеров. Как видно из этого рисунка, ступенчатость существенно снижена. Способ уменьшения и увеличения размеров изображения на рис. 4.18(*в*) был тем же, что и при формировании изображения рис. 4.18(*б*). ■

Пример 4.9. Иллюстрация возникновения ступенчатости при увеличении размеров изображения.

■ В предыдущих двух примерах для увеличения размеров малого передискретизованного изображения использовалось дублирование пикселей. Как показано на рис. 4.19, такой подход, вообще говоря, не является предпочтительным. На рис. 4.19(*а*) показана центральная часть изображения рис. 4.18(*а*) размерами 256×256 пикселей, увеличенная до размера 1024×1024 путем повторения пикселей. Хорошо заметна ступенчатость линий. Увеличенное изображение на рис. 4.19(*б*) было получено из того же фрагмента, однако использовалась билинейная интерполяция. В данном случае контуры выглядят значительно более гладкими. Например, контуры горлышка колбы и большие клетки шахматной доски на (*б*) не настолько ступенчаты, как на (*а*). ■

Муаровые узоры

Прежде чем завершить этот раздел, рассмотрим еще один тип артефактов, называемый *муаром*[13], который иногда возникает в результате дискретизации сцен с периодическими или почти периодическим объектами. В оптике муаровыми узорами называют картины биений (интерференций), возникающие при на-

а *б*

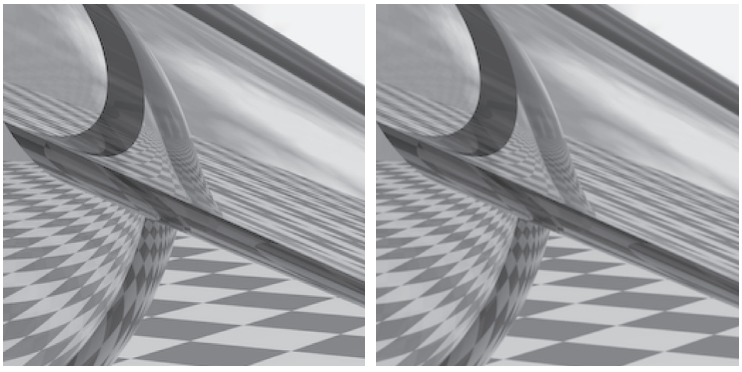


Рис. 4.19. Увеличение изображения. (*а*) Изображение размерами 1024×1024 пикселей — увеличенный центральный фрагмент изображения рис. 4.18(*а*) размерами 256×256, полученный повторением пикселей. (*б*) Результат увеличения того же фрагмента при помощи билинейной интерполяции; заметно значительное снижение ступенчатости

[13] Термин «муар» (фр. *moiré*) — французское слово, а не имя какого-то человека, появилось в оригинале у ткачей, которые первыми обратили внимание на интерференционные картины, заметные на некоторых видах тканей; термин также является корнем слова *мохер* (*mohair*) — материала, изготавливаемого из шерсти ангорских коз.

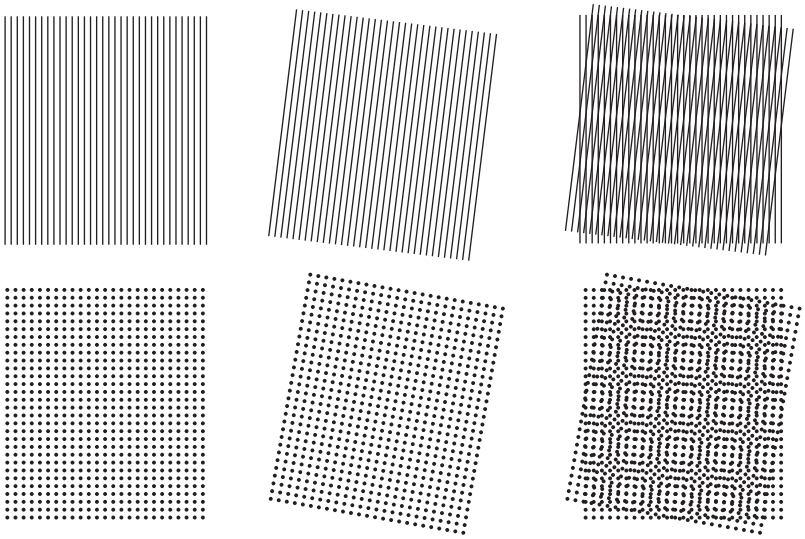


Рис. 4.20. Примеры эффекта муара. Представлены штриховые рисунки, а не дискретизованные образы. Наложение одного рисунка на другой математически эквивалентно перемножению рисунков

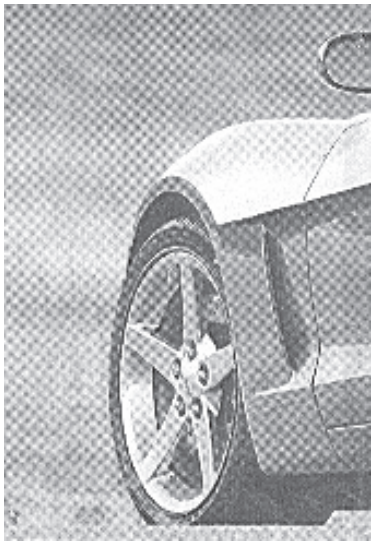


Рис. 4.21. Газетное изображение, дискретизованное с разрешением 75 dpi; хорошо видна картина муара. Данный муар — результат интерференции печатного изображения, растр которого повернут под углами ±45°, и вертикально-горизонтальной ориентации сетки сканирования при оцифровке изображения

ложении двух решеток с приблизительно равными шагами. Подобные узоры возникают повседневно. Их можно заметить, например, при перехлесте тюлевых занавесок или на некоторых телевизионных кадрах при интерференции полосатых материалов сцены и линий телевизионного растра. В цифровой обработке изображений проблема обычно возникает при сканировании изображений, опубликованных в печатных изданиях, таких как газеты и журналы, или же изображений, содержащих объекты с периодическими структурами, шаг которых сравним с шагом между отсчетами. Важно заметить, что муаровые узоры являются более общим явлением, чем артефакты дискретизации. Например, на рис. 4.20 показан эффект муара, возникающий на штриховых рисунках, которые не были дискретизованы, тем не менее наложение одного рисунка на другой создает картину биения с частотами, отсутствующими на обоих исходных рисунках. Обратим также внимание на муар-эффект, вызываемый двумя рисунками из точек, поскольку он является предметом нижеследующего обсуждения.

Газеты и другие печатные материалы используют так называемые *растровые элементы*, являющиеся черными точками или эллипсами;

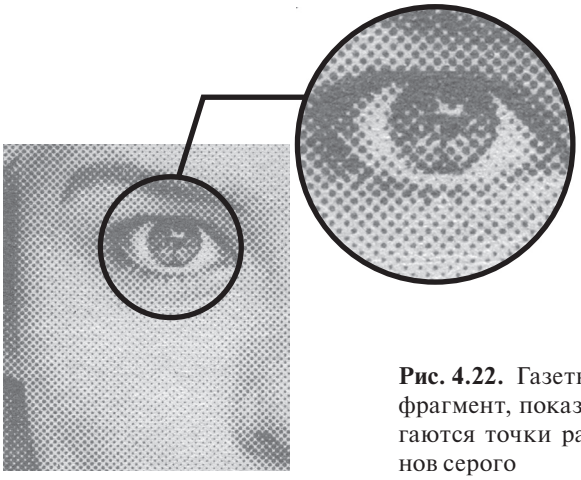


Рис. 4.22. Газетное изображение и его увеличенный фрагмент, показывающий, как выглядят и располагаются точки растра для воспроизведения полутонов серого

для воспроизведения полутонов используются различные схемы изменения их размеров и местоположения. Как правило, в полиграфии применяют следующие цифры разрешений печати: в газетах используется 75 точек на дюйм (dots per inch, *dpi*), журналы печатаются при 133 dpi, а высококачественные издания — при 175 dpi. На рис. 4.21 показано, что получается, когда газетная иллюстрация сканируется с разрешением 75 dpi. Решетка дискретизации (которая ориентирована по горизонтали и вертикали) и рисунки точек газетного изображения (ориентированные под углами $\pm 45^\circ$) взаимодействуют и создают картину однородного муара, в результате чего изображение выглядит пятнистым. (Методы подавления интерференционной картины муара рассматриваются в разделе 4.10.2.)

> В цветной печати для создания ощущения цвета используются красные, зеленые и синие точки[14].

В качестве связанного вопроса на рис. 4.22 представлено газетное изображение, дискретизованное с разрешением 400 dpi для устранения эффекта муара. Увеличение области, окружающей правый глаз модели, показывает, как изменение размеров точек используется для создания полутонов серого. Размеры точек обратно пропорциональны яркости изображения. На светлых областях точки маленькие и почти отсутствуют (см., например, область белка глаза). На светло-серых участках точки больше, как видно на области ниже глаза. В более темных областях, когда размер точки превышает некоторый выбранный размер (обычно 50%), точки могут сливаться вдоль двух выбранных направлений, образуя связанную сетку (см., например, левый уголок глаза). В некоторых случаях точки сливаются только вдоль одного направления, как видно справа вверху, ниже брови.

[14] На самом деле в цветной печати используются точки голубого, пурпурного и желтого цветов, являющихся дополнительными к красному, зеленому и синему цветам (см. главу 6). — *Прим. перев.*

4.5.5. Двумерное дискретное преобразование Фурье и его обращение

Выкладки, подобные изложенным в разделах 4.3 и 4.4, приводят к следующему двумерному *дискретному преобразованию Фурье* (ДПФ):

$$F(u,v)=\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)e^{-i2\pi(ux/M+vy/N)}, \qquad (4.5\text{-}15)$$

где $f(x,y)$ — цифровое изображение размерами $M\times N$. Как и в одномерном случае, выражение (4.5-15) должно быть вычислено для всех значений дискретных переменных u и v в диапазонах $u = 0, 1, 2, ..., M-1$ и $v = 0, 1, 2, ..., N-1$[15].

> Иногда постоянный множитель $1/MN$ вместо формулы обратного преобразования ставят в формулу прямого преобразования Фурье; иногда эту константу выражают как $1/\sqrt{MN}$ и включают в формулы как прямого, так и обратного преобразований, что создает более симметричный вид дискретной пары преобразований Фурье. Любая из указанных формулировок корректна при условии, что все выкладки единообразны по стилю.

Если имеется преобразование $F(u, v)$, можно получить $f(x,y)$ при помощи *обратного дискретного преобразования Фурье* (обратного ДПФ)

$$f(x,y)=\frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}F(u,v)e^{i2\pi(ux/M+vy/N)} \qquad (4.5\text{-}16)$$

для $x = 0, 1, 2, ..., M-1$ и $y = 0, 1, 2, ..., N-1$. Выражения (4.5-15) и (4.5-16) составляют *пару двумерных дискретных преобразований Фурье*. Оставшаяся часть данной главы базируется на свойствах этих двух выражений и использовании их для фильтрации изображений в частотной области.

4.6. Некоторые свойства двумерного дискретного преобразования Фурье

В данном разделе мы познакомимся с несколькими свойствами прямого и обратного двумерного преобразования Фурье.

4.6.1. Взаимосвязи пространственных и частотных интервалов

Взаимосвязи между шагом дискретизации и соответствующими интервалами частотной области были рассмотрены в разделе 4.4.2. Предположим, что непрерывная функция $f(t, z)$ дискретизована в виде цифрового изображения $f(x,y)$, состоящего из $M\times N$ отсчетов, взятых, соответственно, вдоль направлений t и z.

---

[15] Как уже отмечалось в разделе 4.4.1, следует помнить, что в данной главе мы используем (t, z) и (μ, ν) для обозначения *непрерывных* переменных пространственной и частотной областей. Для двумерного *дискретного* случая используются (x, y) для пространственных переменных и (u, v) для переменных частотной области.

Пусть ΔT и ΔZ обозначают шаги между отсчетами (см. рис. 4.14). Тогда соответствующие дискретные интервалы в частотной области составят

$$\Delta u = \frac{1}{M\Delta T} \tag{4.6-1}$$

и

$$\Delta v = \frac{1}{N\Delta Z} \tag{4.6-2}$$

соответственно. Заметим, что интервалы между отсчетами в частотной области обратно пропорциональны как шагам пространственной дискретизации, так и общему числу отсчетов.

4.6.2. Сдвиг и поворот

Прямой подстановкой в выражения (4.5-15) и (4.5-16) можно показать, что Фурье-пара удовлетворяет следующим свойствам сдвига (задача 4.16):

$$f(x,y)e^{i2\pi(u_0x/M+v_0y/N)} \Leftrightarrow F(u-u_0,v-v_0) \tag{4.6-3}$$

и

$$f(x-x_0,y-y_0) \Leftrightarrow F(u,v)e^{-i2\pi(x_0u/M+y_0v/N)}. \tag{4.6-4}$$

То есть умножение $f(x, y)$ на экспоненциальную функцию означает сдвиг начала координат ДПФ в точку (u_0,v_0), и наоборот, умножение $F(u,v)$ на ту же экспоненциальную функцию, но с отрицательным значением аргумента сдвигает начало координат $f(x, y)$ в точку (x_0, y_0). Как иллюстрирует пример 4.13, сдвиг не оказывает влияния на амплитуду спектра $F(u, v)$.

Использование полярных координат

$$x = r\cos\theta \quad y = r\sin\theta \quad u = \omega\cos\varphi \quad v = \omega\sin\varphi$$

приводит к следующей паре:

$$f(r,\theta+\theta_0) \Leftrightarrow F(\omega,\varphi+\theta_0)\,, \tag{4.6-5}$$

которая показывает, что поворот $f(x,y)$ на угол θ_0 приводит к повороту $F(u,v)$ на тот же угол. И наоборот, поворот $F(u, v)$ означает поворот на тот же угол и $f(x, y)$.

4.6.3. Периодичность

Аналогично одномерному случаю прямое и обратное двумерные преобразования Фурье являются периодическими по направлениям u и v; т. е.

$$F(u,v) = F(u+k_1M,v) = F(u,v+k_2N) = F(u+k_1M,v+k_2N) \tag{4.6-6}$$

и

$$f(x,y) = f(x+k_1M,y) = f(x,y+k_2N) = f(x+k_1M,y+k_2N), \tag{4.6-7}$$

где k_1 и k_2 — целые.

Периодичность прямого и обратного преобразований Фурье является важным фактом в реализации алгоритмов, основанных на ДПФ. Рассмотрим одномерный спектр, показанный на рис. 4.23(*a*). Как объяснялось в разделе 4.4.1,

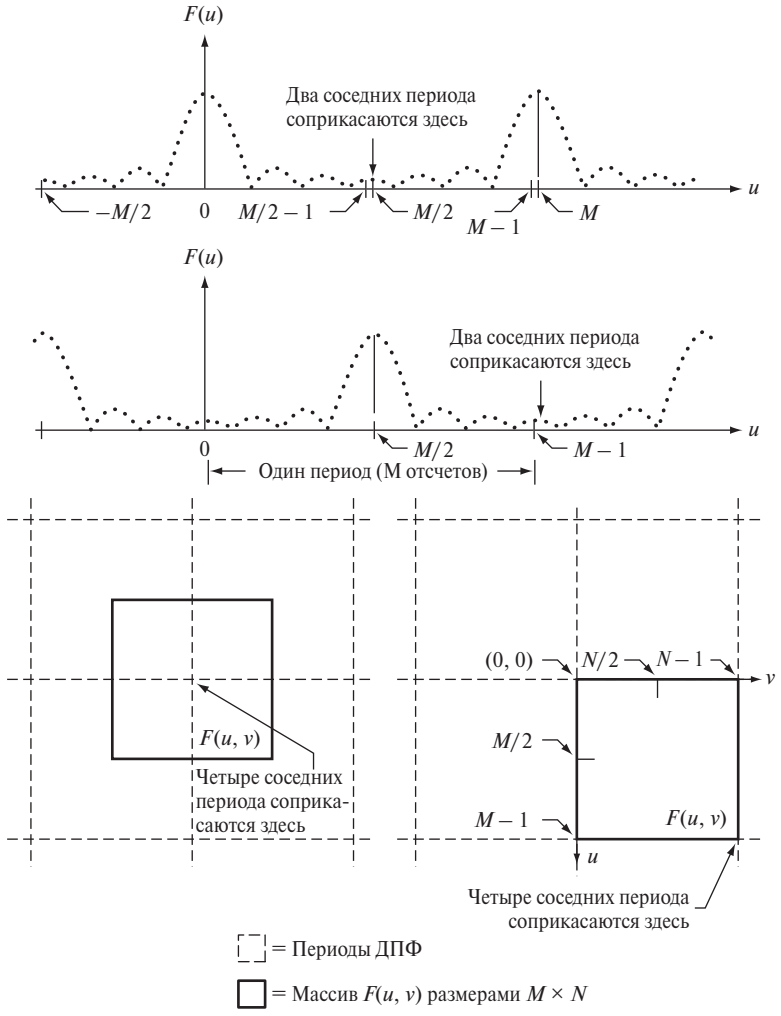


Рис. 4.23. Центрирование Фурье-преобразования. (*а*) Одномерное ДПФ, демонстрирующее бесконечное число периодов. (*б*) Сдвинутое ДПФ, полученное умножением $f(x)$ на $(-1)^x$ до вычисления $F(u)$. (*в*) Двумерное ДПФ, демонстрирующее бесконечное число периодов. Прямоугольник из непрерывных линий — массив данных $F(u, v)$, полученный при помощи выражения (4.5-15). Этот массив состоит из четырех четвертей периода. (*г*) Сдвинутое ДПФ, полученное умножением $f(x, y)$ на $(-1)^{x+y}$ до вычисления $F(u, v)$. Теперь данные содержат один полный центрированный период, как и на рисунке (*б*)

данные преобразования в интервале от 0 до $M-1$ состоят из двух половинок соседних периодов, соприкасающихся в точке $M/2$. Для отображения и фильтрации более удобно иметь на этом интервале полный период, на котором данные непрерывны, как на рис. 4.23(*б*). Из уравнения (4.6-3) следует, что

$$f(x)e^{i2\pi(u_0x/M)} \Leftrightarrow F(u-u_0).$$

Другими словами, умножение $f(x)$ на экспоненциальный член приводит к сдвигу данных, так что начало координат $F(0)$ оказывается в точке u_0. Если положить $u_0 = M/2$, то экспоненциальный член будет $e^{i\pi x} = (-1)^x$, поскольку x — целое. В таком случае

$$f(x)(-1)^x \Leftrightarrow F(u - M/2).$$

Т. е. умножение $f(x)$ на $(-1)^x$ приводит к сдвигу данных, и в результате $F(0)$ оказывается в центре интервала $[0, M-1]$, что соответствует рис. 4.23(*б*), как и требовалось.

Двумерный случай сложнее изображать графически, но принцип тот же самый, как показано на рис. 4.23(*в*). Вместо двух полупериодов теперь имеются четыре четверти периодов, стыкующиеся в точке $(M/2, N/2)$. Пунктирные прямоугольники означают бесконечное число периодов двумерного ДПФ. Как и в одномерном случае, визуализация упрощается, если сдвинуть данные так, чтобы $F(0, 0)$ оказалась в точке $(M/2, N/2)$. Принимая в уравнении (4.6-3) $(u_0, v_0) = (M/2, N/2)$, получим выражение

$$f(x,y)(-1)^{x+y} \Leftrightarrow F(u - M/2, v - N/2). \qquad (4.6\text{-}8)$$

Использование данного уравнения сдвигает данные, в результате чего $F(0,0)$ оказывается в центре *частотного прямоугольника*, заданного интервалами $[0, M-1]$ и $[0, N-1]$, что и требовалось; результат показан на рис. 4.23(*г*). Данные концепции иллюстрируются в настоящем разделе ниже как часть примера 4.11 и рис. 4.24.

4.6.4. Свойства симметрии

Важный результат из функционального анализа состоит в том, что любая действительная *или* комплексная функция $w(x,y)$ может быть выражена суммой четных и нечетных частей (*каждая из которых может являться действительной или комплексной*):

$$w(x,y) = w_e(x,y) + w_o(x,y), \qquad (4.6\text{-}9)$$

где четная и нечетная части определяются соответственно как

$$w_e(x,y) \triangleq \frac{w(x,y) + w(-x,-y)}{2} \qquad (4.6\text{-}10а)$$

и

$$w_o(x,y) \triangleq \frac{w(x,y) - w(-x,-y)}{2}. \qquad (4.6\text{-}10б)$$

Подстановка выражений (4.6-10а) и (4.6-10б) в (4.6-9) дает тождество $w(x,y) \equiv w(x,y)$, доказывая тем самым справедливость последних выражений. Из приведенных определений следует, что

$$w_e(x,y) = w_e(-x,-y) \qquad (4.6\text{-}11а)$$

и

$$w_o(x,y) = -w_o(-x,-y). \qquad (4.6\text{-}11б)$$

Четные функции называют *симметричными*, а нечетные — *антисимметричными*. Поскольку все индексы в прямом и обратном ДПФ являются неотрицательными, то говорить о вышеуказанных свойствах можно только со ссылкой на *центральную точку* последовательности, относительно которой функция является симметричной (антисимметричной). В терминах выражений (4.6-11), индексы справа от центральной точки одномерного массива считаются положительными, а слева — отрицательными. В нашей работе более удобно, когда индексы принимают только неотрицательные значения; в таком случае определения четности и нечетности принимают вид

$$w_e(x,y) = w_e(M-x, N-y) \tag{4.6-12a}$$

и

$$w_o(x,y) = -w_o(M-x, N-y), \tag{4.6-12b}$$

где, как обычно, M и N означают число строк и столбцов двумерного массива.

Из элементарного математического анализа известно, что произведение двух четных или двух нечетных функций даст в результате четную функцию, а произведение четной и нечетной функций будет нечетной функцией. Вдобавок к этому дискретная функция может быть нечетной только в том случае, если сумма всех ее отсчетов равна нулю. Это свойство приводит к важному результату, что

$$\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} w_e(x,y)w_o(x,y) = 0 \tag{4.6-13}$$

для любых дискретных четных и нечетных функций w_e и w_o. Другими словами, поскольку аргумент в формуле (4.6-13) является нечетным, то результат суммирования равен 0. Функции могут быть как действительными, так и комплексными.

> Чтобы удостовериться, что отсчеты нечетной функции в сумме равны нулю, рассмотрите один период одномерной функции синуса вокруг начала координат или любой другой интервал, охватывающий один период.

Пример 4.10. Четные и нечетные функции.

■ Представить себе четность и нечетность для непрерывных функций нетрудно, однако для дискретных последовательностей эти понятия не столь наглядны. Нижеследующая иллюстрация поможет прояснить рассмотренные выше идеи. Рассмотрим одномерную последовательность

$$f = \{f(0) \quad f(1) \quad f(2) \quad f(3)\} = \{2 \quad 1 \quad 1 \quad 1\},$$

в которой $M = 4$. Для проверки на четность должно выполняться условие $f(x) = f(4-x)$; т. е. требуется, чтобы

$$f(0) = f(4), \quad f(1) = f(3), \quad f(2) = f(2), \quad f(3) = f(1).$$

Поскольку $f(4)$ находится вне диапазона проверки и, следовательно, может быть любым, то значение $f(0)$ также оказывается несущественным для проверки на четность. Как видно, остальные значения удовлетворяют вышеуказанным условиям, следовательно, выбранная последовательность является четной. Фактически *любая* четная последовательность из четырех элементов должна иметь вид

$$\{a \quad b \quad c \quad b\}.$$

То есть в 4-элементной четной последовательности должны быть равны только второй и последний элементы.

Нечетная последовательность имеет интересное свойство, что ее первый член, $w_o(0,0)$, всегда равен нулю, — факт, следующий напрямую из (4.6-10b). Рассмотрим одномерную последовательность

$$g = \{g(0) \quad g(1) \quad g(2) \quad g(3)\} = \{0 \quad -1 \quad 0 \quad 1\}.$$

Тот факт, что эта последовательность нечетная, легко подтверждается проверкой условия $g(0) = -g(4 - x)$. Например, $g(1) = -g(3)$. Любая 4-элементная нечетная последовательность имеет вид

$$\{0 \quad a \quad 0 \quad -a\}.$$

Таким образом, при четном M одномерная нечетная последовательность имеет свойство, что ее элементы в точках 0 и $M/2$ всегда равны нулю. При нечетном M первый элемент по-прежнему должен быть равен 0, а оставшиеся элементы, формирующие пары, должны быть равными по величине, но с противоположными знаками.

Предыдущее обсуждение показывает, что свойство четности или нечетности последовательности зависит также от длины последовательности. Например, уже было показано, что последовательность {0 –1 0 1} является нечетной. Однако последовательность {0 –1 0 1 0}, в которой добавлен еще один нулевой элемент, уже не является ни четной, ни нечетной, хотя ее «основная» структура выглядит нечетной. Это важная проблема в интерпретации результатов ДПФ. Позже в данном разделе будет показано, что ДПФ четных и нечетных функций обладают некоторыми весьма важными характеристиками. Так, понимание того, что функция является четной или нечетной, часто играет ключевую роль в возможности интерпретировать результаты преобразования изображения, основанные на ДПФ.

Те же основные соображения справедливы и в двумерном случае. Например, двумерная последовательность размерами 6×6

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & -2 & 0 & 2 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

В качестве упражнения покажите, что данный двумерный массив является нечетным, используя для этого выражение (4.6-12б).

является нечетной. Однако добавление еще по одной строке и столбцу из нулей даст в результате массив, который не будет ни четным, ни нечетным. Обратим внимание, что внутренняя структура данного массива представляет собой ма-

ску Собела, рассматривавшуюся в разделе 3.6.4. Мы еще вернемся к этой маске в примере 4.15. ■

На основе рассмотренных выше концепций можно установить ряд важных свойств симметрии прямого и обратного ДПФ. Часто используемым свойством является то, что Фурье-образ *действительной* функции $f(x, y)$ будет *симметрично сопряженным*:

$$F^*(u,v) = F(-u,-v). \tag{4.6-14}$$

Если $f(x, y)$ является *мнимой*, то ее Фурье-образ будет *антисимметрично сопряженным*: $F^*(-u, -v) = -F(u, v)$. Доказательство утверждения (4.6-14) следующее:

$$F^*(u,v) = \left[\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-i2\pi(ux/M+vy/N)}\right]^* = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f^*(x,y)e^{i2\pi(ux/M+vy/N)} =$$
$$= \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-i2\pi([-u]x/M+[-v]y/N)} = F(-u,-v),$$

где третий шаг следует из того, что $f(x, y)$ — действительная функция. Аналогичный подход может быть использован для доказательства того, что Фурье-образ мнимой функции является антисимметрично сопряженным.

Таблица 4.1. Некоторые свойства симметрии прямого и обратного ДПФ. $R(u, v)$ и $I(u, v)$ являются действительной и мнимой частями $F(u, v)$ соответственно. Термин *комплексная* означает, что действительная и мнимая части функции ненулевые

	Пространственная область†		Частотная область†
1)	$f(x, y)$ действительная	$\Leftrightarrow$	$F^*(u, v) = F(-u, -v)$
2)	$f(x, y)$ мнимая	$\Leftrightarrow$	$F^*(-u, -v) = -F(u, v)$
3)	$f(x, y)$ действительная	$\Leftrightarrow$	$R(u, v)$ четная, $I(u, v)$ нечетная
4)	$f(x, y)$ мнимая	$\Leftrightarrow$	$R(u, v)$ нечетная, $I(u, v)$ четная
5)	$f(-x, -y)$ действительная	$\Leftrightarrow$	$F^*(u, v)$ комплексная
6)	$f(-x, -y)$ комплексная	$\Leftrightarrow$	$F(-u, -v)$ комплексная
7)	$f^*(x, y)$ комплексная	$\Leftrightarrow$	$F^*(-u, -v)$ комплексная
8)	$f(x, y)$ действительная и четная	$\Leftrightarrow$	$F(u, v)$ действительная и четная
9)	$f(x, y)$ действительная и нечетная	$\Leftrightarrow$	$F(u, v)$ мнимая и нечетная
10)	$f(x, y)$ мнимая и четная	$\Leftrightarrow$	$F(u, v)$ мнимая и четная
11)	$f(x, y)$ мнимая и нечетная	$\Leftrightarrow$	$F(u, v)$ действительная и нечетная
12)	$f(x, y)$ комплексная и четная	$\Leftrightarrow$	$F(u, v)$ комплексная и четная
13)	$f(x, y)$ комплексная и нечетная	$\Leftrightarrow$	$F(u, v)$ комплексная и нечетная

† Напомним, что x, y, u и v являются *дискретными* (целыми) переменными, причем x и u находятся в диапазоне $[0, M - 1]$, а y и v — в диапазоне $[0, N - 1]$. Утверждение, что комплексная функция является четной, означает, что u действительная u мнимая ее части являются четными; аналогично — для нечетных комплексных функций.

Симметрично сопряженную функцию также называют симметрично эрмитовой. Иногда используется термин антиэрмитов, означающий «антисимметрично сопряженный».

В табл. 4.1 приведен список симметрий и связанных с этим свойств ДПФ, которые полезны в цифровой обработке изображений. Напомним, что двойная стрелка означает Фурье-пару; т.е. в любой строке таблицы свойства для выражения справа справедливы для Фурье-преобразования функции, свойства которой указаны слева, и наоборот. Например, строчку 5 следует читать: ДПФ действительной функции $f(x, y)$, в которой (x, y) заменено на $(-x, -y)$, будет равно $F^*(u, v)$, где $F(u, v)$ — ДПФ функции $f(x, y)$ — является комплексной функцией, и наоборот.

Пример 4.11. Одномерная иллюстрация свойств из табл. 4.1.

■ В отношении концепций четности и нечетности, рассмотренных выше и иллюстрированных примером 4.10, следующие одномерные последовательности и их преобразования являются короткими примерами свойств, приведенных в табл. 4.1. Числа в круглых скобках справа — значения отдельных элементов $F(u)$, а в нижних двух строчках также и значения $f(x)$ в левом столбце.

Свойство	$f(x)$		$F(u)$
3	$\{1\ \ 2\ \ 3\ \ 4\}$	$\Leftrightarrow$	$\{(10)\ (-2+2i)\ (-2)\ (-2-2i)\}$
4	$i\{1\ \ 2\ \ 3\ \ 4\}$	$\Leftrightarrow$	$\{(2{,}5i)\ (0{,}5-0{,}5i)\ (-0{,}5i)\ (-0{,}5-0{,}5i)\}$
8	$\{2\ \ 1\ \ 1\ \ 1\}$	$\Leftrightarrow$	$\{(5)\ (1)\ (1)\ (1)\}$
9	$\{0\ \ -1\ \ 0\ \ 1\}$	$\Leftrightarrow$	$\{(0)\ (2i)\ (0)\ (-2i)\}$
10	$i\{2\ \ 1\ \ 1\ \ 1\}$	$\Leftrightarrow$	$\{(5i)\ (i)\ (i)\ (i)\}$
11	$i\{0\ \ -1\ \ 0\ \ 1\}$	$\Leftrightarrow$	$\{(0)\ (-2)\ (0)\ (2)\}$
12	$\{(4+4i)\ (3+2i)\ (0+2i)\ (3+2i)\}$	$\Leftrightarrow$	$\{(10+10i)\ (4+2i)\ (-2+2i)\ (4+2i)\}$
13	$\{(0+0i)\ (1+1i)\ (0+0i)\ (-1-i)\}$	$\Leftrightarrow$	$\{(0+0i)\ (2-2i)\ (0+0i)\ (-2+1i)\}$

Например, из свойства 3 видно, что Фурье-преобразование действительной функции с элементами $\{1\ \ 2\ \ 3\ \ 4\}$ будет иметь действительную часть $\{10\ \ -2\ \ -2\ \ -2\}$, которая является четной, и мнимую $\{0\ \ 2\ \ 0\ \ -2\}$, которая нечетная. Свойство 8 говорит, что Фурье-преобразование действительной четной функции также будет действительным и четным. Свойство 12 показывает, что Фурье-преобразование комплексной четной функции также будет комплексным и четным. Примеры остальных свойств анализируются аналогичным образом. ■

Пример 4.12. Доказательства некоторых свойств симметрии ДПФ из табл. 4.1.

■ В данном примере будут доказаны некоторые из важных свойств табл. 4.1, чтобы продолжить ознакомление с ними, а также создать основу для решения некоторых из задач в конце данной главы. Мы будем доказывать только свойства

в правой части на основе свойств, заданных в левой части. Доказательства в обратную сторону осуществляются аналогичными способами.

Рассмотрим свойство 3, которое означает: если $f(x,y)$ является действительной функцией, то действительная часть ДПФ будет четной, а мнимая — нечетной. И наоборот, если действительная и мнимая части ДПФ являются соответственно четной и нечетной, то обратное ДПФ будет действительной функцией. Докажем это свойство формально следующим образом. $F(u,v)$, вообще говоря, является комплексной, значит, она может быть выражена в виде суммы действительной и мнимой частей: $F(u,v) = R(u,v) + iI(u,v)$. Тогда $F^*(u,v) = R(u,v) - iI(u,v)$, а $F(-u,-v) = R(-u,-v) + iI(-u,-v)$. Но как было доказано ранее, если $f(x,y)$ — действительная, то $F^*(u,v) = F(-u,-v)$, что с учетом предыдущих двух формул означает, что $R(u,v) = R(-u,-v)$ и $I(u,v) = -I(-u,-v)$. Учитывая выражения (4.6-11а) и (4.6-11б), это доказывает, что R является четной, а I — нечетной функцией.

Теперь докажем свойство 8. Если $f(x,y)$ — действительная, то из свойства 3 известно, что действительная часть $F(u,v)$ является четной. Таким образом, остается доказать, что если $f(x,y)$ — действительная u четная, то мнимая часть $F(u,v)$ равна 0 (что и будет означать, что F — действительная). Действия состоят в следующем:

$$F(u,v) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-i2\pi(ux/M+vy/N)},$$

что можно записать как

$$\begin{aligned}
F(u,v) &= \sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[f_r(x,y)]e^{-i2\pi(ux/M+vy/N)} = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[f_r(x,y)]e^{-i2\pi(ux/M)}e^{-i2\pi(vy/N)} = \\
&= \sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[\text{четная}][\text{четная} - i\,\text{нечетная}][\text{четная} - i\,\text{нечетная}] = \\
&= \sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[\text{четная}][\text{четная}\cdot\text{четная} - 2i\,\text{четная}\cdot\text{нечетная} - \text{нечетная}\cdot\text{нечетная}] = \\
&= \sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[\text{четная}\cdot\text{четная}] - 2i\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[\text{четная}\cdot\text{нечетная}] - \sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[\text{четная}\cdot\text{четная}] = \\
&= \text{действительная}.
\end{aligned}$$

Третий шаг следует из формулы Эйлера и того факта, что косинус и синус являются четной и нечетной функциями. Из свойства 8 также известно, что функция f, кроме того, что она действительная, также является четной. В предпоследней строке единственным мнимым является второй член, который согласно (4.6-13) равен нулю. Таким образом, если f является действительной и четной, то F — действительная. Что и требовалось доказать.

В заключение докажем справедливость свойства 6. Согласно определению ДПФ,

$$\Im\{f(-x,-y)\} = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(-x,-y)e^{-i2\pi(ux/M+vy/N)}.$$

Заметим, что здесь мы не делаем замену переменных. Мы выражаем ДПФ функции $f(-x, -y)$, поэтому просто подставляем ее в формулу, как поступили бы с любой другой функцией.

Вследствие периодичности $f(-x, -y) = f(M - x, N - y)$. Если сделать замену $m = M - x$ и $n = N - y$, то тогда

$$\mathfrak{F}\{f(-x,-y)\} = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,n)e^{-i2\pi(u[M-m]/M+v[N-n]/N)}.$$

(Чтобы удостовериться, что суммирования корректны, выполните одномерное преобразование и раскройте несколько первых членов вручную.) Поскольку $\exp\{-i2\pi(\text{целое})\} = 1$, следовательно?

$$\mathfrak{F}\{f(-x,-y)\} = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,n)e^{i2\pi(um/M+vn/N)} = F(-u,-v).$$

Этим завершается доказательство. ■

4.6.5. Фурье-спектр и фаза

Поскольку двумерное ДПФ является, вообще говоря, комплексным, оно может быть выражено в полярных координатах

$$F(u,v) = |F(u,v)|e^{i\phi(u,v)}, \tag{4.6-15}$$

где амплитуда

$$|F(u,v)| = \left[R^2(u,v) + I^2(u,v)\right]^{1/2} \tag{4.6-16}$$

называется *Фурье-спектром* (или *частотным спектром*), а

$$\phi(u,v) = \text{arctg}\left[\frac{I(u,v)}{R(u,v)}\right] \tag{4.6-17}$$

называется *фазовым углом* или просто *фазой*. Из раздела 4.2.1 напомним, что вычисление арктангенса должно учитывать все четыре квадранта, как, например, функция MATLAB atan2(Imag, Real).

Наконец, *энергетический спектр* определяется как

$$P(u,v) = |F(u,v)|^2 = R^2(u,v) + I^2(u,v). \tag{4.6-18}$$

Как и ранее, R и I — действительная и мнимая части $F(u, v)$, и все вычисления выполняются для дискретных переменных $u = 0, 1, 2, ..., M - 1$ и $v = 0, 1, 2, ..., N - 1$. Таким образом, $|F(u, v)|$, $\varphi(u, v)$ и $P(u, v)$ являются массивами размерами $M \times N$.

Фурье-преобразование действительной функции является симметрично сопряженным (см. выражение (4.6-14)); это означает, что ее спектр имеет центральную симметрию:

$$|F(u,v)| = |F(-u,-v)|. \tag{4.6-19}$$

Фаза демонстрирует центральную антисимметрию:

$$\phi(u,v) = -\phi(-u,-v)\,. \tag{4.6-20}$$

Из (4.5-15) следует, что

$$F(0,0) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\,;$$

это показывает, что член, соответствующий нулевой частоте, пропорционален среднему значению функции $f(x, y)$. То есть

$$F(0,0) = MN\,\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y) = MN\,\bar{f}(x,y)\,, \tag{4.6-21}$$

где $\bar{f}$ означает среднее значение f. Тогда

$$\left|F(0,0)\right| = MN\left|\bar{f}(x,y)\right|. \tag{4.6-22}$$

Коэффициент пропорциональности MN, как правило, достаточно велик, поэтому $|F(0, 0)|$ обычно является самой большой компонентой спектра, на несколько порядков превышая амплитуды остальных составляющих. Поскольку в начале координат частотные компоненты u и v равны нулю, $F(0, 0)$ часто называют *постоянной составляющей* преобразования (в оригинале *dc component*, где «dc» означает *direct current* (*англ.*), т. е. постоянный ток, или ток нулевой частоты) — терминология, идущая из электротехники.

Пример 4.13. Двумерный Фурье-спектр прямоугольной функции.

■ На рис. 4.24(*а*) показано изображение простой прямоугольной функции, а на рис. 4.24(*б*) — ее спектр, значения которого масштабированы в диапазон [0, 255] и представлены в виде изображения. Начала координат как пространственной, так и частотной областей находятся в верхнем левом углу. Как и должно быть, область вокруг начала координат Фурье-образа содержит самые большие значения (и поэтому выглядит на изображении ярче). Заметим однако, что все четыре угла спектра содержат аналогичные высокие значения. Причина этого — свойство периодичности, которое обсуждалось в предыдущем разделе. Как следует из уравнения (4.6-8), для центрирования спектра достаточно перед вычислением ДПФ умножить исходное изображение (*а*) на $(-1)^{x+y}$. Результат представлен на рис. 4.24(*в*), который много удобнее для визуализации (обратим внимание на симметрию относительно центральной точки). Поскольку компонента $F(0, 0)$ доминирует в спектре, динамический диапазон всех остальных составляющих на изображении сжат. Чтобы отобразить остальные детали, было использовано логарифмическое преобразование, как описано в разделе 3.2.2. На рис. 4.24(*г*) показан результат преобразования по формуле $\log(1 + |F(u, v)|)$. Налицо существенное увеличение заметности деталей. Большинство спектров, представленных в данном и последующих разделах, масштабированы именно таким способом.

Из уравнений (4.6-4) и (4.6-5) следует, что спектр нечувствителен к сдвигу изображения (абсолютная величина экспоненциального члена равна 1), но при вращении изображения он поворачивается на тот же угол. Рис. 4.25 иллюстрирует эти свойства. Спектр на рис. 4.25(*б*) идентичен спектру на рис. 4.24(*г*). Поскольку изображения на рис. 4.24(*а*) и 4.25(*а*) различны, ясно, что если их Фурье-спектры совпадают, то, значит, согласно выражению (4.6-15) должны

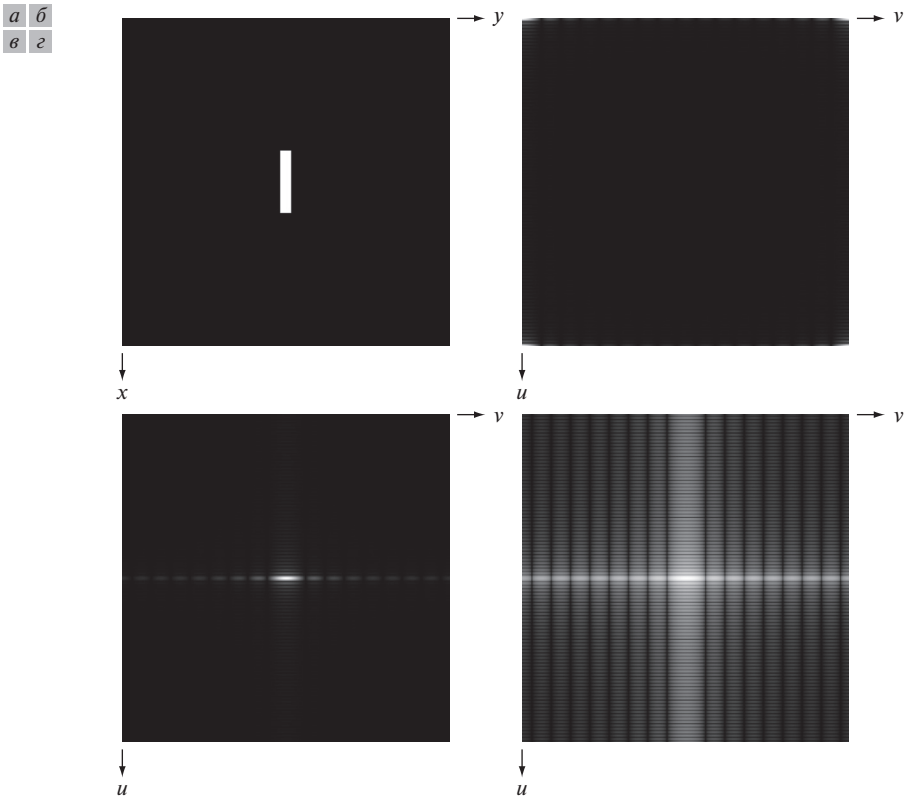
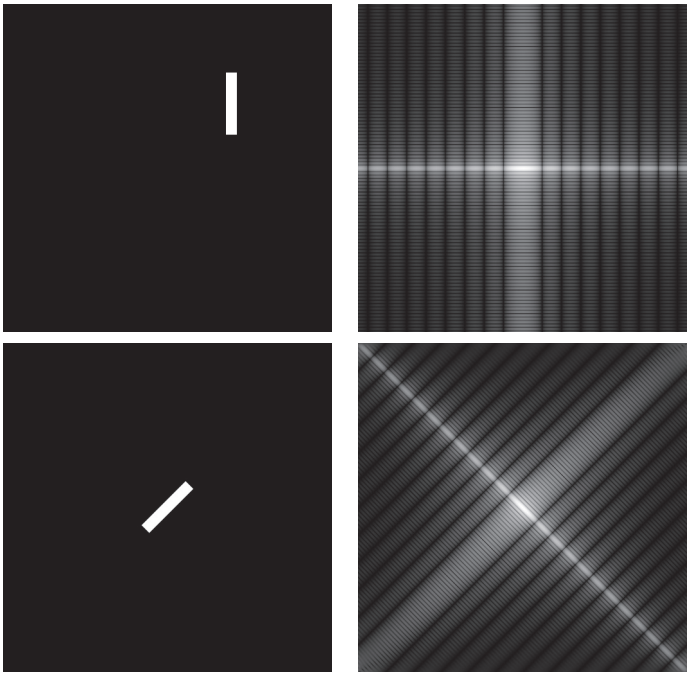


Рис. 4.24. (*а*) Изображение простой прямоугольной функции. (*б*) Спектр, демонстрирующий яркие пятна в четырех углах. (*в*) Центрированный спектр. (*г*) Результат после логарифмического преобразования, демонстрирующий улучшение видимости деталей. Поскольку прямоугольник вытянут в вертикальном направлении, точки перехода через ноль спектра в вертикальном направлении оказываются значительно ближе, чем в горизонтальном. Как обычно в книге, начала координат пространственной и частотной областей размещены вверху слева

быть различны их фазы. Это подтверждает рис. 4.26. На рис. 4.26(*а*) и (*б*) в виде изображений представлены массивы фаз Фурье-образов для изображений на рис. 4.24(*а*) и 4.25(*а*). Обратим внимание на непохожесть фазовых изображений несмотря на то, что единственным различием между исходными изображениями был простой сдвиг. Вообще визуальный анализ фазовых изображений дает мало интуитивной информации. Например, из-за того, что на рис. 4.26(*а*) прослеживается четкий рельеф под углом в 45°, можно было бы заключить, что это фазы изображения на рис. 4.25(*в*), а не на рис. 4.24(*а*). Фаза же повернутого изображения, как это видно на рис. 4.26(*в*), содержит четкие структуры под углом, существенно меньшим 45°. ■

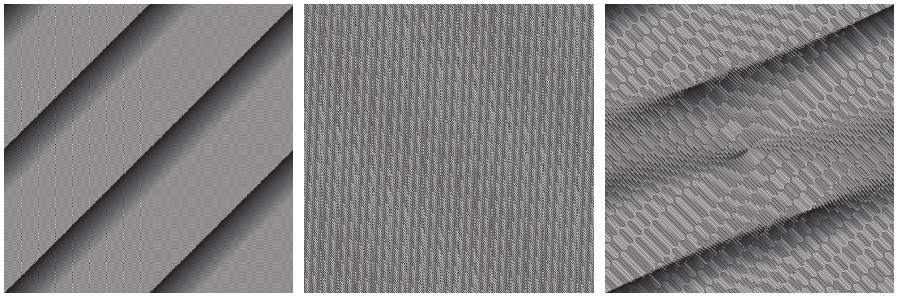
Компоненты спектра ДПФ задают амплитуды синусоид, из комбинации которых формируется изображение. На любой выбранной частоте Фурье-спектра изображения бóльшая амплитуда означает больший вклад синусоиды данной



а б
в г

Рис. 4.25. (*а*) Тот же прямоугольник, что и на рис. 4.24(*а*), но сдвинутый и (*б*) соответствующий спектр. (*в*) Повернутое изображение и (*г*) соответствующий спектр. Спектр, соответствующий сдвинутому изображению, идентичен спектру исходного изображениия на рис. 4.24(*а*)

частоты в изображение. И наоборот, малая амплитуда означает малый вклад данной синусоиды в изображение. Хотя, как подтверждает рис. 4.26, картину фазы трудно интерпретировать, тем не менее она весьма важна. Фаза является характеристикой смещения каждой из синусоид по отношению к началу координат. Таким образом, если амплитуда двумерного ДПФ есть массив, компоненты которого задают интенсивности на изображении, соответствующие фазы составляют массив смещений, который содержит значительную часть информации о том, где видимые объекты размещаются на изображении. Следующий пример помогает прояснить эти концепции.



а б в

Рис. 4.26. Массивы фаз, соответствующие изображениям: (*а*) центрированного прямоугольника на рис. 4.24(*а*); (*б*) сдвинутого прямоугольника на рис. 4.25(*а*); (*в*) повернутого прямоугольника на рис. 4.25(*в*)

Пример 4.14. Дальнейшая иллюстрация свойств Фурье-спектра и фазы.

■ На рис. 4.27(*а*) представлен портрет, а на рис. 4.27(*б*) — фаза его ДПФ. На последнем нет никаких деталей, которые бы помогли нашему визуальному анализу ассоциировать их с особенностями исходного изображения (нет даже видимой симметрии фаз). Однако важность фазы в задании формы исходного сигнала становится очевидной из рис. 4.27(*в*), который получен вычислением обратного ДПФ, используя исключительно фазовую информацию, т. е. подставив $|F(u, v)| = 1$ в выражение (4.6-15). Хотя информация о яркости и утеряна (напомним, что она содержится в Фурье-спектре), все основные детали на данном изображении безошибочно соответствуют исходному на рис. 4.27(*а*).

Рис. 4.27(*г*) был получен с использованием только Фурье-спектра, для чего экспоненциальному члену в (4.6-15) присваивалось значение 1 (что эквивалентно установке фазы в 0) и вычислялось обратное ДПФ. Результат нельзя назвать неожиданным. Он содержит лишь яркостную информацию, в которой доминирующей является постоянная составляющая. Никакой информации о форме нет, поскольку фаза была обнулена.

И наконец, рис. 4.27(*д*) и (*е*) еще раз демонстрируют главенство фазы в формировании детального содержания изображения. Рис. 4.27(*д*) был получен вычислением обратного ДПФ, используя в (4.6-15) спектр прямоугольника на рис. 4.24(*а*) и фазу портрета на рис. 4.27(*а*). Лицо женщины явно доминирует на результате. Наоборот, использование фазы прямоугольника на рис. 4.24(*а*)

а *б* *в*
г *д* *е*

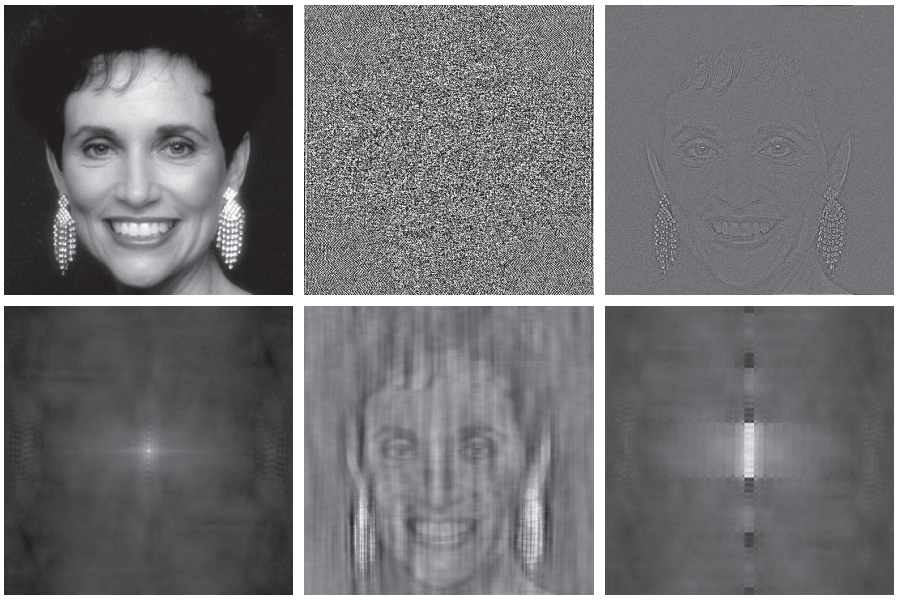


Рис. 4.27. (*а*) Портрет женщины. (*б*) Фаза. (*в*) Портрет, восстановленный при использовании только фазы. (*г*) Портрет, восстановленный при использовании только Фурье-спектра. (*д*) Восстановление при использовании фазы портрета и Фурье-спектра прямоугольника на рис. 4.24(*а*). (*е*) Восстановление при использовании фазы прямоугольника и Фурье-спектра портрета

и спектра портрета дает при обратном ДПФ изображение на рис. 4.27(*e*), в котором доминирует прямоугольник. ■

4.6.6. Двумерная теорема о свертке

Распространяя уравнение (4.4-10) на две переменные, получим в результате следующее выражение для двумерной *круговой свертки*:

$$f(x,y)\star h(x,y)=\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}f(m,n)h(x-m,y-n) \tag{4.6-23}$$

для $x = 0, 1, 2, ..., M-1$ и $y = 0, 1, 2, ..., N-1$. Как и (4.4-10), уравнение (4.6-23) задает один период двумерной периодической последовательности. Двумерная теорема о свертке задается соотношениями

$$f(x,y)\star h(x,y) \Leftrightarrow F(u,v)H(u,v), \tag{4.6-24}$$

и наоборот,

$$f(x,y)h(x,y) \Leftrightarrow F(u,v)\star H(u,v), \tag{4.6-25}$$

где F и H получены с использованием выражения (4.5-15), а двойная стрелка, как и ранее, указывает, что выражения в правой и левой частях образуют Фурье-пару. В оставшейся части данного раздела наш интерес будет сконцентрирован на соотношении (4.6-24), которое утверждает, что обратное ДПФ произведения $F(u,v)H(u,v)$ дает в результате $f(x,y)\star h(x,y)$ — двумерную пространственную свертку f и h. Аналогично ДПФ пространственной свертки дает в результате произведение преобразований в частотной области. Соотношение (4.6-24) является фундаментом линейной фильтрации и, как объясняется в разделе 4.7, создает основу для всех методов фильтрации, рассматриваемых в настоящей главе.

Поскольку мы здесь имеем дело с дискретными величинами, вычисление Фурье-преобразования выполняется с использованием алгоритма ДПФ. Если мы выбираем вычисление пространственной свертки при помощи обратного ДПФ произведения двух преобразований, во внимание должна приниматься периодичность, обсуждаемая в разделе 4.6.3. Мы приведем пример одномерной ситуации, а затем расширим выводы на двумерный случай. Левый столбец на рис. 4.28 отображает выполнение свертки двух функций f и h (каждая длиной в 400 элементов), используя одномерный эквивалент уравнения (3.4-2), которое, поскольку обе функции имеют одинаковый размер, записывается в виде

$$f(x)\star h(x)=\sum_{m=0}^{399}f(m)h(x-m).$$

Эффективные способы вычисления ДПФ будут рассматриваться в разделе 4.11.

Это уравнение идентично (4.4-10), *но* имеется требование к смещению x, ибо оно должно быть достаточно большим, обеспечивая возможность перевернутой версии h полностью сдвигаться по f. Иными словами, процедура состоит

из (1) зеркального переворота h вокруг начала координат (т. е. поворота на 180°) (рис. 4.28(*в*)), (2) сдвига перевернутой функции на величину x (рис. 4.28(*г*)) и (3) вычисления *полной* суммы произведений в правой части предыдущего уравнения для *каждого* из значений сдвига x. В обозначениях рис. 4.28 это означает умножение функции на рис. 4.28(*а*) на функцию на рис. 4.28(*г*) для *каждого*

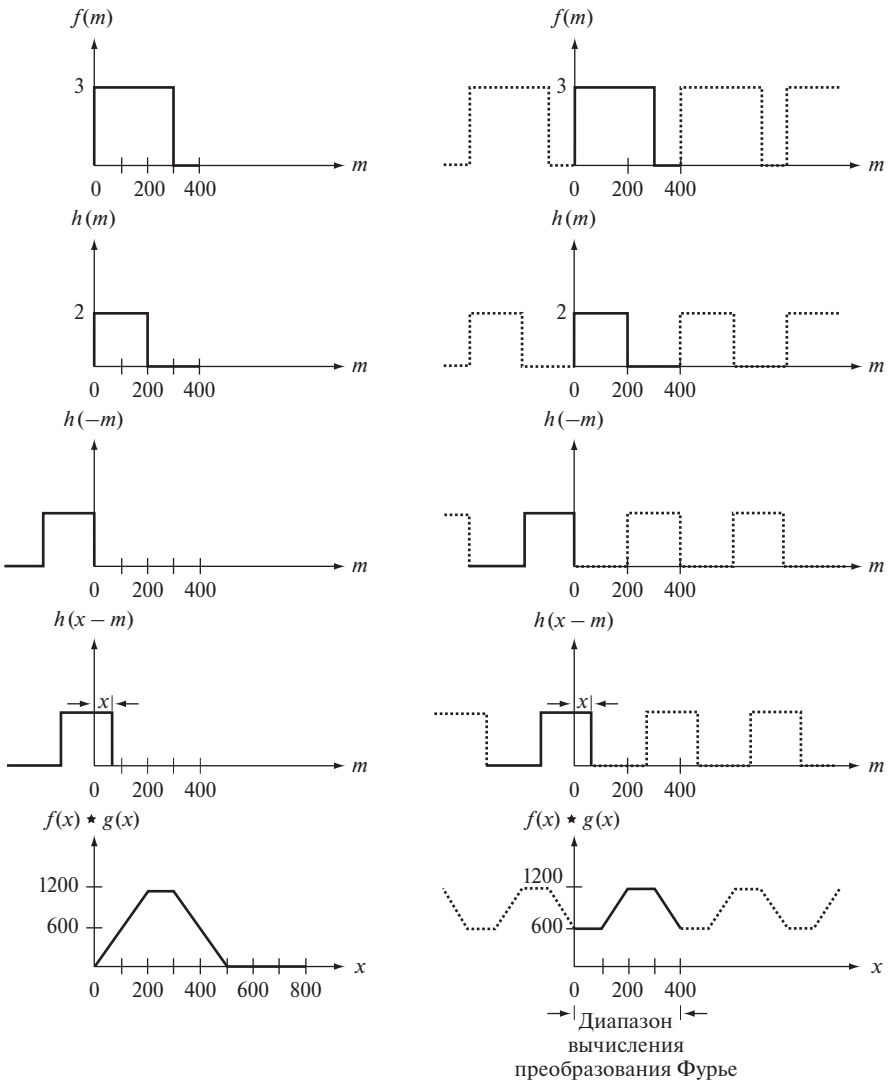


Рис. 4.28. Левый столбец: свертка двух дискретных функций, получаемая при использовании подхода, рассмотренного в разделе 3.4.2. Корректным является результат (*д*). Правый столбец: свертка тех же функций, но с учетом периодичности, являющейся следствием ДПФ. Обратим внимание на (*к*), где возникают ошибки перехлеста данных из соседних периодов, приводящие к неверным результатам свертки. Для получения корректных результатов следует использовать расширение функций (дополнение нулями)

значения x. Смещение x пробегает весь диапазон значений, требуемых для полного сдвига h по f. На рис. 4.28(*д*) показана свертка этих двух функций. Заметим, что свертка есть функция переменной смещения x и что для полного сдвига h по f диапазон изменения x составляет от 0 до 799.

Если для получения того же результата, что изображен в левом столбце рис. 4.28, используется ДПФ и теорема о свертке, то необходимо принимать во внимание периодичность, присущую выражениям, связанным с ДПФ. Это эквивалентно свертке двух периодических функций на рис. 4.28(*е*) и (*ж*). Процедура свертки та же самая, что только что рассматривалась, но функции теперь являются периодическими. Оперируя с этими двумя функциями так же, как и в предыдущем параграфе, получим результат, показанный на рис. 4.28(*к*), который, очевидно, является неверным. Поскольку в свертке участвуют периодические функции, то свертка также будет периодической. Близость областей данных соседних периодов на рис. 4.28 оказывается таковой, что они интерферируют друг с другом, вызывая так называемые *ошибки перехлеста*. В соответствии с теоремой о свертке, если вычислить ДПФ двух 400-элементных функций f и h, перемножить два преобразования и затем найти обратное ДПФ, то результатом был бы 400-элементный сегмент свертки, показанный на рис. 4.28(*к*) и являющийся неверным.

К счастью, имеется сравнительно простое решение проблемы ошибок перехлеста. Рассмотрим две функции $f(x)$ и $h(x)$ длиной A и B отсчетов соответственно. Можно показать ([Brigham, 1988]), что если мы добавим нули в конце каждой из функций так, чтобы они имели одинаковую длину, обозначаемую P, то ошибки перехлеста пропадают при выборе

$$P \geq A + B - 1. \qquad (4.6\text{-}26)$$

> Нули могут добавляться как в конце, так и в начале каждой из функций или же часть в начале и часть в конце функции. Но проще добавлять их в конце.

В нашем примере каждая из функций имеет по 400 точек, так что минимальное значение P, которое должно быть для отсутствия перехлеста, составляет $P = 799$; что означает, что необходимо добавить 399 нулей к концу каждой из функций. Этот процесс называется *дополнением нулями*. В качестве примера можете убедиться, что если периоды функций на рис. 4.28(*е*) и (*ж*) расширить добавлением в каждый из них как минимум 399 нулей, то результатом будет периодическая свертка, в которой каждый период идентичен корректному результату на рис. 4.28(*д*). Использование теоремы о свертке и ДПФ приводит к получению пространственной функции длиной в 799 точек, идентичной изображенной на рис. 4.28(*д*). Можно сделать вывод, что для получения одинаковых результатов при «прямом» использовании уравнения свертки в главе 3 и при подходе с использованием ДПФ в последнем случае функции должны дополняться нулями прежде, чем вычисляются их преобразования.

В двумерном случае продемонстрировать аналогичный пример труднее, но можно прийти к тем же выводам в отношении ошибок перехлеста и к необходимости дополнения функций нулями. Пусть $f(x, y)$ и $h(x, y)$ — два изображения (двумерных массива) размерами $A \times B$ и $C \times D$ элементов соответственно. Появле-

ния ошибок перехлеста при круговой свертке можно избежать при дополнении массивов нулями с соблюдением следующих условий:

$$f_p(x,y)=\begin{cases} f(x,y) & 0\le x\le A-1 \text{ и } 0\le y\le B-1 \\ 0 & A\le x\le P \text{ или } B\le y\le Q \end{cases} \tag{4.6-27}$$

и

$$h_p(x,y)=\begin{cases} h(x,y) & 0\le x\le C-1 \text{ и } 0\le y\le D-1 \\ 0 & C\le x\le P \text{ или } D\le y\le Q \end{cases} \tag{4.6-28}$$

при

$$P\ge A+C-1 \tag{4.6-29}$$

и

$$Q\ge B+D-1. \tag{4.6-30}$$

Результирующие дополненные нулями массивы будут иметь размеры $P\times Q$. Если оба изображения имели размеры $M\times N$, то требования будут иметь вид

$$P\ge 2M-1 \tag{4.6-31}$$

и

$$Q\ge 2N-1. \tag{4.6-32}$$

В разделе 4.7.2 дается пример, показывающий эффект от ошибок перехлеста на изображениях. Как правило, алгоритмы ДПФ производят вычисления с массивами четного размера быстрее, чем нечетного, поэтому можно рекомендовать выбирать в качестве P и Q минимальные четные числа, удовлетворяющие вышеуказанным условиям. Если массивы по размерам одинаковы, то в качестве P и Q стоит выбирать удвоенный размер массива.

Обе функции на рис. 4.28(*а*) и (*б*) принимают нулевые значения еще до конца интервала дискретизации, что удобно для дополнения нулями. Однако если одна или обе функции на конце интервала не равны нулю, то дополнение функции нулями для устранения ошибок перехлеста вызовет нарушение непрерывности (разрыв) функции. Это аналогично умножению исходной функции на прямоугольную функцию, что в частотной области может быть выражено как свертка исходного Фурье-образа с функцией sinc (см. пример 4.1). Это, в свою очередь, приведет к так называемой *спектральной утечке*, вызываемой высокочастотными компонентами sinc-функции. Утечка приводит к возникновению блочного эффекта на изображении. Хотя устранить утечку полностью невозможно, она может быть существенно уменьшена умножением дискретизуемой функции на другую функцию, которая плавно снижается до нуля на обоих концах функции, подавляя резкие переходы (и тем самым высокочастотные составляющие), имеющиеся у прямоугольной функции. Такой подход, называемый *выделение по окну* или *кадрирование*, является важным фактором, когда требуется достичь высокого качества восстановления (как, например, в графике высокого разрешения). Если вы столкнулись с необходимостью использования отсечения, хорошим подходом является применение двумерной гауссовой функции (см. раздел 4.8.3). Одним из преимуществ этой функции является то, что ее Фурье-спектр также является гауссовым, а значит, вызывающим малые утечки.

Таблица 4.2. Сводка определений, относящихся к ДПФ, с соответствующими выражениями

	Название	Выражение
1)	Дискретное преобразование Фурье (ДПФ) функции $f(x,y)$	$F(u,v)=\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)e^{-i2\pi(ux/M+vy/N)}$
2)	Обратное дискретное преобразование Фурье (обратное ДПФ) функции $F(u,v)$	$f(x,y)=\frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}F(u,v)e^{i2\pi(ux/M+vy/N)}$
3)	Представление в полярных координатах	$F(u,v)=\lvert F(u,v)\rvert e^{i\phi(u,v)}$
4)	Спектр	$\lvert F(u,v)\rvert=\left[R^2(u,v)+I^2(u,v)\right]^{1/2}$ $R=\mathrm{Re}(F);\quad I=\mathrm{Im}(F)$
5)	Фаза (фазовый угол)	$\phi(u,v)=\mathrm{arctg}\left[\frac{I(u,v)}{R(u,v)}\right]$
6)	Энергетический спектр	$P(u,v)=\lvert F(u,v)\rvert^2$
7)	Среднее значение	$\bar{f}(x,y)=\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)=\frac{1}{MN}F(0,0)$
8)	Периодичность (k_1 и k_2 — целые)	$F(u,v)=F(u+k_1M,v)=F(u,v+k_2N)=$ $=F(u+k_1M,v+k_2N)$ $f(x,y)=f(x+k_1M,y)=f(x,y+k_2N)=$ $=f(x+k_1M,y+k_2N)$
9)	Свертка	$f(x,y)\bigstar h(x,y)=\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}f(m,n)h(x-m,y-n)$
10)	Корреляция	$f(x,y)\star h(x,y)=\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}f^*(m,n)h(x+m,y+n)$
11)	Разделимость (сепарабельность)	Двумерное ДПФ может быть выполнено вычислением одномерных ДПФ по строкам (столбцам) и затем вычислением одномерных ДПФ по столбцам (строкам). См. раздел 4.11.1
12)	Вычисление обратного ДПФ, используя алгоритм прямого ДПФ	$MNf^*(u,v)=\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}F^*(u,v)e^{-i2\pi(ux/M+vy/N)}$ Выражение показывает, что подавая $F^*(u,v)$ на вход алгоритма вычисления прямого ДПФ, получаем $MNf^*(u,v)$. Выполнение операций комплексного сопряжения и деления на MN дает искомый результат. См. раздел 4.11.2

Таблица 4.3. Сводка Фурье-пар дискретного преобразования. Аналитические выражения в последних двух строках таблицы справедливы только для непрерывных переменных. Для использования с дискретными переменными непрерывные аналитические выражения должны быть дискретизованы

	Название	Фурье-пары дискретного преобразования
1)	Свойства симметрии	См. табл. 4.1
2)	Линейность	$af_1(x,y)+bf_2(x,y) \Leftrightarrow aF_1(u,v)+bF_2(u,v)$
3)	Сдвиг (общий случай)	$f(x,y)e^{i2\pi(u_0x/M+v_0y/N)} \Leftrightarrow F(u-u_0,v-v_0)$ $f(x-x_0,y-y_0) \Leftrightarrow F(u,v)e^{-i2\pi(ux_0/M+vy_0/N)}$
4)	Сдвиг в центр частотного прямоугольника ($M/2$, $N/2$)	$f(x,y)(-1)^{x+y} \Leftrightarrow F(u-M/2,v-N/2)$ $f(x-M/2,y-N/2) \Leftrightarrow F(u,v)(-1)^{u+v}$
5)	Поворот	$f(r,\theta+\theta_0) \Leftrightarrow F(\omega,\varphi+\theta_0)$ $x=r\cos\theta \quad y=r\sin\theta \quad u=\omega\cos\varphi \quad v=\omega\sin\varphi$
6)	Теорема о свертке†	$f(x,y)\bigstar h(x,y) \Leftrightarrow F(u,v)H(u,v)$ $f(x,y)h(x,y) \Leftrightarrow F(u,v)\bigstar H(u,v)$
7)	Теорема о корреляции†	$f(x,y)\star h(x,y) \Leftrightarrow F^*(u,v)H(u,v)$ $f^*(x,y)h(x,y) \Leftrightarrow F(u,v)\star H(u,v)$
8)	Дискретный единичный импульс	$\delta(x,y) \Leftrightarrow MN$
9)	Прямоугольник	$\text{rect}[a,b] \Leftrightarrow ab\dfrac{\sin(\pi ua)}{(\pi ua)}\dfrac{\sin(\pi vb)}{(\pi vb)}e^{-i\pi(ua+vb)}$
10)	Синус	$\sin(2\pi u_0x+2\pi v_0y) \Leftrightarrow i\dfrac{1}{2}[\delta(u+Mu_0,v+Nv_0)-\delta(u-Mu_0,v-Nv_0)]$
11)	Косинус	$\cos(2\pi u_0x+2\pi v_0y) \Leftrightarrow \dfrac{1}{2}[\delta(u+Mu_0,v+Nv_0)+\delta(u-Mu_0,v-Nv_0)]$

Следующие Фурье-пары выводятся лишь для непрерывных переменных, обозначаемых, как и ранее, t и z для пространственных и μ и v для частотных переменных. Эти результаты могут быть использованы для работы с ДПФ после дискретизации непрерывных форм.

12)	*Дифференцирование* (выражение справа предполагает, что $f(\pm\infty,\pm\infty)=0$)	$\left(\dfrac{\partial}{\partial t}\right)^m\left(\dfrac{\partial}{\partial z}\right)^n f(t,z) \Leftrightarrow (i2\pi\mu)^m(i2\pi v)^n F(\mu,v)$ $\dfrac{\partial^m f(t,z)}{\partial t^m} \Leftrightarrow (i2\pi\mu)^m F(\mu,v); \quad \dfrac{\partial^n f(t,z)}{\partial z^n} \Leftrightarrow (i2\pi v)^n F(\mu,v)$
13)	Гауссиан	$A2\pi\sigma^2 e^{-i2\pi^2\sigma^2(t^2+z^2)} \Leftrightarrow Ae^{-(\mu^2+v^2)/2\sigma^2}$ (A — константа)

† Предполагается, что функции расширены путем дополнения нулями. Свертка является ассоциативной, коммутативной и дистрибутивной операцией. Корреляция является дистрибутивной операцией.

Простой функцией отсечения является треугольник, центрированный в середине функции и спадающий до 0 на обоих концах функции. Он называется «окно Бартлетта» (в оригинале Bartlett window). Другими используемыми функциями являются окно Хэмминга (Hamming window) и окно Ханна (Hann window). Можно также использовать гауссову функцию. Мы вернемся к рассмотрению вопроса окон в разделе 5.11.5.

4.6.7. Краткое изложение свойств двумерного дискретного преобразования Фурье

В табл. 4.2 приведены основные определения, связанные с ДПФ, которые введены в настоящей главе. Разделимость рассматривается в разделе 4.11.1. Получение обратного преобразования Фурье с использованием алгоритма прямого преобразования обсуждается в разделе 4.11.2. Корреляция рассматривается в главе 12.

В табл. 4.3 приводятся некоторые важные пары преобразований Фурье. Хотя наше основное внимание обращено на дискретные функции, последние две строки таблицы представляют пары преобразований Фурье, которые могут быть получены лишь для непрерывных переменных (отметим использование обозначений непрерывных переменных). Они включены сюда, поскольку при правильной интерпретации являются весьма полезными в цифровой обработке изображений. Пара дифференцирования может быть использована для получения в частотной области эквивалента лапласиана, определенного уравнением (3.6.3) (задача 4.26). Пара гауссиана рассматривается в разделе 4.7.4.

Табл. 4.1—4.3 дают краткое изложение различных свойств, полезных при работе с ДПФ. Многие из этих свойств являются важнейшими элементами при обсуждении материала оставшейся части настоящей главы, а некоторые используются и в последующих главах.

4.7. Основы фильтрации в частотной области

В данном разделе закладываются основы для всех методов фильтрации, рассматриваемых в оставшейся части главы.

4.7.1. Дополнительные характеристики частотной области

Мы начинаем с того, что согласно выражению (4.5-15) *каждый* элемент Фурье-образа $F(u,v)$ содержит *все* отсчеты функции $f(x,y)$, умноженные на значения экспоненциальных членов. Поэтому обычно, за исключением тривиальных случаев, невозможно установить прямое соответствие между характерными деталями изображения и его образа. Однако некоторые общие утверждения относительно взаимосвязи частотных составляющих Фурье-образа и пространственных особенностей изображения могут быть сделаны. Например, поскольку частота прямо связана со скоростью пространственных изменений сигнала, то интуитивно ясно, что частоты в Фурье-преобразовании связаны с вариацией яркости

на изображении. В разделе 4.6.5 было показано, что наиболее медленно меняющаяся (постоянная) частотная составляющая ($u = v = 0$) пропорциональна средней яркости изображения. Низкие частоты, отвечающие точкам вблизи начала координат Фурье-преобразования, соответствуют медленно меняющимся компонентам изображения. На изображении комнаты, например, они могут соответствовать плавным изменениям яркости стен и пола. По мере удаления от начала координат более высокие частоты начинают соответствовать все более и более быстрым изменениям яркости, которые суть границы объектов и другие детали изображения, характеризуемые резкими изменениями яркости.

Методы фильтрации в частотной области основаны на модифицировании Фурье-образа для достижения конкретных целей и выполнении затем обратного ДПФ, чтобы вернуться в пространственную область изображения, как излагалось в разделе 2.6.7. Из (4.6-15) следует, что двумя компонентами Фурье-преобразования, к которым у нас имеется доступ, являются амплитуда преобразования (Фурье-спектр) и фаза. Основные свойства этих двух компонент рассмотрены в разделе 4.6.5. Там было показано, что визуальный анализ фазовой компоненты, как правило, имеет мало смысла. Зато спектр дает некоторые полезные указания на общие характеристики исходного изображения. Например, рассмотрим рис. 4.29(*а*), на котором представлено увеличенное примерно в 2500 раз изображение интегральной схемы, полученное при помощи сканирующего электронного микроскопа. Помимо собственно представляющей интерес конструкции, мы видим две характерные особенности. Это резкие контуры деталей, проходящие под углом примерно ±45°, и два белых выступа окалины, возникшие в результате неудачно проведенной термической обработки. В Фурье-спектре на рис. 4.29(*б*) хорошо видны диагональные составляющие, которые отвечают упомянутым контурам. При внимательном рассматривании области, расположенной вдоль вертикальной оси, можно заметить частотную составляющую, слегка повернутую против часовой стрелки. Наличие этой составляющей обусловлено контурами выступов окалины. Обратим внимание на то, как угол поворота этой частотной составляющей, связан с отклонением длинного белого элемента от горизонтали. Обратим также внимание на поло-

а *б*

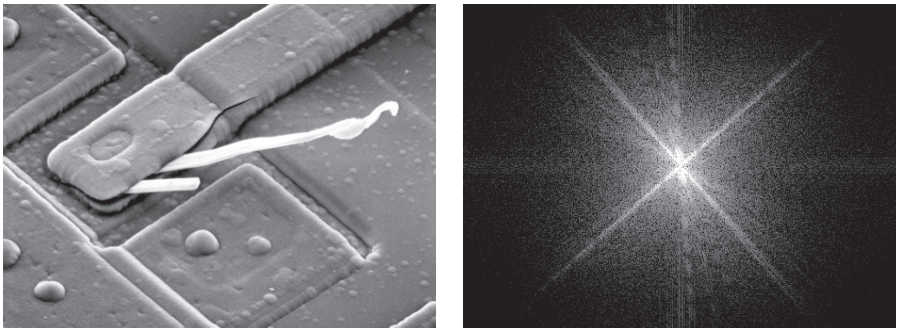


Рис. 4.29. (*а*) Изображение поврежденной интегральной схемы, полученное при помощи сканирующего электронного микроскопа. (*б*) Фурье-спектр изображения (*а*). (Исходное изображение предоставил д-р Д. М. Хьюдек, Brockhouse Institute for Material Research, университет Мак-Мастер, г. Гамильтон, шт. Онтарио, Канада)

жение нулей этой частотной составляющей связанное с малым поперечным размером выступов окалины.

Это типичные варианты связей, которые вообще могут быть установлены между частотной и пространственной областью. Как будет показано позже в данной главе, даже такие очевидные типы связей вместе с упомянутой выше зависимостью между частотными составляющими и скоростью изменения яркости на изображении могут приводить ко многим очень полезным результатам. В следующем разделе будут рассмотрены эффекты, возникающие при модификации различных частотных диапазонов Фурье-преобразования изображения на рис. 4.29(*a*).

4.7.2. Основы частотной фильтрации

Фильтрация в частотной области заключается в модифицировании Фурье-образа изображения и последующем выполнении обратного преобразования для получения обработанного результата. Это утверждение можно сформулировать в виде следующего основного уравнения частотной фильтрации:

$$g(x,y) = \mathfrak{F}^{-1}[H(u,v)F(u,v)], \tag{4.7-1}$$

где $F(u,v)$ — ДПФ исходного изображения $f(x,y)$ размерами $M \times N$, $H(u,v)$ — *фильтр-функция* (также называемая просто *фильтром* или *передаточной функцией фильтра*), $\mathfrak{F}^{-1}$ — обратное ДПФ, а $g(x,y)$ — результат фильтрации (выходное изображение). Функции F, H, и g представляют собой массивы размерами $M \times N$, как и исходное изображение. Произведение $H(u,v)F(u,v)$ формируется как поэлементное произведение массивов (см. раздел 2.6.1). Фильтр-функция модифицирует Фурье-преобразование исходного изображения, чтобы получить на выходе обработанное изображение $g(x,y)$. Задание функции $H(u,v)$ значительно упростится, если использовать функции, симметричные относительно своего центра; это в свою очередь требует, чтобы $F(u,v)$ также была центрирована тем же самым образом. Как объясняется в разделе 4.6.3, это достигается умножением исходного изображения на $(-1)^{x+y}$ перед выполнением прямого ДПФ[16].

> Если H является действительной и симметричной, а f — действительной (как это обычно и бывает в реальной ситуации), то обратное ДПФ в (4.7-1) теоретически должно в результате давать действительные значения. Однако на практике обратное преобразование обычно имеет паразитные комплексные члены из-за округлений и других неточностей вычислений. Поэтому для формирования g обычно берут действительную часть обратного ДПФ.

[16] Многие программные реализации двумерного ДПФ (например MATLAB) не осуществляют центрирования преобразования Фурье. Это подразумевает, что фильтр-функция должна быть сформирована соответствующим образом, чтобы совпадать с форматом данных нецентрированного преобразования (т. е. иметь начало координат слева вверху). Конечным результатом является то, что такой фильтр труднее реализовывать и отображать. В наших обсуждениях мы используем центрирование, чтобы обеспечить более удобную визуализацию — ответственный момент в формировании ясного понимания концепций фильтрации. На практике возможна реализация и других способов выстраивания данных при условии, что соблюдается единообразие.

Перейдем к детальному рассмотрению процесса фильтрации. Одним из простейших фильтров, которые можно построить, является фильтр $H(u,v)$, равный 0 в центре, когда $u = v = 0$, и 1 во всех остальных точках. Такой фильтр при выполнении произведения $H(u,v)F(u,v)$ будет задерживать член, несущий информацию о постоянной составляющей, и будет «пропускать» (т. е. оставлять неизменными) все остальные составляющие $F(u,v)$. Как известно из (4.6-21), постоянная составляющая определяет среднюю яркость изображения, так что приравнивая ее к нулю, тем самым средняя яркость выходного изображения уменьшается до нуля. На рис. 4.30 показан результат такой операции, выполненный при использовании выражения (4.7-1). Как и ожидалось, изображение стало много темнее. (Равенство нулю среднего значения предполагает существование отрицательных величин. Но поскольку при отображении все отрицательные значения были «срезаны», т. е. обнулены, то изображение на рис. 4.30 иллюстрирует лишь принцип, но не является точным отображением полученного результата.)

Как отмечалось ранее, низкие частоты в преобразовании связаны с областями медленно меняющейся яркости, такими как стены комнаты или безоблачное небо на изображении пейзажа. Высокие частоты, наоборот, вызываются резкими переходами яркости, такими как контуры и шум. Таким образом, можно ожидать, что фильтр $H(u,v)$, подавляющий высокие частоты и пропускающий низкие и соответственно называемый *фильтром низких частот*, будет размывать изображение. В это же время фильтр с противоположными характеристиками (называемый *фильтром высоких частот*) будет повышать контраст резких деталей, но вызовет снижение общего контраста на изображении. Эти эффекты иллюстрируются на рис. 4.31. Отметим сходство изображений на рис. 4.31(*д*) и рис. 4.30. Причина заключается в том, что показанный фильтр высоких частот удаляет постоянную составляющую, давая в результате тот же самый основной эффект, что и на рис. 4.30. Добавление к значениям фильтра небольшой постоянной составляющей не приводит к заметному изменению резкости, но предотвращает полное удаление члена, отвечающего за постоянную составляющую, благодаря чему в определенной степени сохраняет тональности изображения, как это видно на рис. 4.31(*е*).

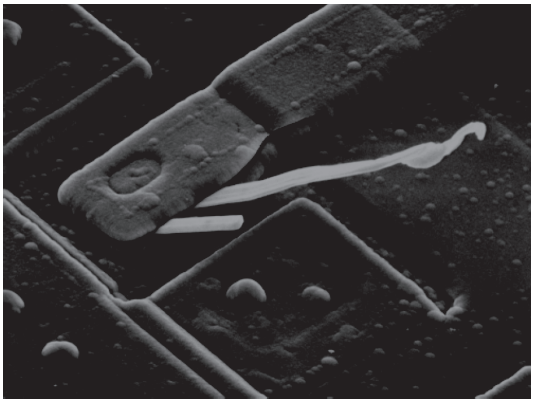


Рис. 4.30. Результат фильтрации изображения на рис. 4.29(*а*) обнулением члена $F(M/2,N/2)$ Фурье-преобразования

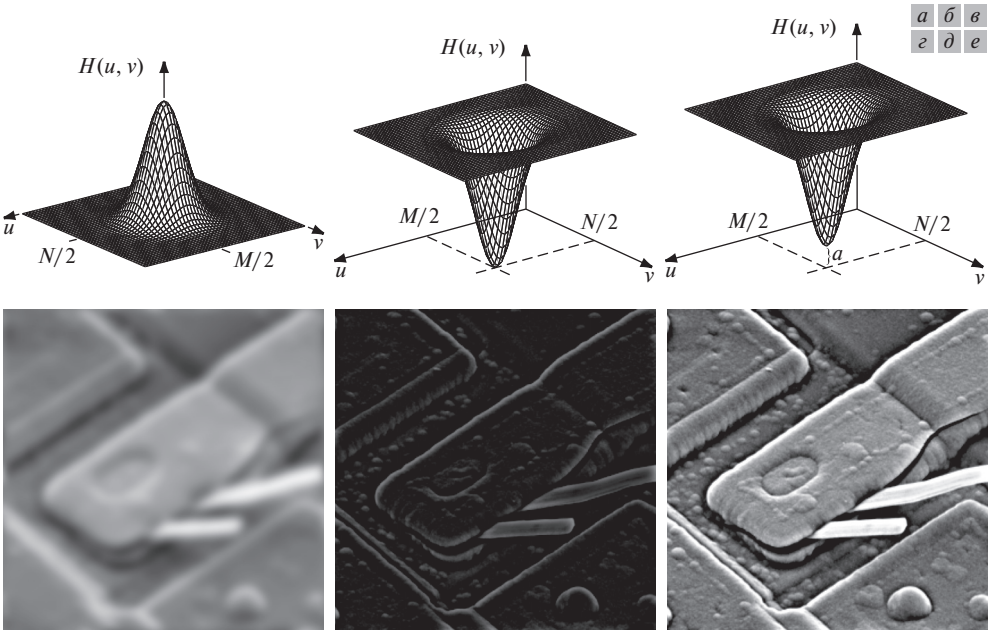


Рис. 4.31. Верхний ряд: спектры частотных фильтров. Нижний ряд: изображения после обработки соответствующим фильтром согласно (4.7-1). Высота всех фильтров равна 1. В фильтре (*в*) при формировании изображения (*е*) использовалось значение $a = 0{,}85$. Сравните (*е*) и рис. 4.29(*а*)

Согласно выражению (4.7-1), осуществляется перемножение двух функций в частотной области, которое, согласно теореме о свертке, означает свертку в пространственной области. Из обсуждения в разделе 4.6.6 известно, что если функции не расширены, то можно ожидать возникновения ошибок перехлеста. Рассмотрим на примере, что происходит при фильтрации изображения согласно (4.7-1) без расширения. На рис. 4.32(*а*) представлено простое изображение, а на рис. 4.32(*б*) — результат его низкочастотной фильтрации га-

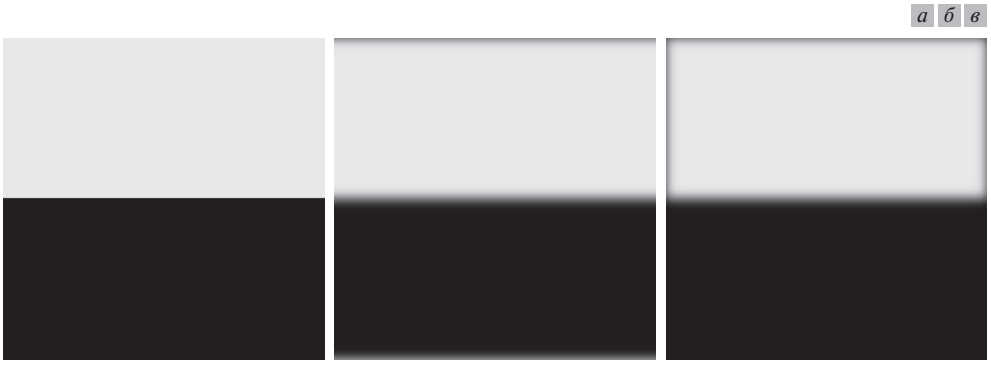


Рис. 4.32. (*а*) Простое изображение. (*б*) Результат размывания гауссовым низкочастотным фильтром без расширения. (*в*) Результат размывания гауссовым низкочастотным фильтром с предварительным расширением. Сравните вертикальные контуры светлых областей на (*б*) и (*в*)

а б

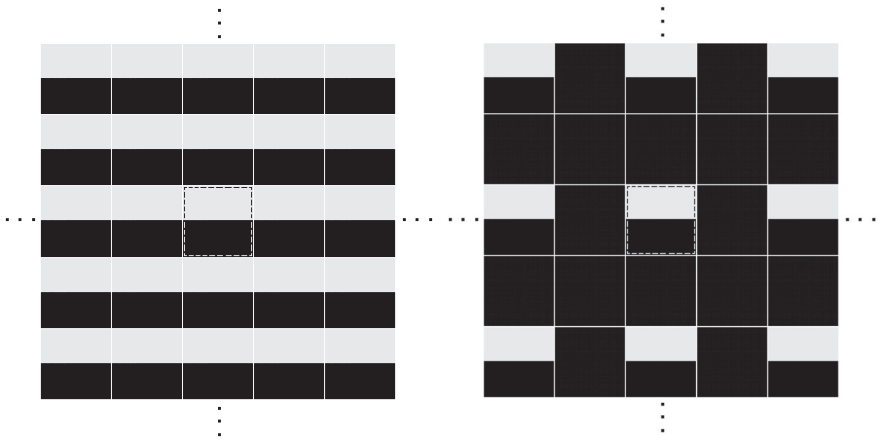


Рис. 4.33. Двумерная периодичность, присущая ДПФ. (*а*) Периодичность без расширения изображения. (*б*) Периодичность после расширения нулями (черные области). Обведенные пунктиром области в центре соответствуют изображению на рис. 4.32(*а*). (Тонкие белые линии на обоих изображениях нанесены для наглядности; они не являются частью данных)

уссовым низкочастотным фильтром, форма которого показана на рис. 4.31(*а*). Как и следовало ожидать, изображение размыто. Однако это размывание не одинаково; верхний светлый (и нижний темный) контуры размыты, тогда как боковые контуры — нет. Расширение исходного изображения нулями в соответствии с выражениями (4.6-31) и (4.6-32) до применения фильтрации (4.7-1) приводит к результату, показанному на рис. 4.32(*в*). Результат таков, как и следовало ожидать.

На рис. 4.33 объясняется причина различий между изображениями на рис. 4.32(*б*) и (*в*). Область, обведенная пунктиром на рис. 4.33, соответствует изображению на рис. 4.32(*а*). Оставшаяся область на рис. 4.33(*а*) демонстрирует эффект периодичности, присущий использованию ДПФ, как объяснялось в разделе 4.6.3. Представим свертку этого изображения *пространственным* аналогом размывающего фильтра. Когда фильтр проходит через верхний край обведенного пунктиром прямоугольника, он будет захватывать как часть верха изображения, так и часть его низа, который вследствие периодичности располагается непосредственно сверху. Когда под окном фильтра оказывается граница темной и светлой областей, результатом будет размытый серый тон. При пересечении верхней половины правой вертикальной границы картина другая: фильтр будет захватывать только светлые области — самого изображения и его соседа справа. Среднее значение константы равно самой константе, поэтому эффекта фильтрации не будет, что и видно на рис. 4.32(*б*). Расширение изображения нулями создает однородный бордюр вокруг исходного изображения[17], как это видно на рис. 4.33(*б*). Свертка размывающей функции с расширенной «мозаикой» на рис. 4.33(*б*) дает правильный результат, пока-

[17] А также увеличивает значения вертикального и горизонтального периодов. — *Прим. перев.*

занный на рис. 4.32(*в*). Из данного примера видно, что отсутствие расширения изображения может привести к ошибочным результатам. Зачастую, когда целью фильтрации является лишь грубый визуальный анализ, расширение не осуществляется.

До настоящего момента обсуждение касалось расширения исходного изображения, но выражение (4.7-1) распространяется также и на фильтр, который может задаваться либо в пространственной, либо в частотной области. Расширение, однако, выполняется в пространственной области, что вызывает важный вопрос о взаимосвязи между пространственным расширением и фильтрами, задаваемыми непосредственно в частотной области.

На первый взгляд, для выполнения расширения в частотной области можно предложить следующий вариант: создать фильтр того же размера, что и изображение, выполнить обратное ДПФ фильтра, чтобы получить соответствующий пространственный фильтр, расширить его в пространствен-

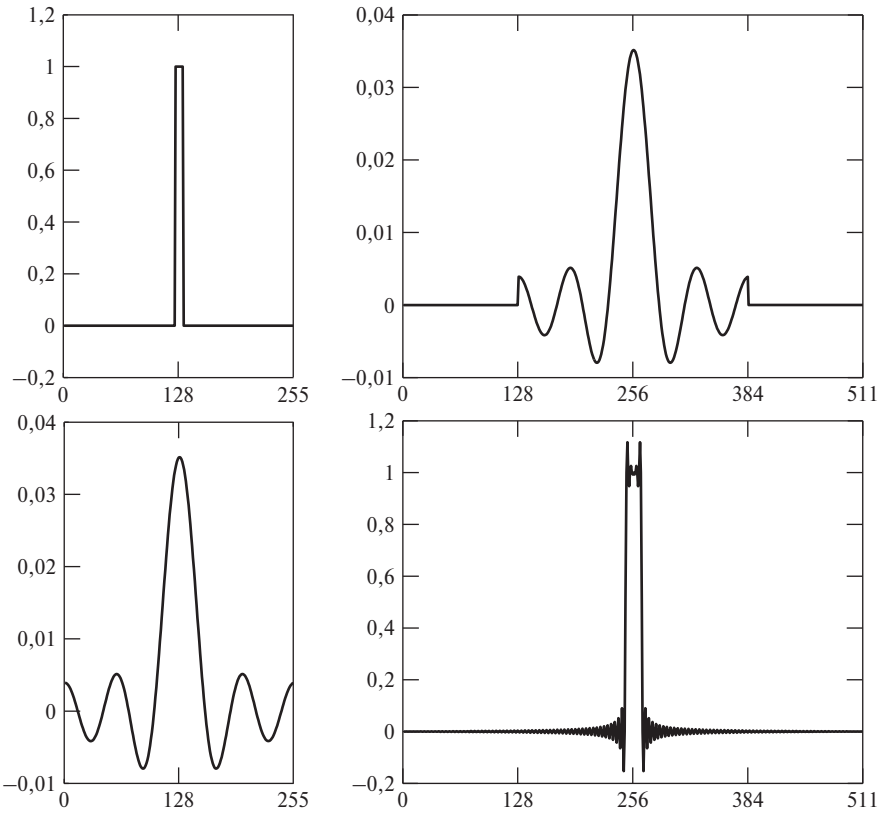


Рис. 4.34. (*а*) Исходный частотный фильтр (центрирован). (*б*) Пространственное представление, полученное вычислением обратного ДПФ. (*в*) Результат расширения нулями (*б*) до удвоенной длины (заметны разрывы). (*г*) Соответствующий фильтр в частотной области, полученный вычислением прямого ДПФ от (*в*). Обратите внимание на «звон», вызванный разрывами в (*в*). (Кривые выглядят непрерывными, поскольку точки были соединены для упрощения визуального анализа)

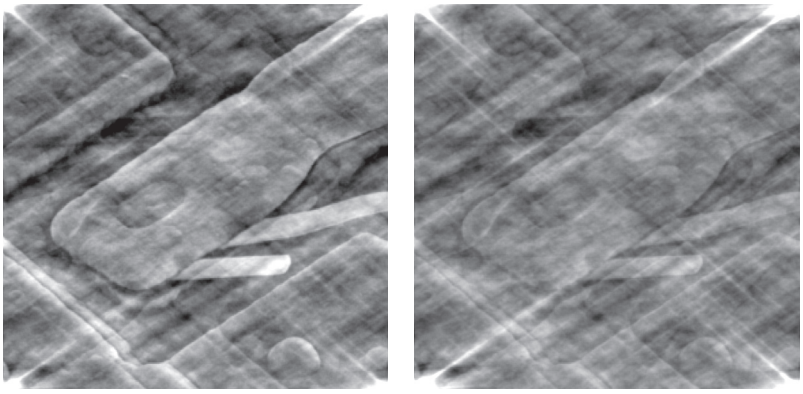
ной области, после чего выполнить прямое ДПФ для получения частотного фильтра. Одномерный пример на рис. 4.34 иллюстрирует подводные камни такого подхода. На рис. 4.34(*а*) изображен одномерный идеальный частотный фильтр. Фильтр является действительным и имеет четную симметрию, так что из свойства 8 в табл. 4.1 известно, что его обратное ДПФ также будет действительным и симметричным. На рис. 4.34(*б*) показан соответствующий пространственный фильтр, полученный умножением элементов исходного частотного фильтра на $(-1)^u$ и вычислением обратного ДПФ. Значения фильтра на краях не равны нулю, поэтому, как видно на рис. 4.34(*в*), расширение функции фильтра нулями приводит к появлению двух разрывов (поскольку функция периодична, то расширение функции на двух концах эквивалентно расширению только на одном конце при условии, что число нулевых элементов одинаково).

Чтобы вернуться в частотную область, выполняется прямое ДПФ пространственного расширенного фильтра. Результат представлен на рис. 4.34(*г*). Разрывы в пространственной области вызывают появление «звона»[18] в частотном эквиваленте, аналогично тому, как это было в примере 4.1. Из того примера известно, что Фурье-преобразование прямоугольной функции является функцией sinc с частотными компонентами, продолжающимися в бесконечность; следует ожидать аналогичного поведения и от обратного преобразования прямоугольной функции. То есть пространственное представление идеального (прямоугольного) частотного фильтра продолжается в бесконечность. Таким образом, любое пространственное усечение фильтра для расширения его нулями вообще говоря приведет к разрывам, которые затем вызовут появление звона в частотной области (разрывов можно избежать, если делать усечение в точке перехода функции через ноль; но нас интересует общий случай, тем более что не все функции имеют переход через ноль).

См. окончание раздела 4.3.3 касательно определения идеального фильтра.

Последние результаты показывают, что поскольку отсутствует возможность работать с бесконечным набором компонент, то невозможно одновременно использовать идеальный частотный фильтр (как на рис. 4.34(*а*)) и расширение нулями для устранения ошибок перехлеста. Требуется решение, которое из ограничений принять. Наша цель — работать с частотными фильтрами определенной формы (включая идеальные фильтры), не очень беспокоясь по поводу возможных эффектов урезания. Одним из вариантов является расширение нулями изображения и затем создание частотных фильтров такого же размера, что и полученное расширенное изображение (при использовании ДПФ, как мы помним, изображения и фильтры должны быть одного размера). Конечно, это будет приводить к ошибкам перехлеста, поскольку для фильтра не используется расширение, но на практике эти ошибки значительно смягчаются раздвижкой сигналов копий изображений, происходящей благодаря

[18] Термин «звон» происходит из электро- и радиотехники, где им обозначают весьма близкое явление. Рассматриваемое явление соответствует тому, что в математике называется «явлением Гиббса». Последний термин также используется в обработке изображений. — *Прим. перев.*



а *б*

Рис. 4.35. (*а*) Изображение получено умножением значений фазовых углов в (4.6-15) на 0,5 и последующим вычислением обратного ДПФ. (*б*) Результат умножения массива фаз на 0,25. В обоих случаях значения спектра не менялись

расширению, и это предпочтительно по сравнению с возможным возникновением «звона». Сглаживающие фильтры (как на рис. 4.31) приносят даже еще меньше проблем. В настоящей главе, чтобы работать с фильтрами заданной формы непосредственно в частотной области, мы будем придерживаться подхода, при котором изображения расширяются до размера $P \times Q$ и фильтры создаются того же размера. Как объяснялось ранее, P и Q задаются выражениями (4.6-29) и (4.6-30).

Мы завершаем раздел рассмотрением фазы фильтрованного преобразования Фурье. Поскольку ДПФ является комплексным массивом, его можно выразить в виде суммы действительной и мнимой частей:

$$F(u,v) = R(u,v) + iI(u,v). \tag{4.7-2}$$

Тогда выражение (4.7-1) примет вид

$$g(x,y) = \Im^{-1}[H(u,v)R(u,v) + iH(u,v)I(u,v)]. \tag{4.7-3}$$

Описанный способ фильтрации не меняет фазы, поскольку $H(u,v)$ сокращается в выражении угла фазы, как отношения мнимой и действительной частей сигнала (4.6-17). Такие фильтры, которые равно воздействуют на действительную и мнимую части сигнала и тем самым не изменяют фазу, соответственно называются фильтрами *с нулевым сдвигом фазы*. В настоящей главе рассматриваются лишь такие фильтры.

Даже малые изменения фазы могут привести к значительным (обычно нежелательным) изменениям результата фильтрации. Эффект простого скалярного изменения фазы проиллюстрирован на рис. 4.35. На рис. 4.35(*а*) показан результат умножения массива фазовых углов в (4.6-15) на 0,5 без изменения значений амплитуд $|F(u,v)|$ и последующего вычисления обратного ДПФ[19]. Основные контуры остались на месте, но распределение яркостей совершенно ис-

[19] Очевидно, исходными данными для применения указанного преобразования служит ДПФ фрагмента изображения на рис. 4.29(*а*). — *Прим. перев.*

кажено. На рис. 4.35(*а*) представлен результат умножения массива фаз на 0,25. Изображение стало почти неузнаваемым[20].

4.7.3. Последовательность шагов частотной фильтрации

Материал двух предшествующих разделов может быть кратко суммирован следующим образом:

1. Для заданного изображения $f(x,y)$ размерами $M\times N$ при помощи выражений (4.6-31) и (4.6-32) вычислить значения параметров расширения P и Q. Как правило, мы выбираем $P = 2M$ и $Q = 2N$.
2. Сформировать расширенное изображение $f_p(x,y)$ размерами $P\times Q$ добавлением к $f(x,y)$ необходимого числа нулей.
3. Умножить $f_p(x,y)$ на $(-1)^{x+y}$ чтобы центрировать его Фурье-образ.

> Как отмечалось ранее, центрирование упрощает построение фильтр-функций и улучшает визуализацию процесса фильтрации, но не является существенным требованием.

4. Вычислить $F(u,v)$ — прямое ДПФ изображения, полученного на шаге 3.
5. Сформировать действительную симметричную фильтр-функцию $H(u,v)$ размерами $P\times Q$ с центром в точке $(P/2, Q/2)$.[21] Вычислить произведение $G(u,v) = H(u,v)F(u,v)$, используя поэлементное произведение массивов, т. е. $G(i, k) = H(i, k)F(i, k)$.
6. Получить обработанное изображение при помощи вычисления обратного ДПФ

$$g_p(x,y) = \left\{\text{real}\left[\Im^{-1}\left[G(u,v)\right]\right]\right\}(-1)^{x+y},$$

   в котором, чтобы убрать паразитную мнимую компоненту, возникающую из-за неточностей вычислений берется лишь действительная часть; нижний индекс p означает, что это расширенный массив.
7. Получить окончательный результат обработки $g(x,y)$ вырезанием области размерами $M\times N$ из левого верхнего угла $g_p(x,y)$.

Указанные шаги проиллюстрированы на рис. 4.36. В описании каждого изображения указано содержание каждого действия. Если увеличить изображение рис. 4.36(*в*), то будут видны черные точки, перемежающие пиксели изображения, — следствие того, что отрицательные яркости срезаны и заменены значением 0. Отметим характерную темную границу на рис. 4.36(*з*), возникшую вследствие низкочастотной фильтрации расширенного нулями изображения.

---

[20] Вопросы изменения фазы стоило бы дополнить тем, что *прибавление* константы к значениям степени в полярном представлении (4.6 15) элементов Фурье-образа приводит к простому сдвигу всего изображения без дополнительных искажений, что, по сути, перекликается с вопросами, рассматривавшимися в разделе 4.6.2. — *Прим. перев.*

[21] Если $H(u, v)$ формируется из *заданного* пространственного фильтра $h(x, y)$, то при формировании $h_p(x, y)$ расширением фильтра до размеров $P\times Q$ следует умножить расширенный массив на $(-1)^{x+y}$ и вычислить ДПФ полученного результата, чтобы получить центрированный массив $H(u, v)$. Эту процедуру иллюстрирует пример 4.15.

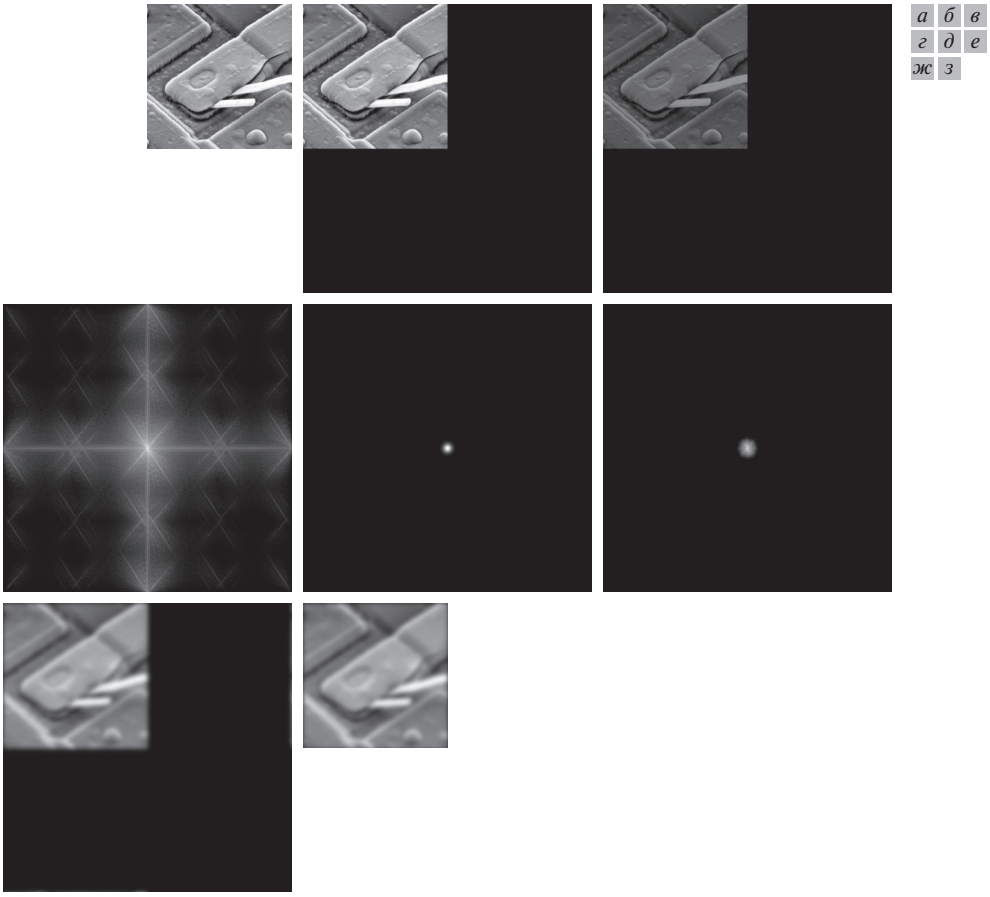


Рис. 4.36. (*а*) Изображение f размерами $M \times N$. (*б*) Расширенное изображение f_p размерами $P \times Q$. (*в*) Результат умножения f_p на $(-1)^{x+y}$. (*г*) Спектр F_p. (*д*) Центрированный гауссов низкочастотный фильтр H размерами $P \times Q$. (*е*) Спектр произведения HF_p. (*ж*) g_p, произведение $(-1)^{x+y}$ и действительной части обратного ДПФ от HF_p. (*з*) Окончательный результат g, полученный вырезанием первых M строк и N столбцов из g_p

4.7.4. Соответствие между пространственными и частотными фильтрами

Связь между фильтрами пространственной и частотной областей определяется теоремой о свертке. Фильтрация в частотной области была определена в разделе 4.7.2 как произведение фильтра-функции $H(u,v)$ на преобразование Фурье исходного изображения $F(u,v)$. Пусть задан фильтр $H(u,v)$; предположим, мы хотим определить его эквивалентное представление в пространственной области. Для этого возьмем единичный импульс $f(x,y) = \delta(x,y)$; из табл. 4.3 следует, что $F(u,v) = 1$. Тогда согласно (4.7-1) выход фильтра будет равен $\Im^{-1}\{H(u,v)\}$ — обратному преобразованию частотного фильтра, соответствующего пространственному фильтру. Из аналогичного анализа и теоремы о свертке следует и обратное, что если задан пространственный фильтр, его

представление в частотной области получается выполнением прямого Фурье-преобразования пространственного фильтра. Таким образом два фильтра образуют Фурье-пару:

$$h(x,y) \Leftrightarrow H(u,v), \tag{4.7-4}$$

где $h(x,y)$ — пространственный фильтр. Поскольку этот фильтр может быть получен как отклик частотного фильтра на единичный импульс, то часто $h(x,y)$ называют *импульсной характеристикой* (*импульсным откликом*) *фильтра* $H(u,v)$. Также, поскольку при дискретной реализации уравнения (4.7-4) как пространственные (временные) параметры, так и значения коэффициентов являются ограниченными, то такие фильтры называют фильтрами с *конечными импульсными характеристиками* (КИХ-фильтрами). В данной книге рассматриваются линейные пространственные фильтры исключительно такого типа.

Пространственная свертка была введена в разделе 3.4.1, и ее реализация рассматривалась в связи с уравнением (3.4-2), которое относится к функциям свертки различных размеров. Когда мы говорим о пространственной свертке в терминах теоремы о свертке и ДПФ, то подразумевается, что в операции участвуют периодические функции, как это объяснялось в комментариях к рис. 4.28. По этой причине, как уже говорилось ранее, уравнение (4.6-23) называют *круговой сверткой*. Более того, свертка в контексте ДПФ предполагает наличие одинаковых по размерам функций, тогда как в (3.4-2) функции могут быть и совершенно различных размеров.

На практике часто требуется осуществлять фильтрацию сверткой (согласно уравнению (3.4-2)) с маской небольшого размера, что может выполняться с достаточно высокой скоростью и не слишком трудно в аппаратной реализации. Однако концепции фильтрации в частотной области интуитивно более понятны. Один из способов воспользоваться преимущественными свойствами каждой из областей состоит в том, чтобы задать фильтр в частотной области, вычислить его обратное ДПФ и полученный полноразмерный пространственный фильтр использовать в качестве *образца* для построения масок пространственных фильтров меньшего размера (более формальные подходы рассмотрены в разделе 4.11.4.). Это разбирается уже в следующих параграфах. Далее будет рассматриваться и обратный случай, когда задан небольшой пространственный фильтр и формируется его полноразмерное представление в частотной области. Такой подход удобен для частотного анализа поведения пространственных фильтров. В последующих обсуждениях необходимо помнить, что прямое и обратное преобразования Фурье являются линейными процессами (задача 4.14), так что рассмотрение ограничено линейной фильтрацией.

В качестве примера того, как частотные фильтры могут использоваться в качестве эталона для задания коэффициентов некоторых из набольших масок, рассматривавшихся в главе 3, воспользуемся гауссовым фильтром. Фильтры, основанные на гауссовых функциях, представляют особый интерес, поскольку, как указано в табл. 4.3, и прямое, и обратное Фурье-преобразования гауссовой функции являются действительными гауссовыми функциями. Мы ограничимся рассмотрением одномерного случая для иллюстрации лежащих в основе фильтрации принципов. Двумерные гауссовы фильтры будут рассмотрены в данной главе позже.

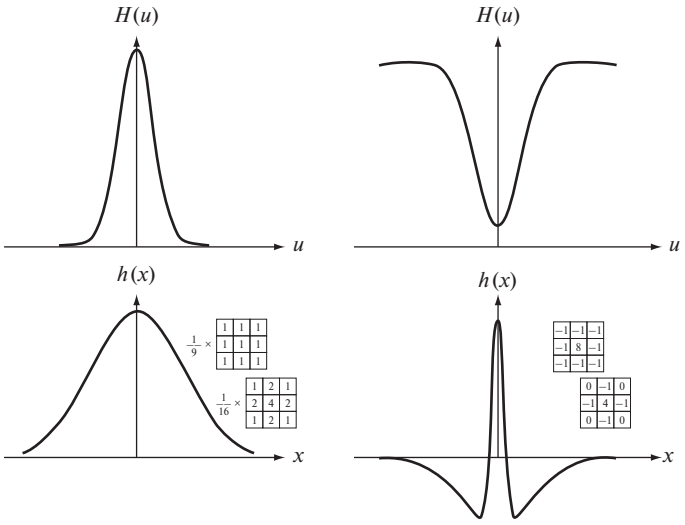


Рис. 4.37. (*а*) Одномерный гауссов низкочастотный фильтр в частотной области. (*б*) Пространственный низкочастотный фильтр, соответствующий (*а*). (*в*) Гауссов высокочастотный фильтр в частотной области. (*г*) Пространственный высокочастотный фильтр, соответствующий (*в*). Представленные небольшие двумерные маски использовались в главе 3

Обозначим $H(u)$ одномерный частотный гауссов фильтр:

$$H(u) = Ae^{-u^2/2\sigma^2}, \tag{4.7-5}$$

где σ — стандартное отклонение гауссовой функции. Для получения соответствующего пространственного фильтра нужно взять обратное преобразование Фурье от $H(u)$ (задача 4.31):

$$h(x) = \sqrt{2\pi}\sigma A e^{-2\pi^2\sigma^2x^2}. \tag{4.7-6}$$

Данные выражения[22] важны по двум причинам. (1) Они образуют Фурье-пару, причем каждая функция является гауссовой и *действительной*. Это облегчает анализ, поскольку нет необходимости заниматься комплексными числами. Кроме того, гауссовы кривые наглядны и удобны для обработки. (2) Функции ведут себя взаимно противоположно: если $H(u)$ широкая (значение σ большое), то $h(x)$ узкая и наоборот. Фактически, если σ устремить в бесконечность, то $H(u)$ будет стремиться к константе, а $h(x)$ — к импульсу, что означает отсутствие фильтрации в частотной и пространственной областях соответственно.

На рис. 4.37(*а*) представлен график гауссового низкочастотного фильтра в частотной области, а на (*б*) — соответствующего пространственного фильтра. Предположим, мы хотим использовать форму $h(x)$ на рис. 4.37(*б*) в качестве

[22] Как указано в табл. 4.3, аналитические выражения для прямого и обратного преобразования Фурье гауссовых функций применимы лишь для непрерывных функций. Для использования дискретных форм следует просто дискретизовать непрерывные преобразования гауссовой функции. Применение нами дискретных переменных означает, что работа происходит с дискретизованными преобразованиями.

ориентира для задания коэффициентов небольшой пространственной маски. Важное сходство этих двух фильтров состоит в том, что все значения у них положительны. Таким образом, можно обеспечить низкочастотную фильтрацию в пространственной области при помощи маски, содержащей только положительные коэффициенты, как это было сделано в разделе 3.5.1; для сравнения на рис. 4.37(*б*) представлены две из рассматривавшихся там масок. Обратим внимание на взаимосвязь ширины соответствующих фильтров, про которую говорилось в предыдущем абзаце: чем более узок фильтр в частотной области, тем сильнее он подавляет высокие частоты, приводя к более сильному размыванию. В пространственной области это означает, что для увеличения размывания должна быть использована маска большего размера, как это иллюстрировалось в примере 3.13.

На базе простой гауссовой функции, заданной выражением (4.7-5), можно создавать более сложные фильтры. Например, можно построить высокочастотный фильтр в виде *разности* двух гауссианов

$$H(u) = Ae^{-u^2/2\sigma_1^2} - Be^{-u^2/2\sigma_2^2} \tag{4.7-7}$$

с $A \geq B$ и $\sigma_1 > \sigma_2$. Соответствующий пространственный фильтр будет выглядеть как

$$h(x) = \sqrt{2\pi}\sigma_1 Ae^{-2\pi^2\sigma_1^2 x^2} - \sqrt{2\pi}\sigma_2 Be^{-2\pi^2\sigma_2^2 x^2}. \tag{4.7-8}$$

На рис. 4.37(*в*) и (*г*) показаны графики этих двух выражений. Еще раз отмечаем взаимосвязь функций по ширине, но наиболее важной особенностью здесь является то, что в маске $h(x)$ центральный элемент имеет положительное значение, а краевые элементы — отрицательные. Ту же особенность передают и небольшие маски, показанные на рис. 4.37(*г*). Эти две маски использовались в главе 3 в качестве фильтров повышения резкости, которые, как мы теперь знаем, являются высокочастотными фильтрами.

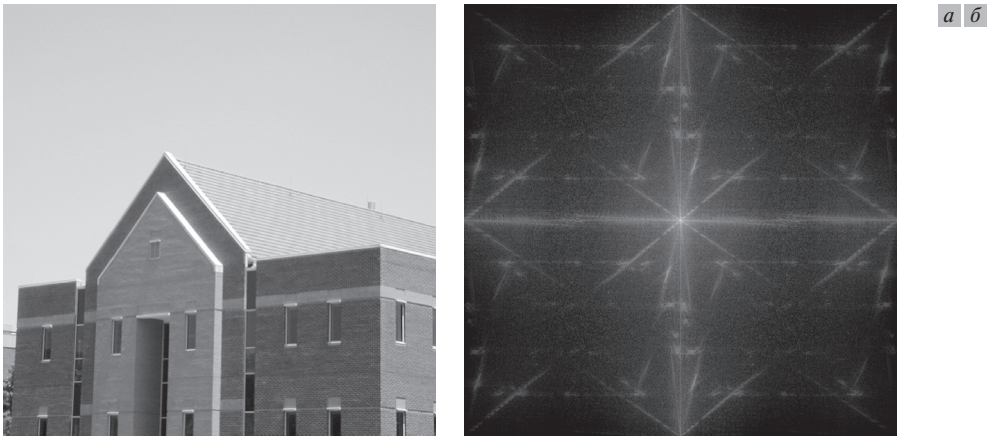
Изучение вышеизложенного материала требует значительных усилий, но не подлежит сомнению, что правильно понять фильтрацию в частотной области невозможно без тех основ, которые были сформулированы. На практике частотная область может рассматриваться как некая «лаборатория», в которой более удобно изучать соотношение между видом изображения и его частотным содержанием. Как многократно демонстрируется ниже в данной главе, некоторые задачи, которые исключительно трудно или даже невозможно сформулировать непосредственно в пространственной области, становятся почти тривиальными в частотной области. Когда при помощи экспериментов в частотной области удается выбрать конкретный фильтр, действительная реализация метода обычно осуществляется в пространственной области. Одним из принципов является формирование небольших пространственных масок, способных вобрать в себя «основное содержание» полной фильтр-функции в частотной области, как это объяснялось в контексте рис. 4.37. Более формальный подход заключается в создании двумерного цифрового фильтра использованием приближений, базирующихся на математических или статистических критериях. Мы еще раз коснемся этого вопроса в разделе 4.11.4.

Пример 4.15. Получение частотного фильтра из небольшой пространственной маски.

■ В данном примере мы начнем с пространственной маски и покажем, как создать соответствующий ей частотный фильтр. Затем мы сравним результаты фильтрации, полученные при помощи частотных и пространственных методов. Такой анализ полезен, когда возникает желание сравнить качества некоторой конкретной пространственной маски с одним или более «полноразмерными» фильтрами-кандидатами в частотной области или глубже понять действие выбранной маски. Для упрощения материала выберем детектор Собела вертикальных контуров, имеющий маску 3×3 и показанный на рис. 3.41(*д*). На рис. 4.38(*а*) показано исходное изображение размерами 600×600, которое мы собираемся фильтровать, а на рис. 4.38(*б*) — его спектр.

На рис. 4.39(*а*) показана маска Собела $h(x, y)$ и трехмерный график его частотного представления (объясняется ниже). Поскольку исходное изображение имеет размеры 600×600 пикселей, а фильтр — 3×3 точек, то для устранения эффектов перехлеста мы расширим f и h до размеров 602×602 пикселей в соответствии с выражениями (4.6-29) и (4.6-30). Маска Собела имеет нечетную симметрию при условии, что она вставлена в массив нулей четного размера (см. пример 4.10). Чтобы соблюсти эту симметрию, разместим $h(x, y)$ так, чтобы его центр совпадал с центром расширенного массива размерами 602×602 пикселя; это является важным аспектом построения фильтра. Если при формировании $h_p(x, y)$ сохраняется нечетная симметрия расширенного массива, то согласно свойству 9 из табл. 4.1 $H(u,v)$ будет чисто мнимой. Как будет показано в конце примера, это приведет к результатам, идентичным использованию $h(x, y)$ в пространственной области. Если симметрия не выполняется, то результаты уже не будут теми же самыми.

Процедура получения $H(u,v)$ состоит из следующих шагов: (1) умножить $h_p(x, y)$ на $(-1)^{x+y}$, чтобы центрировать частотный фильтр; (2) вычислить прямое ДПФ результата шага (1); (3) обнулить действительную часть результата ДПФ, в которой могут содержаться паразитные значения (как известно, $H(u,v)$ должна быть чисто мнимой); и (4) умножить результат на $(-1)^{u+v}$. Последний

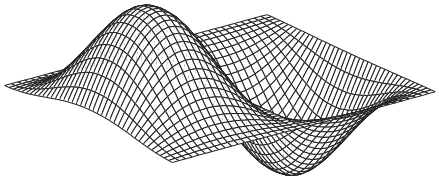


а *б*

Рис. 4.38. (*а*) Изображение здания и (*б*) его спектр

а б
в г

–1	0	1
–2	0	2
–1	0	1



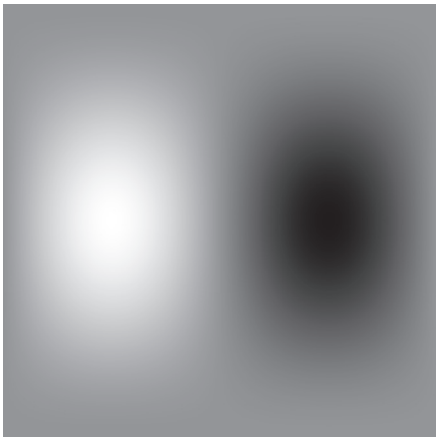


Рис. 4.39. (*а*) Пространственная маска и трехмерный график ее частотного представления. (*б*) Фильтр, показанный в виде изображения. (*в*) Результат фильтрации изображения на рис. 4.38(*а*) частотным фильтром (*б*). (*г*) Результат фильтрации изображения на рис. 4.38(*а*) пространственным фильтром (*а*). Результаты идентичны

шаг компенсирует скрытое умножение $H(u,v)$ на $(-1)^{u+v}$, которое по сути происходит при сдвиге $h(x, y)$ в центр $h_p(x, y)$. Функция $H(u,v)$ показана на рис. 4.39(*а*) в виде трехмерного графика, а на рис. 4.39(*б*) — в виде полутонового изображения, в котором ноль отображается серым уровнем. Функция является нечетной, т. е. антисимметричной относительно центра, что хорошо видно на рисунках. Как и любой другой частотный фильтр, $H(u,v)$ используется согласно процедуре, очерченной в разделе 4.7.3.

На рис. 4.39(*в*) представлен результат преобразования изображения на рис. 4.38(*а*) только что рассмотренным фильтром согласно процедуре, изложенной в разделе 4.7.3. Как и следовало ожидать от дифференцирующего фильтра, улучшены контуры, ориентированные близко к вертикали, а участки всех остальных интенсивностей, включая постоянные области и горизонтальные контуры, приближены к нулю (для удобства визуализации нулевые значения сдвинуты в область серого). На рис. 4.39(*г*) показан результат фильтрации того

же изображения соответствующим фильтром $h(x, y)$ непосредственно в пространственной области согласно процедуре, изложенной в разделе 3.6.4. Результаты идентичны. ■

4.8. Частотные фильтры сглаживания изображения

Остаток данной главы посвящен различным методам частотной фильтрации. Начнем с низкочастотной фильтрации. Значительный вклад в высокие частоты Фурье-преобразования изображения дают контуры и другие резкие яркостные переходы, включая шумы. Следовательно, в частотной области сглаживание (размывание) достигается подавлением высоких частот, т. е. при помощи *низкочастотной* фильтрации. В данном разделе будут рассмотрены три вида низкочастотных фильтров: идеальный фильтр, фильтр Баттерворта и гауссов фильтр. Эти три фильтра покрывают диапазон от очень резких фильтров (идеальный) до очень гладких фильтров (гауссов). Фильтр Баттерворта характеризуется параметром, который называется *порядком* фильтра. При больших значениях этого параметра фильтр Баттерворта приближается по форме к идеальному фильтру. При малых значениях он имеет гладкую форму, похожую на форму гауссова фильтра. Таким образом, фильтр Баттерворта может рассматриваться как переходный между двумя «крайностями». Все методы фильтрации, рассматриваемые в настоящем разделе, следуют процедуре, изложенной в разделе 4.7.3, следовательно предполагается, что все фильтр-функции $H(u,v)$ являются дискретными функциями размеров $P \times Q$, т. е. дискретные частотные переменные находятся в диапазонах $u = 0, 1, 2, ..., P - 1$ и $v = 0, 1, 2, ..., Q - 1$.

4.8.1. Идеальные фильтры низких частот

Двумерный фильтр низких частот, который пропускает без затухания все частотные составляющие внутри круга радиусом D_0 от начала координат и «обрезает» все частотные составляющие вне этого круга, называется *идеальным фильтром низких частот* (идеальный ФНЧ); он определяется следующей функцией:

$$H(u,v) = \begin{cases} 1 & \text{при } D(u,v) \le D_0 \\ 0 & \text{при } D(u,v) > D_0, \end{cases} \tag{4.8-1}$$

где D_0 — заданная положительная константа, а $D(u,v)$ — расстояние от точки (u,v) частотной области до начала координат (центра частотного прямоугольника); т. е.

$$D(u,v) = \left[(u - P/2)^2 + (v - Q/2)^2\right]^{1/2}, \tag{4.8-2}$$

где, как и ранее, P и Q — размеры расширенного изображения согласно выражениям (4.6-31) и (4.6-32). На рис. 4.40(*а*) фильтр $H(u,v)$ представлен в виде трехмерного графика, а на рис. 4.40(*б*) — в виде изображения. Согласно разделу 4.3.3 название фильтра *идеальный* указывает, что все частоты внутри круга радиуса D_0 проходят без изменения, а все частоты вне круга полностью подавляются (отфильтровываются). Идеальный низкочастотный фильтр яв-

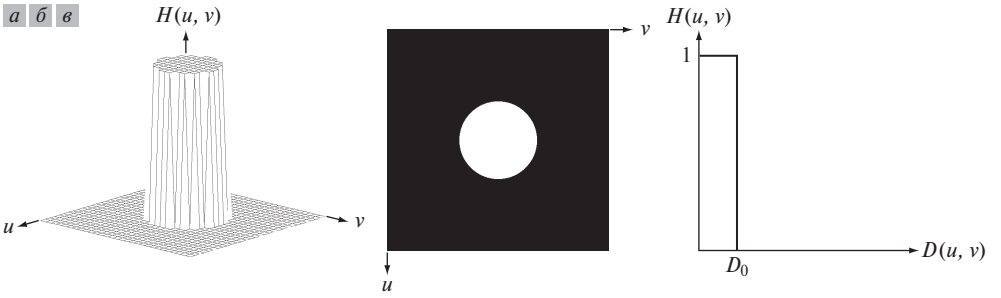


Рис. 4.40. (*а*) Трехмерный график передаточной функции идеального низкочастотного фильтра. (*б*) Представление фильтра в виде изображения. (*в*) Радиальный профиль фильтра

ляется радиально симметричным вокруг начала координат, а значит, он полностью задается своим радиальным сечением (профилем), которое показано на рис. 4.40(*в*). Вращение сечения вокруг начала координат на 360° дает в результате двумерный фильтр.

Для профиля идеального ФНЧ точка, в которой происходит переход от значений $H(u,v) = 1$ к значениям $H(u,v) = 0$, называется *частотой среза*. В случае, показанном на рис. 4.40, например, частота среза равна D_0. Резкое обрезание частот, присущее идеальному ФНЧ, не может быть осуществлено в электронных устройствах, однако может быть реализовано при компьютерных вычислениях. Эффекты, возникающие на цифровом изображении при использовании таких «нефизических» фильтров, обсуждаются ниже в данном разделе.

Низкочастотные фильтры, представленные в этой главе, сравниваются исследованием их поведения как функций одинаковых частот среза. Один из способов ввести эталонный набор положений частот среза состоит в том, чтобы определить круги, в которых заключена заданная часть полной энергии изображения P_T. Полная энергия определена как сумма компонент энергетического спектра расширенного изображения во всех точках (u,v), $u = 0, 1, 2, ..., P - 1$ и $v = 0, 1, 2, ..., Q - 1$, т. е.

$$P_T = \sum_{u=0}^{P-1}\sum_{v=0}^{Q-1} P(u,v), \tag{4.8-3}$$

где значения $P(u,v)$ заданы формулой (4.6.-18). Если ДПФ центрировано, то круг радиусом D_0, расположенный в центре частотного прямоугольника, содержит α процентов энергии спектра, где

$$\alpha = 100\left[\sum_u\sum_v P(u,v)/P_T\right], \tag{4.8-4}$$

причем суммирование в последней формуле идет по значениям (u,v), лежащим внутри круга и на его границе.

На рис. 4.41(*а*) и (*б*) показано модельное тестовое изображение и его спектр. Наложенные на спектр круги имеют радиусы 10, 30, 60, 160 и 460 пикселей. В этих кругах заключено α процентов энергии изображения, $\alpha = 87{,}0$; 93,1; 95,7; 97,8 и 99,2 % соответственно. Спектр убывает весьма быстро; 87 % полной энергии заключено в относительно малом круге радиуса 10.

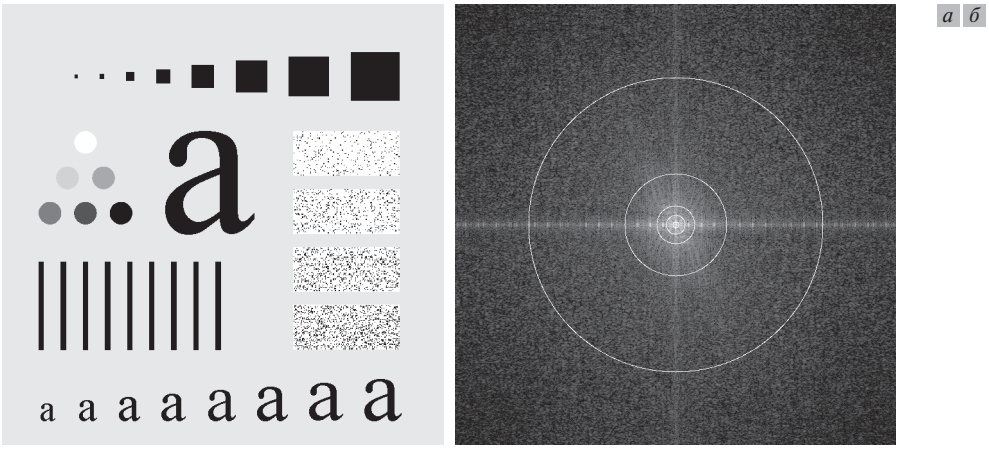


Рис. 4.41. (*а*) Тестовое изображение размерами 688×688 пикселей и (*б*) его Фурье-спектр. Спектр вдвое больше размеров изображения из-за расширения, но показан в половинном размере, чтобы уместиться на странице. Наложенные на полноразмерный спектр изображения круги имеют радиусы 10, 30, 60, 160 и 460. Эти круги заключают в себе соответственно 87,0; 93,1; 95,7; 97,8 и 99,2 % энергии расширенного изображения

Пример 4.16. Сглаживание изображения при помощи идеального ФНЧ.

■ На рис. 4.42 показаны результаты применения идеального ФНЧ с частотами среза, равными значениям радиусов на рис. 4.41(*б*). Результат на рис. 4.42(*б*) практически совершенно бесполезен, если только задача сглаживания не состоит в устранении всех деталей изображения, за исключением пятен, представляющих большие объекты. Очень сильное размывание на этом изображении ясно указывает, что бóльшая часть информации о резких деталях на картинке содержится в 13 % энергии, отсеченной фильтром. По мере увеличения радиуса фильтра все меньшая и меньшая часть энергии подлежит отсечению, что выражается в уменьшении степени размывания. Отметим, что для изображений на рис. 4.42(*в*)—(*д*) характерен «звон», структура которого становится тоньше по мере уменьшения энергии, отсекаемой высокочастотной составляющей. Звон хорошо заметен даже на изображении (рис. 4.42(*д*)), из которого удалено лишь 2 % полной энергии. Обсуждаемое явление, как будет вскоре объяснено, характерно для идеальных фильтров. Наконец, тщательное рассмотрение результата для $\alpha = 99{,}2\,\%$ (рис. 4.42(*е*)) демонстрирует сравнительно небольшое размывание контуров и областей, содержащих шум, при том что в остальном это изображение весьма близко к оригиналу. Это показывает, что в рассматриваемом частном случае 0,8 % энергии верхней части спектра содержат не очень большое количество контурной информации.

Из приведенного примера понятно, что идеальные фильтры низких частот не слишком удобны. Однако полезно изучить их поведение в качестве части изложения концепций фильтрации. Кроме того, как показывает дальнейшее обсуждение, попытки объяснить в пространственной области свойство звона идеального ФНЧ позволяют достичь некоторого дополнительного понимания вопросов фильтрации. ■

а б
в г
д е

Рис. 4.42. (*а*) Исходное изображение. (*б*)—(*е*) Результаты фильтрации идеальными ФНЧ с частотой среза 10, 30, 60, 160 и 460, как показано на рис. 4.41(*б*). Фильтры отсекают соответственно 13; 6,9; 4,3; 2,2 и 0,8 % полной энергии спектра

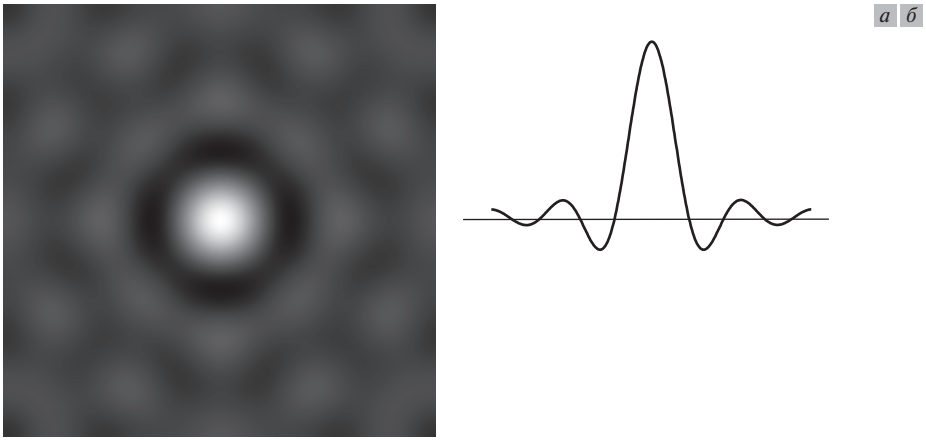


Рис. 4.43. (*а*) Представление идеального ФНЧ с частотой среза 5 и размерами 1000×1000 в пространственной области. (*б*) Профиль яркости вдоль горизонтальной линии, проходящей через центр изображения

Возникающие при использовании идеального ФНЧ эффекты размывания и звона могут быть объяснены при помощи теоремы о свертке. На рис. 4.43(*а*) показано пространственное представление $h(x,y)$ идеального ФНЧ радиуса 5, а на рис. 4.43(*б*) — профиль яркости вдоль строки, проходящей через центр изображения (*а*). Поскольку сечение идеального ФНЧ в частотной области выглядит как прямоугольный фильтр, нет ничего неожиданного в том, что профиль соответствующего пространственного фильтра выглядит как sinc-функция. Фильтрация в пространственной области производится сверткой $h(x,y)$ с изображением. Представим каждый пиксель изображения как дискретный импульс, интенсивность которого пропорциональна яркости изображения в этой точке. Свертка sinc-функции с импульсом копирует sinc в местоположение импульса. Центральный лепесток sinc-функции в основном определяет размывание, а остальные, меньшие лепестки в основном вызывают звон. Свертка sinc с каждым пикселем изображения дает отличную модель для объяснения поведения идеального ФНЧ. Поскольку «рассеяние» sinc-функции обратно пропорционально радиусу $H(u,v)$, то чем больше D_0 (см. рис. 4.40(*в*)), тем больше пространственный sinc приближается к импульсу, который в пределе совсем не вызывает размывания при свертке с изображением. К настоящему моменту такой тип взаимно-обратного поведения уже выглядит совершенно обычным. В последующих двух разделах будет показано, что можно достичь размывания при весьма малом или даже полном отсутствии звона, что является важной целью низкочастотной фильтрации.

4.8.2. Фильтры низких частот Баттерворта

Передаточная функция низкочастотного фильтра Баттерворта (ФНЧ Баттерворта) порядка n с частотой среза на расстоянии D_0 от начала координат задается формулой

$$H(u,v)=\frac{1}{1+\left[D(u,v)/D_0\right]^{2n}}, \qquad (4.8\text{-}5)$$

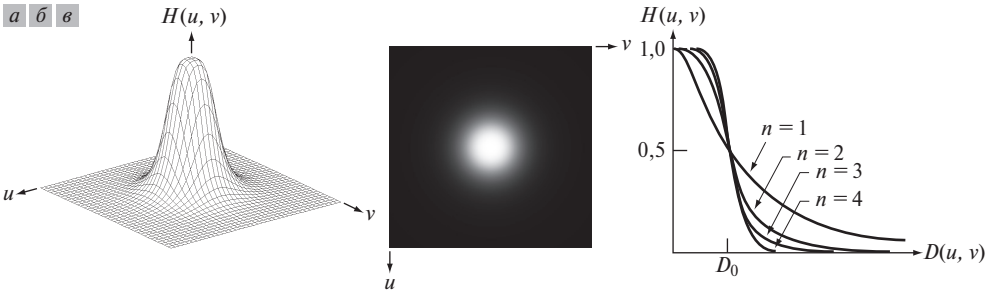


Рис. 4.44. (*а*) Трехмерный график передаточной функции низкочастотного фильтра Баттерворта. (*б*) Полутоновое изображение фильтра. (*в*) Радиальные профили фильтров с порядками от 1 до 4

где расстояние $D(u,v)$ задано формулой (4.8-2). Трехмерный график, полутоновое изображение и радиальные профили передаточной функции ФНЧ Баттерворта представлены на рис. 4.44.

Передаточная функция ФНЧ Баттерворта обычно записывается как квадратный корень из выражения (4.8-5). Однако мы интересуемся здесь в основном формой фильтра и поэтому исключили квадратный корень для удобств вычислений.

В отличие от идеального ФНЧ, передаточная функция ФНЧ Баттерворта не имеет разрыва, который задает точную границу между пропускаемыми и обрезаемыми частотами. Для фильтров с гладкой передаточной функцией обычной практикой является определение местоположения частот среза как множества точек, в которых значения функции $H(u,v)$ становятся меньше определенной доли ее максимального значения. В выражении (4.8-5) такой точкой является $D(u,v) = D_0$, когда $H(u,v)$ становится меньше 50 % максимального значения, равного 1.

Пример 4.17. Сглаживание изображения низкочастотным фильтром Баттерворта.

■ На рис. 4.45 представлены результаты применения к рис. 4.45(*а*) ФНЧ Баттерворта, заданного формулой (4.8-5) с $n = 2$. Значения частоты среза D_0 при этом равны значениям радиусов кругов, показанных на рис. 4.41(*б*). В отличие от представленных на рис. 4.42 результатов, относящихся к случаю идеального ФНЧ, мы видим здесь плавное уменьшение степени размывания при увеличении частоты среза. Более того, ни на одном из обработанных изображений при помощи этого конкретного ФНЧ Баттерворта звон не заметен, что объясняется свойственным фильтру гладким переходом между низкими и высокими частотами. ■

При использовании ФНЧ Баттерворта порядка 1 в пространственной области звона не возникает. Обычно звон является незаметным для фильтров порядка 2, но может стать значительным при использовании фильтров более высокого порядка. Рис. 4.46 дает возможность провести интересное сравнение *пространственных* представлений ФНЧ Баттерворта различных порядков (с частотой среза, равной 5 во всех случаях). На рисунке представлены также про-

Рис. 4.45. (*а*) Исходное изображение. (*б*)—(*е*) Результаты фильтрации с использованием ФНЧ Баттерворта порядка 2 с частотами среза, показанными на рис. 4.41(*б*). Сравните с рис. 4.42

а б в г

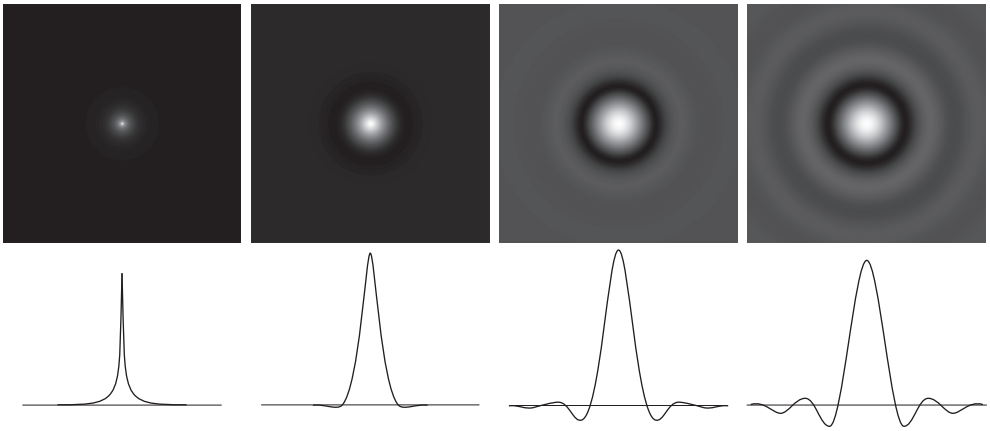


Рис. 4.46. (*а*)—(*г*) Представления ФНЧ Баттерворта порядка 1, 2, 5 и 20 в пространственной области и соответствующие профили яркости, проходящие через центр фильтров (размеры изображений составляют 1000×1000, а частота среза всех фильтров равна 5). Обратите внимание, как возрастает звон с увеличением порядка фильтра

фили яркости соответствующих фильтров вдоль горизонтальной прямой, проходящей через их центр. Для получения этих фильтров была использована та же процедура, что и для фильтра на рис. 4.43(*б*). Для того чтобы облегчить визуальное сравнение, изображения на рис. 4.46 были дополнительно улучшены с помощью гамма-коррекции (см. (3.2-3)). ФНЧ Баттерворта порядка 1 (рис. 4.46(*а*)) не имеет ни концентрических колец, ни отрицательных значений. Фильтр порядка 2 имеет слабые кольца и небольшие отрицательные значения, но они выражены заведомо менее отчетливо, чем в случае идеального ФНЧ. Как показывают оставшиеся изображения, для ФНЧ Баттерворта более высоких порядков звон (кольца и отрицательные значения) становится значительным. Фильтр Баттерворта порядка 20 демонстрирует характеристики, подобные идеальным ФНЧ (в пределе оба фильтра становятся идентичными). Вообще ФНЧ Баттерворта порядка 2 дает пример хорошего компромисса между эффективностью низкочастотной фильтрации и приемлемым уровнем звона.

4.8.3. Гауссовы фильтры низких частот

Для одномерного случая гауссовы фильтры низких частот (ФНЧ Гаусса) были введены в разделе 4.7.4, где они использовались для того, чтобы установить некоторые важные взаимосвязи между пространственной и частотной областями. В двумерном случае эти фильтры задаются формулой

$$H(u,v) = e^{-D^2(u,v)/2\sigma^2}, \tag{4.8-6}$$

где, как и в (4.8-2), $D(u,v)$ — расстояние от центра частотного прямоугольника. В отличие от раздела 4.7.4, мы опускаем константу перед выражением, задающим фильтр, чтобы сохранить единообразие с остальными фильтрами, рассматриваемыми в настоящем разделе, максимальное значение которых равно 1.

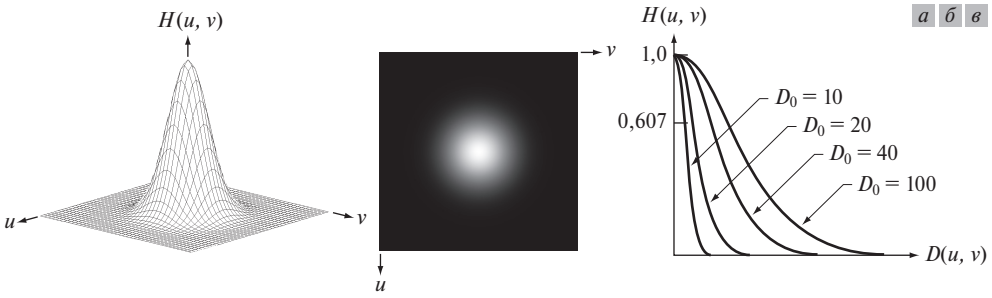


Рис. 4.47. (*а*) Трехмерный график передаточной функции ФНЧ Гаусса. (*б*) Полутоновое изображение фильтра. (*в*) Радиальные профили фильтров для различных значений D_0

Как и раньше, σ задает размах вокруг центра. Обозначив $\sigma = D_0$, мы можем переписать выражение для фильтра, используя форму записи остальных фильтров данного раздела:

$$H(u,v) = e^{-D^2(u,v)/2D_0^2}, \tag{4.8-7}$$

где D_0 — частота среза. Когда $D(u,v) = D_0$, значение передаточной функции ФНЧ Гаусса падает до 0,607 от своего максимального значения.

Как показано в табл. 4.3, обратное Фурье-преобразование от ФНЧ Гаусса также есть гауссова функция. Это означает, что пространственный гауссов фильтр, полученный вычислением обратного ДПФ согласно выражениям (4.8-6) или (4.8-7), не будет создавать звона. Трехмерный график ФНЧ Гаусса, его полутоновое изображение и радиальные профили передаточной функции ФНЧ Гаусса представлены на рис. 4.47; вместе с табл. 4.4 они подводят итог обсуждению низкочастотных фильтров в данном разделе.

Таблица 4.4. Низкочастотные фильтры. D_0 есть частота среза, а n — порядок фильтра Баттерворта

Идеальный	Баттерворта	Гауссов
$H(u,v) = \begin{cases} 1 & \text{при } D(u,v) \le D_0 \\ 0 & \text{при } D(u,v) > D_0 \end{cases}$	$H(u,v) = \dfrac{1}{1+\left[D(u,v)/D_0\right]^{2n}}$	$H(u,v) = e^{-D^2(u,v)/2D_0^2}$

Пример 4.18. Сглаживание изображения низкочастотным гауссовым фильтром.

■ На рис. 4.48 представлены результаты применения к изображению на рис. 4.48(*а*) ФНЧ Гаусса, заданного формулой (4.8-7). Значения частот среза D_0 равнялись значениям радиусов пяти кругов, показанных на рис. 4.41(*б*). Как и в случае ФНЧ Баттерворта порядка 2 (рис. 4.45), мы отмечаем плавное уменьшение степени размывания при увеличении частоты среза. Применение ФНЧ Гаусса по сравнению с ФНЧ Баттерворта порядка 2 дает слегка меньшее сглаживание при одинаковом значении частоты среза, что можно увидеть, например, при сравнении рис. 4.45(*в*) и рис. 4.48(*в*). Этого и следовало ожидать, поскольку профиль ФНЧ Гаусса не такой «сжатый», как профиль ФНЧ Бат-

Рис. 4.48. (*а*) Исходное изображение. (*б*)—(*е*) Результаты фильтрации ФНЧ Гаусса с частотами среза, показанными на рис. 4.41(*б*). Сравните с рис. 4.42 и рис. 4.45

терворта порядка 2. Однако в целом результаты вполне сопоставимы, и, кроме того, в случае ФНЧ Гаусса мы гарантированы от появления звона. Это свойство важно на практике, особенно в тех ситуациях (например при обработке медицинских изображений), когда недопустимы артефакты любого вида. В случаях, когда необходим жесткий контроль зоны перехода от низких к высоким частотам около частоты среза, ФНЧ Баттерворта предоставляет более подходящий выбор. Платой за этот дополнительный контроль над формой фильтра является необходимость считаться с возможностью появления звона. ■

4.8.4. Дополнительные примеры низкочастотной фильтрации

В дальнейших обсуждениях мы приведем несколько примеров практического применения низкочастотной фильтрации в частотной области. Первый пример относится к области машинного восприятия и связан с распознаванием текста, второй связан с полиграфией и издательским делом, третий — с обработкой аэрофотоснимков и изображений, полученных со спутников. Аналогичные результаты могут быть получены методами низкочастотной пространственной фильтрации, рассмотренными в разделе 3.5.

На рис. 4.49 представлен образец текста низкого разрешения. Мы сталкиваемся с текстом подобного рода в случаях, когда имеем дело, например, с передачей сообщений по факсу, с материалами, полученными в результате копирования или с архивными записями. Данный конкретный образец не содержит дополнительно проблемных участков в виде пятен, складок и разрывов. Увеличенный фрагмент на рис. 4.49(*а*) показывает, что буквы в документе из-за недостаточного разрешения сильно искажены, причем многие из них разорваны. Хотя человеческое зрение без труда заполняет образовавшиеся пробелы, автоматические системы сталкиваются с серьезными трудностями при распознавании разорванных

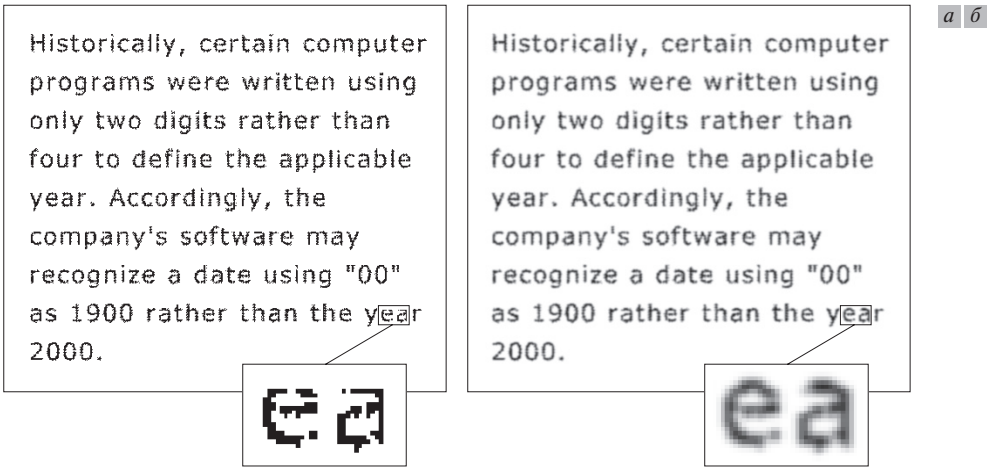


Рис. 4.49. (*а*) Образец текста низкого разрешения (обратите внимание на разрывы букв в увеличенном фрагменте). (*б*) Результат фильтрации с применением ФНЧ Гаусса (произошло соединение разорванных сегментов)

символов. Один из подходов к решению этой проблемы заключается в сглаживании исходного изображения, позволяющем перекрыть небольшие разрывы. Рис. 4.49(*а*) показывает, насколько хорошо мы можем восстановить знаки при помощи простой процедуры, использующей гауссов низкочастотный фильтр с частотой среза $D_0 = 80$. Размер изображений составляет 444×508 пикселей.

Низкочастотная фильтрация является одним из основных инструментов в полиграфии и издательском деле, где она используется в многочисленных процедурах предобработки, включая нерезкое маскирование, как это обсуждалось в разделе 3.6.3. Другое применение низкочастотной фильтрации — «косметическая» обработка, предшествующая печати. Рис. 4.50 демонстрирует применение низкочастотной фильтрации для получения более гладкого и приятного для глаза изображения из резкого оригинала. В случае человеческого лица типичной задачей является понижение резкости тонких линий и небольших пятен на коже. На увеличенных фрагментах рис. 4.50(*б*) и (*в*) хорошо видно значительное снижение резкости морщин вокруг глаз. В действительности сглаженные изображения выглядят вполне привлекательно и приятно для глаза.

Нерезкое маскирование в частотной области рассматривается в разделе 4.9.5.

На рис. 4.51 показаны примеры применения к одному и тому же изображению двух процедур низкочастотной фильтрации, преследовавших совершенно различные цели. На рис. 4.51(*а*) показан фрагмент панорамы размерами 808×754 элементов, на котором изображена часть Мексиканского залива (темные участки) и Флориды (светлые участки), полученной со спутника NOAA при помощи радиометра высокого разрешения. Обратите внимание на горизонтальные линии, совпадающие с направлением движения сенсоров. (Появление границ между большими участками водной поверхности связано с петлевыми течени-

а б в

Рис. 4.50. (*а*) Исходное изображение (784×732 пикселя). (*б*) Результат фильтрации с применением ФНЧ Гаусса с $D_0 = 100$. (*в*) Результат фильтрации с применением ФНЧ Гаусса с $D_0 = 80$. Обратите внимание на значительное снижение резкости морщин в увеличенных фрагментах (*б*) и (*в*)

а б в

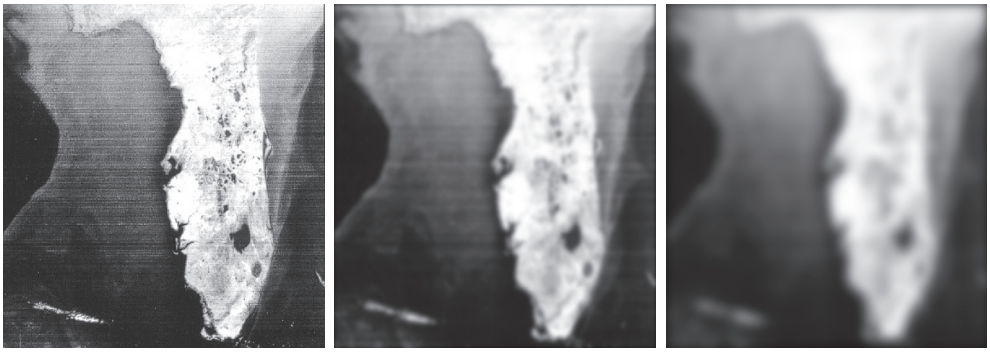


Рис. 4.51. (*а*) Изображение с заметными горизонтальными линиями сканирования. (*б*) Результат, полученный при помощи ФНЧ Гаусса с $D_0 = 50$. (*в*) Результат, полученный при помощи ФНЧ Гаусса с $D_0 = 20$. (Исходное изображение предоставлено NOAA)

ями). Данный фрагмент представляет собой типичный пример изображений, формируемых сканерами с линейками сенсоров, широко используемыми в дистанционном зондировании; характеристики сенсоров в линейке различаются, в результате чего возникают отчетливые линии в направлении сканирования (см. пример 4.24, иллюстрирующий физические причины этого). Низкочастотная фильтрация является грубым, но простым способом уменьшить визуальный эффект, обусловленный наличием этих линий, что показано на рис. 4.51(*б*) (более действенный подход будет рассмотрен в разделах 4.10 и 5.4.2). Это изображение было получено использованием ФНЧ Гаусса с $D_0 = 50$. Достигнутое уменьшение обсуждаемого эффекта может упростить обнаружение таких характерных деталей, как границы раздела океанических течений.

Рис. 4.21(*в*) представляет результат применения значительно более сильного гауссова низкочастотного фильтра ($D_0 = 20$). Здесь цель обработки состояла в том, чтобы стереть как можно больше деталей, оставив узнаваемыми лишь большие характерные части изображения. Между прочим, фильтрация такого типа бывает частью этапа предварительной обработки в системах анализа изображений, ориентированных на поиск объектов в банке данных изображений. Примером таких объектов могут служить озера заданного размера (такие как озеро Окичоби в юго-восточной части полуострова Флорида, которое на рис. 4.51(*в*) выглядит как почти круглая темная область). Низкочастотная фильтрация помогает упростить анализ за счет сглаживания изображения на участках с деталями, имеющими размеры меньше интересующих.

4.9. Повышения резкости изображений частотными фильтрами

В предыдущем разделе было показано, что изображение может быть сглажено путем подавления высокочастотных составляющих его Фурье-преобразования. Поскольку контуры и другие скачкообразные изменения яркости связаны с вы-

сокочастотными составляющими, повышение резкости изображения может быть достигнуто при помощи процедуры *высокочастотной фильтрации* в частотной области, которая подавляет низкочастотные составляющие и не затрагивает высокочастотную часть Фурье-преобразования. Как и в разделе 4.8, мы рассматриваем центрально-симметричные фильтры с нулевым фазовым сдвигом. Все обсуждаемые в этом разделе методы фильтрации основаны на схеме, описанной в разделе 4.7.3. Так что все фильтр-функции $H(u,v)$ понимаются как дискретные функции размерами $P \times Q$; т. е. дискретные частотные переменные находятся в диапазоне $u = 0, 1, 2, ..., P-1$ и $v = 0, 1, 2, ..., Q-1$.

Передаточная функция высокочастотного фильтра может быть получена из заданного низкочастотного фильтра при помощи соотношения

$$H_{\text{HP}}(u,v) = 1 - H_{\text{LP}}(u,v), \tag{4.9-1}$$

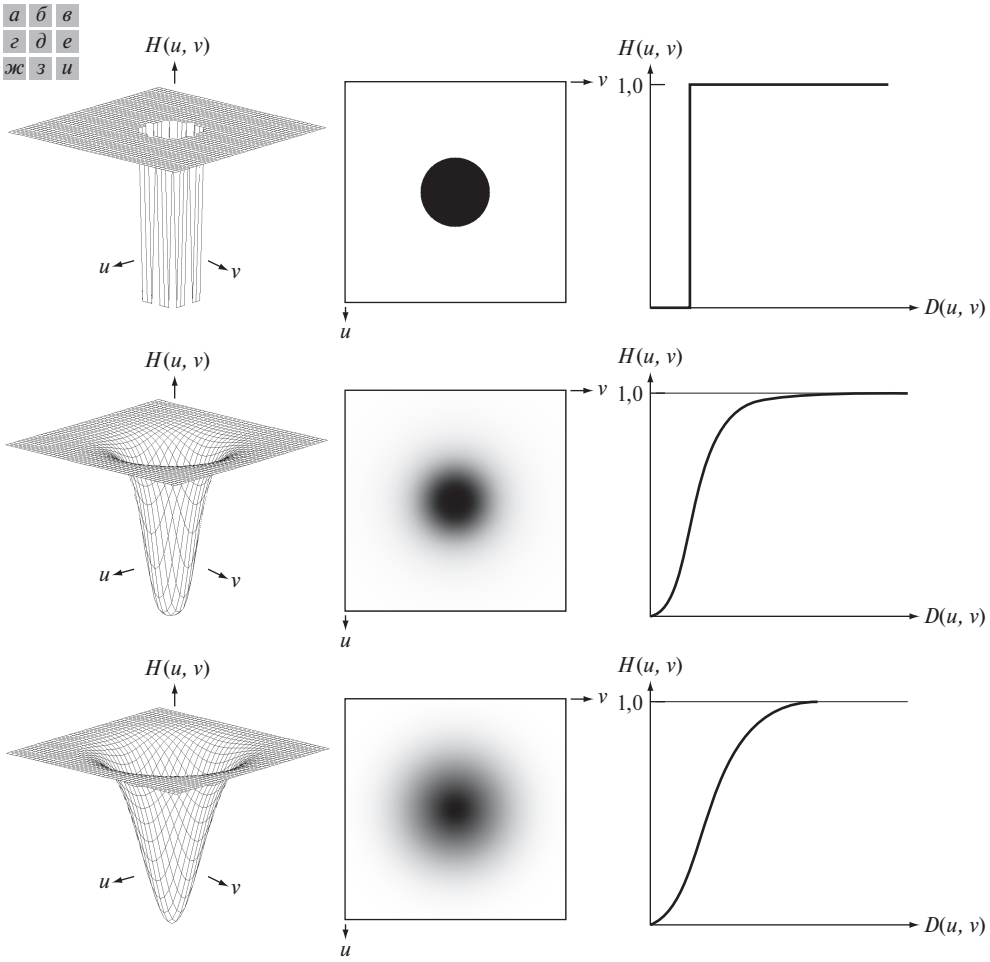


Рис. 4.52. Верхний ряд: трехмерный график, полутоновое изображение и профиль типичного идеального высокочастотного фильтра. Средний и нижний ряды: та же последовательность для типичных высокочастотных фильтров Баттерворта и Гаусса

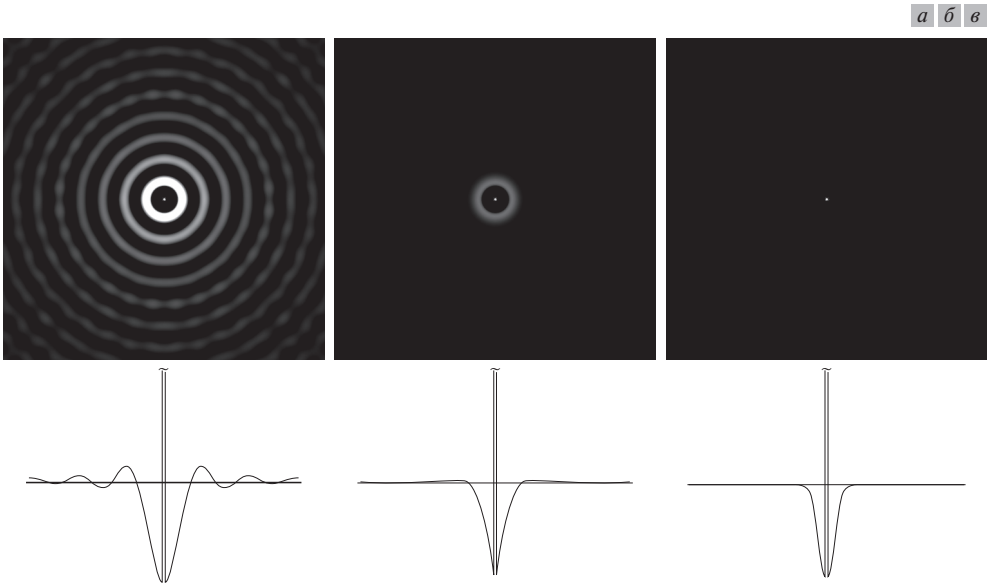


Рис. 4.53. Представление в пространственной области типичных высокочастотных фильтров: (*а*) идеальный фильтр, (*б*) фильтр Баттерворта, (*в*) гауссов фильтр; внизу представлены соответствующие им профили яркости строк, проходящих через центр

где $H_{LP}(u,v)$ обозначает передаточную функцию соответствующего низкочастотного фильтра. Таким образом, частоты, ослабляемые низкочастотным фильтром, пропускаются высокочастотным фильтром, и наоборот.

В этом разделе мы рассматриваем идеальные высокочастотные фильтры, высокочастотные фильтры Баттерворта и гауссовы высокочастотные фильтры. Как и в предыдущем разделе, мы изучаем свойства этих фильтров как в частотной, так и в пространственной областях. На рис. 4.22 представлены трехмерные и полутоновые изображения, а также профили типичных фильтров каждого из перечисленных видов. Как и ранее, мы видим, что фильтр Баттерворта занимает промежуточное положение между резкостью идеального фильтра и сильной сглаженностью гауссова фильтра. Рис. 4.53, обсуждаемый в следующем разделе, дает представление о поведении этих фильтров в пространственной области. Пространственные фильтры были получены и отображены тем же способом, что и рис. 4.43 и 4.46.

4.9.1. Идеальные фильтры высоких частот

Двумерный идеальный фильтр высоких частот (идеальный ФВЧ) определяется формулой

$$H(u,v)=\begin{cases}0 & \text{при } D(u,v)\le D_0;\\ 1 & \text{при } D(u,v)>D_0,\end{cases} \qquad (4.9\text{-}2)$$

где D_0 — частота среза, а величина $D(u,v)$ задается формулой (4.8-2). Это прямо следует из (4.8-1) и (4.9-1). Как и подразумевалось, действие идеального ФВЧ противоположно действию идеального ФНЧ в том смысле, что он обнуляет все

частоты, попадающие внутрь круга радиуса D_0, одновременно пропуская без ослабления все частоты, лежащие вне круга. Как и идеальный низкочастотный фильтр, идеальный ФВЧ не может быть реализован при помощи электронных устройств. Однако мы рассмотрим его для полноты, а также поскольку его свойства могут быть использованы для объяснений различных явлений, таких как звон в пространственной области. Наше обсуждение будет кратким.

Исходя из соотношения (4.9-1), связывающего фильтры высоких и низких частот, мы вправе ожидать, что идеальные ФВЧ обладают такими же свойствами в отношении звона, как и идеальные ФНЧ. Это ясно демонстрирует рис. 4.54, который состоит из результатов обработки исходного изображения, представленного на рис. 4.41(*а*), при помощи различных идеальных ФВЧ со значениями частоты среза $D_0 = 30$, 60 и 160 пикселей соответственно. Звон на рис. 4.54(*а*) настолько велик, что он привел к деформированию и утолщению границ объектов (посмотрите, например, на большую букву «а»). Границы трех верхних кругов почти не видны, поскольку их контраст мал по сравнению с контрастами других объектов на изображении (яркость этих трех объектов гораздо ближе к яркости фона, что приводит к меньшей величине разрывов). Если посмотреть на размер «пятна» на изображении идеального ФВЧ (см. рис. 4.53(*а*)) и вспомнить, что фильтрация в пространственной области есть свертка пространственного фильтра с изображением, то это поможет объяснить, почему маленькие объекты и линии выглядят как почти целиком белые. Посмотрите, в частности, на три маленьких квадрата в верхнем ряду и на тонкие вертикальные полосы на рис. 4.54(*а*). Ситуация до некоторой степени улучшается в случае $D_0 = 60$. Деформация контуров все еще достаточно очевидна, но теперь мы начинаем видеть фильтрацию на маленьких объектах. Уже хорошо знакомое обратное отношение между шириной фильтра в частотной и пространственной области говорит о том, что размер пятна этого фильтра меньше, чем размер пятна фильтра с $D_0 = 30$. Результат для $D_0 = 160$ близок к тому, как должен выглядеть результат высокочастотной фильтрации. Контуры здесь гораздо более ровные и меньше искажены, и маленькие объекты отфильтрованы надлежащим образом. Постоянный фон на всех изображениях, обработанных высокочастотными фильтрами, конечно, равен нулю, поскольку высокочастотная фильтрация аналогична дифференцированию в пространственной области.

а *б* *в*

Рис. 4.54. Результаты высокочастотной фильтрации изображения на рис. 4.41(*а*); использовался идеальный ФВЧ с $D_0 = 30$, 60 и 160

4.9.2. Фильтры высоких частот Баттерворта

Передаточная функция двумерного *высокочастотного фильтра Баттерворта* (ФВЧ Баттерворта) порядка n с частотой среза D_0 задается формулой

$$H(u,v)=\frac{1}{1+\left[D_0/D(u,v)\right]^{2n}}, \tag{4.9-3}$$

где расстояние $D(u,v)$ вычисляется согласно (4.8-2). Это выражение прямо следует из (4.8-5) и (4.9-1). На рис. 4.52 в среднем ряду представлено изображение и профиль передаточной функции ФВЧ Баттерворта.

Как и в случае низкочастотных фильтров, мы вправе ожидать от высокочастотных фильтров Баттерворта более гладкого поведения по сравнению с идеальными ФВЧ. На рис. 4.55 показаны результаты применения ФНЧ Баттерворта порядка 2 с частотами среза D_0, принимающими те же значения, что и на рис. 4.54. Искажения границ объектов существенно меньше, чем на рис. 4.54, даже для наименьшего значения частоты среза. Поскольку размеры центрального пятна для идеального ФВЧ и ФВЧ Баттерворта близки (см. рис. 4.53(*а*) и (*б*)), результаты фильтрации малых объектов обоими фильтрами приводит к сопоставимым результатам. Переход к более высоким значениям частоты среза для ФВЧ Баттерворта совершается гораздо более плавно.

4.9.3. Гауссовы фильтры высоких частот

Передаточная функция гауссова фильтра высоких частот (ФВЧ Гаусса) с частотой среза, расположенной на расстоянии D_0 от центра частотного прямоугольника, задается формулой

$$H(u,v)=1-e^{-D^2(u,v)/2D_0^2}, \tag{4.9-4}$$

где расстояние $D(u,v)$ вычисляется согласно (4.8-2). Это выражение прямо следует из (4.8-7) и (4.9-1). На рис. 4.52 в нижнем ряду представлены трехмерный график, изображение и профиль передаточной функции ФНЧ Гаусса. Придер-

а б в

Рис. 4.55. Результаты высокочастотной фильтрации изображения на рис. 4.41(*а*) с использованием ФВЧ Баттерворта порядка 2 с D_0 = 30, 60 и 160, что соответствует кругам на рис. 4.41(*б*). Данные результаты существенно более гладкие, чем полученные с применением идеального ФВЧ

а б в

Рис. 4.56. Результаты высокочастотной фильтрации изображения на рис. 4.41(*а*) с использованием ФВЧ Гаусса с D_0 = 30, 60 и 160, что соответствует кругам на рис. 4.41(*б*). Сравните с рис. 4.54 и 4.55

живаясь того же формата, как при обсуждении ФНЧ Баттерворта, мы приводим на рис. 4.56 заслуживающие сравнения результаты, посчитанные с использованием ФНЧ Гаусса. Как и следовало ожидать, полученные изображения являются более гладкими, чем при использовании предыдущих двух фильтров. Гауссов фильтр дает хорошее качество фильтрации даже для маленьких объектов и тонких полос. В табл. 4.5 приведена краткая сводка высокочастотных фильтров, рассмотренных в настоящем разделе.

Таблица 4.5. Высокочастотные фильтры. D_0 есть частота среза, а n — порядок фильтра Баттерворта

Идеальный	Баттерворта	Гауссов
$H(u,v)=\begin{cases}0 & \text{при } D(u,v)\le D_0\\ 1 & \text{при } D(u,v)>D_0\end{cases}$	$H(u,v)=\dfrac{1}{1+\left[D_0/D(u,v)\right]^{2n}}$	$H(u,v)=1-e^{-D^2(u,v)/2D_0^2}$

Пример 4.19. Применение высокочастотной фильтрации и срезки для повышения контраста изображений.

■ На рис. 4.57(*а*) представлено изображение отпечатка большого пальца размерами 1026×962 пикселей с хорошо заметными пятнами грязи (обычная проблема). Важным шагом процесса автоматического распознавания отпечатков является улучшение отпечатков линий папиллярного узора и устранение загрязнения. Улучшение полезно и для визуальной интерпретации отпечатка экспертом. В данном примере рассматривается применение высокочастотной фильтрации для улучшения отпечатков линий и уменьшения эффекта загрязнения. Улучшение отпечатков линий достигается использованием того факта, что они в основном содержат высокие частоты, которые остаются неизменными в результате высокочастотной фильтрации. С другой стороны, ФВЧ снижает низкочастотные компоненты, которые отвечают за медленные изменения яркости, такие как фон и грязь. Тем самым улучшение достигается за счет снижения влияния всех особенностей, кроме тех, которые передаются высокими частотами и представляют интерес в данном случае.

Рис. 4.57. (*а*) Изображение отпечатка большого пальца. (*б*) Результат высокочастотной фильтрации (*а*). (*в*) Результат порогового разделения (*б*). (Исходное изображение предоставлено Национальным институтом стандартов и технологий США)

Изображение на рис. 4.57(*б*) — результат применения фильтра высоких частот Баттерворта порядка 4 с частотой среза равной 50. Как и следовало ожидать, в результате высокочастотной фильтрации яркость изображения упала, поскольку постоянная составляющая яркости была уменьшена до 0. Можно сделать вывод, что после высокочастотной фильтрации на изображении доминируют темные тона, что требует дополнительной обработки для улучшения интересующих деталей. Простым приемом может являться пороговая срезка фильтрованного изображения. Результат присвоения элементам с отрицательными величинами значения черного, а элементам с положительными величинами — белого цвета представлен на рис. 4.57(*в*). Отметим, насколько ясно стали видны папиллярные линии и значительно уменьшился эффект загрязнения. Фактически линии, которые в верхней правой части изображения рис. 4.57(*а*) были едва заметны, на рис. 4.57(*в*) стали прекрасно видны. ■

Значение $D_0 = 50$ составляет приблизительно 2,5 % размера короткой стороны расширенного изображения. Рекомендуется выбирать D_0 близко к началу координат, чтобы низкие частоты подавлялись, но не убирались полностью. Диапазон от 2 % до 5 % размера короткой стороны изображения можно считать хорошей отправной точкой.

4.9.4. Лапласиан в частотной области

В разделе 3.6.2 лапласиан в пространственной области уже применялся для улучшения изображения. Здесь мы вернемся к лапласиану еще раз и покажем, что при использовании частотных методов он дает эквивалентные результаты. Можно показать (задача 4.26), что лапласиан может быть реализован в частотной области при помощи фильтра

$$H(u,v) = -4\pi^2(u^2 + v^2). \qquad (4.9\text{-}5)$$

Используя функцию расстояния от центра частотной области $D(u,v)$, определенную выражением (4.8-2), можно переписать (4.9-5) в виде

$$H(u,v) = -4\pi^2[(u - P/2)^2 + (v - Q/2)^2] = -4\pi^2 D^2(u,v). \qquad (4.9\text{-}6)$$

Тогда лапласиан изображения будет выглядеть как

$$\nabla^2 f(x,y) = \Im^{-1}\{H(u,v)F(u,v)\}, \tag{4.9-7}$$

где $F(u,v)$ есть ДПФ от $f(x,y)$. Как объяснялось в разделе 3.6.2, улучшение изображения достигается использованием следующего выражения:

$$g(x,y) = f(x,y) + c\nabla^2 f(x,y). \tag{4.9-8}$$

Здесь $c = -1$, поскольку $H(u,v)$ отрицательно. В главе 3 $f(x,y)$ и $\nabla^2 f(x,y)$ имели сравнимые по величине значения. Однако вычисление $\nabla^2 f(x,y)$ согласно выражению (4.9-7) вносит масштабный коэффициент обратного ДПФ, который может на несколько порядков превышать максимальное значение f. Значит, необходимо вернуть разницу между f и ее лапласианом в сравнимые диапазоны. Простейший путь решения этой проблемы состоит в нормировке значений $f(x,y)$ в диапазон [0, 1] перед вычислением ДПФ и делением лапласиана $\nabla^2 f(x,y)$ на свое максимальное значение, что приведет его в приблизительный диапазон [–1, 1] (напомним, что лапласиан имеет также и отрицательные значения). После этого уже можно применять выражение (4.9-8).

В частотной области выражение (4.9-8) записывается в виде

$$\begin{aligned} g(x,y) &= \Im^{-1}\{F(u,v) - H(u,v)F(u,v)\} = \\ &= \Im^{-1}\{[1 - H(u,v)]F(u,v)\} = \\ &= \Im^{-1}\{[1 + 4\pi^2 D^2(u,v)]F(u,v)\}. \end{aligned} \tag{4.9-9}$$

Хотя этот результат выглядит элегантно, тем не менее ему присущи такие же самые трудности с масштабом, как были только что рассмотрены, осложняющиеся тем, что нормировочный коэффициент не так легко вычислить. По этим причинам выражение (4.9-8) является предпочтительным способом реализации в частотной области при вычислении $\nabla^2 f(x,y)$ согласно (4.9-7) и масштабировании с помощью подхода, изложенного в предыдущем параграфе.

Пример 4.20. Повышение резкости изображения с помощью лапласиана в частотной области.

■ Изображение на рис. 4.58(*а*) является тем же, что и на рис. 3.38(*а*), рис. 4.58(*б*) демонстрирует результат применения выражения (4.9-8), в котором лапласиан вычислялся в частотной области по формуле (4.9-7). Масштабирование было таким, как описано в контексте этого выражения. Сравнение рис. 4.58(*б*) и 3.38(*д*) показывает, что результаты, полученные в частотной и пространственной областях, визуально идентичны. Обратим внимание, что оба приведенных результата соответствуют маске лапласиана на рис. 3.37(*б*), имеющей в центре коэффициент –8 (задача 4.26). ■

4.9.5. Нерезкое маскирование, высокочастотная фильтрация с подъемом частотной характеристики, фильтрация с усилением высоких частот

В этом разделе будут рассматриваться способы реализации в частотной области методов нерезкого маскирования и повышения резкости, использующих филь-

а б

Рис. 4.58. (*а*) Исходное размытое изображение. (*б*) Улучшение изображения использованием лапласиана в частотной области. Сравните с рис. 3.38(*д*)

трацию с подъемом высоких частот, пространственная реализация которых обсуждалась в разделе 3.6.3. Маска, задаваемая выражением (3.6-8), с помощью средств частотной области записывается в виде

$$g_{\text{mask}}(x,y) = f(x,y) - f_{\text{LP}}(x,y) \tag{4.9-10}$$

с

$$f_{\text{LP}}(x,y) = \Im^{-1}[H_{\text{LP}}(u,v)F(u,v)], \tag{4.9-11}$$

где $H_{\text{LP}}(u,v)$ является фильтром низких частот, а $F(u,v)$ — Фурье-преобразованием от $f(x,y)$. Здесь $f_{\text{LP}}(x,y)$ является сглаженным изображением, аналогом $\overline{f}(x,y)$ в (3.6-8). Тогда, как и в (3.6-9),

$$g(x,y) = f(x,y) + kg_{\text{mask}}(x,y). \tag{4.9-12}$$

При $k = 1$ данное выражение задает нерезкое маскирование, а при $k > 1$ — фильтрацию с усилением высоких частот. Используя предыдущие результаты, (4.9-12) можно выразить полностью в терминах частотной области, используя фильтр низких частот:

$$g(x,y) = \Im^{-1}\left\{\left[1 + k[1 - H_{\text{LP}}(u,v)]\right]F(u,v)\right\}. \tag{4.9-13}$$

Используя (4.9-1), можно выразить результат в терминах фильтра высоких частот:

$$g(x,y) = \Im^{-1}\left\{\left[1 + kH_{\text{HP}}(u,v)\right]F(u,v)\right\}. \tag{4.9-14}$$

В данном выражении внутри квадратных скобок содержится так называемый *фильтр усиления высоких частот*. Фильтр высоких частот, как отмечалось ранее, устанавливает постоянную составляющую в ноль, тем самым снижая среднюю яркость фильтрованного изображения до 0. Фильтр усиления высоких частот лишен этого недостатка, поскольку в нем к значению фильтра высоких частот

прибавляется 1. Константа k управляет пропорцией вклада высоких частот в финальный результат. Несколько более общей формулировкой фильтра усиления высоких частот является следующее выражение:

$$g(x,y) = \Im^{-1}\left\{\left[k_1 + k_2 * H_{\text{НР}}(u,v)\right]F(u,v)\right\}, \qquad (4.9\text{-}15)$$

где $k_1 \geq 0$ задает общую яркость (смещение относительно начала координат, см. рис. 4.31(*в*)), а $k_2 \geq 0$ управляет вкладом высоких частот.

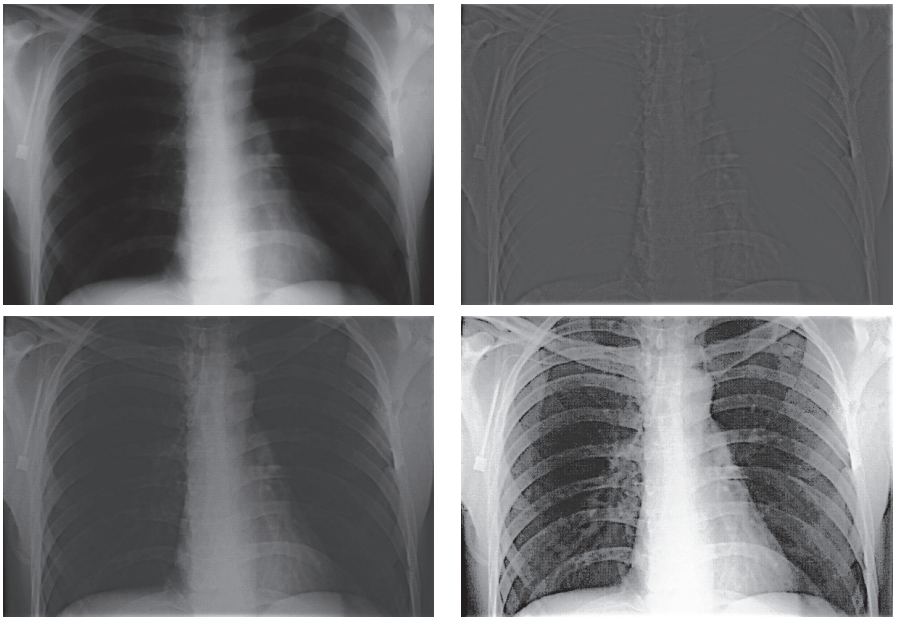
Пример 4.21. Улучшение изображения при помощи фильтрации с усилением высоких частот.

■ На рис 4.59(*а*) представлен рентгеновский снимок грудной клетки размерами 416×596 пикселей с узким диапазоном изменения яркости. Целью данного примера является улучшение изображения при помощи фильтрации с усилением высоких частот. Поскольку рентгеновские лучи не могут быть сфокусированы аналогично фокусировке световых лучей при помощи линз, то рентгеновские снимки, как правило, выглядят слегка расплывчато. Поскольку в нашем конкретном случае яркость изображения в целом сдвинута в темную область, мы также используем связанную с этим возможность и приведем пример того, как обработка в пространственной области может дополнять обработку в частотную области.

Рис. 4.59(*б*) отображает результат высокочастотной фильтрации с использованием гауссова фильтра с частотой среза $D_0 = 40$ (приблизительно равной 5% размера короткой стороны расширенного изображения. Как и можно было ожидать, полученное после фильтрации изображение маловыразительно, на нем лишь едва видны основные контуры оригинала. Изображение на рис. 4.59(*в*) демонстрирует преимущества фильтрации с усилением высоких частот, где использовалось выражение (4.9-14) с $k_1 = 0{,}5$ и $k_2 = 0{,}75$. Хотя изображение попрежнему темное, общий яркостной тон, обусловленный низкочастотными составляющими, сохранен.

> Возникновение артефактов, таких как звон, на медицинских изображениях недопустимо. Поэтому правильным будет не допускать использования фильтров, имеющих потенциальные возможности появления артефактов на обработанном изображении. Поскольку гауссовы фильтры в пространственной и частотной областях являются Фурье-парой, эти фильтры дают результаты сглаживания, свободные от артефактов.

Как обсуждалось в разделе 3.3.1, изображение, характеризующееся узким диапазоном яркости, является идеальным кандидатом для обработки с помощью метода эквализации гистограммы. Как показывает рис. 4.59(*г*), применение этого метода оказалось целесообразным для дальнейшего улучшения нашего изображения. Обратите внимание на проявление костной структуры и других деталей, которые просто не видны на других трех изображениях. Шум на результирующем изображении немного увеличился, но это типично для рентгеновских изображений при растяжении диапазона их яркостей. Результат, полученный при совместном использовании фильтрации с усилением высоких частот и эквализации гистограммы, превосходит по качеству тот, который может быть получен при использовании каждого из методов по отдельности. ■



а б
в г

Рис. 4.59. (*а*) Рентгеновский снимок грудной клетки. (*б*) Результат высокочастотной фильтрации с использованием гауссова фильтра. (*в*) Результат фильтрации с усилением высоких частот при использовании того же фильтра. (*г*) Результат применения метода эквализации гистограммы к (*в*). (Исходное изображение предоставил д-р Томас Р. Гест, отделение анатомии медицинского факультета университета шт. Мичиган)

4.9.6. Гомоморфная фильтрация

Модель формирования изображения, основанная на освещении—отражении, с которой мы познакомились в разделе 2.3.4, может быть использована для развития еще одного метода обработки в частотной области, направленного на улучшение изображения путем одновременного сжатия яркостного диапазона и усиления контраста. Согласно приводимому там обсуждению, изображение $f(x,y)$ может быть представлено в виде произведения его освещенности $i(x,y)$ и коэффициента отражения $r(x,y)$:

$$f(x,y) = i(x,y)r(x,y). \tag{4.9-16}$$

Данное равенство не дает непосредственной возможности работать с частотными составляющими освещенности и коэффициента отражения по отдельности, поскольку преобразование Фурье произведения не равно произведению преобразований Фурье сомножителей; другими словами,

$$\mathfrak{F}\{f(x,y)\} \neq \mathfrak{F}\{i(x,y)\}\mathfrak{F}\{r(x,y)\}. \tag{4.9-17}$$

Рассмотрим, однако, величину

$$z(x,y) = \ln f(x,y) = \ln i(x,y) + \ln r(x,y). \tag{4.9-18}$$

Если изображение $f(x,y)$ с яркостями в диапазоне $[0, L-1]$ имеет нулевые значения, то к значению каждого элемента следует добавить 1, чтобы предотвратить появление ln(0). По окончании процесса фильтрации данную единицу следует вычесть из результата.

Тогда

$$\Im\{z(x,y)\} = \Im\{\ln f(x,y)\} = \Im\{\ln i(x,y)\} + \Im\{\ln r(x,y)\} \tag{4.9-19}$$

или

$$Z(u,v) = F_i(u,v) + F_r(u,v), \tag{4.9-20}$$

где $F_i(u,v)$ и $F_r(u,v)$ — Фурье-образы функций ln $i(x,y)$ и ln $r(x,y)$ соответственно.

Функцию $Z(u,v)$ можно подвергнуть процедуре фильтрации с помощью фильтра $H(u,v)$, что даст Фурье-образ результата $S(u,v)$:

$$S(u,v) = H(u,v)Z(u,v) = H(u,v)F_i(u,v) + H(u,v)F_r(u,v). \tag{4.9-21}$$

В пространственной области получим фильтрованное изображение:

$$s(x,y) = \Im^{-1}\{S(u,v)\} = \Im^{-1}\{H(u,v)F_i(u,v)\} + \Im^{-1}\{H(u,v)F_r(u,v)\}. \tag{4.9-22}$$

Если обозначить

$$i'(x,y) = \Im^{-1}\{H(u,v)F_i(u,v)\} \tag{4.9-23}$$

и

$$r'(x,y) = \Im^{-1}\{H(u,v)F_r(u,v)\}, \tag{4.9-24}$$

то равенство (4.9-22) можно выразить в виде

$$s(x,y) = i'(x,y) + r'(x,y). \tag{4.9-25}$$

Наконец, поскольку функция $z(x,y)$ была сформирована как натуральный логарифм исходного изображения $f(x,y)$, то обратная операция (потенцирование) результата фильтрации позволяет получить искомое обработанное изображение, обозначаемое $g(x,y)$:

$$g(x,y) = e^{s(x,y)} = e^{i'(x,y)} \cdot e^{r'(x,y)} = i_0(x,y)r_0(x,y), \tag{4.9-26}$$

где

$$i_0(x,y) = e^{i'(x,y)} \tag{4.9-27}$$

и

$$r_0(x,y) = e^{r'(x,y)} \tag{4.9-28))}$$

представляют собой освещенность и коэффициент отражения результирующего (обработанного) изображения.

Только что описанный подход к фильтрации схематически представлен на рис. 4.60. Данный метод является частным случаем класса систем, называемых *гомоморфными системами*. В нашем случае ключевым моментом рассматриваемого подхода является разложение изображения на составляющие, связанные с освещенностью и коэффициентом отражения, в виде (4.9-20). После этого

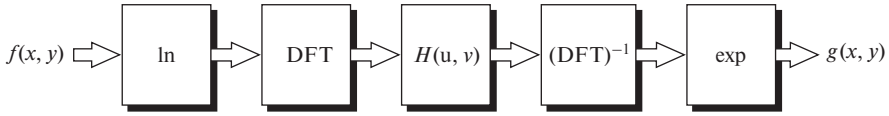


Рис. 4.60. Этапы гомоморфной фильтрации

гомоморфный фильтр $H(u,v)$ действует на каждую из полученных составляющих по отдельности, так как показывает (4.9-21).

Составляющая изображения, связанная с освещенностью, обычно характеризуется медленными изменениями в пространственной области, в то время как составляющая, обусловленная коэффициентом отражения, склонна к резким изменениям, особенно в местах соединения разнородных объектов. Такое поведение позволяет ассоциировать низкочастотную составляющую преобразования Фурье от логарифма изображения с освещенностью, а высокочастотную — с коэффициентом отражения. Хотя такие ассоциации правильны лишь в грубом приближении, они могут быть полезны при фильтрации изображений, как это показано в примере 4.22.

Использование гомоморфного фильтра предоставляет возможность в значительной степени контролировать каждую из означенных составляющих. Для этого требуется задать передаточную функцию $H(u,v)$ так, чтобы на низкочастотные и высокочастотные составляющие Фурье-преобразования фильтр воздействовал по-разному контролируемым образом. На рис. 4.61 показан профиль такого фильтра. В том случае, если параметры γ_L и γ_H выбраны так, что $\gamma_L < 1$ и $\gamma_H > 1$, то показанный на рис. 4.61 фильтр будет ослаблять вклад, вносимый низкими частотами (освещенностью), и усиливать вклад, вносимый высокими частотами (коэффициентом отражения). Конечный результат заключается в одновременном сжатии динамического диапазона и усилении контраста.

Форма кривой, показанной на рис. 4.61, может быть аппроксимирована с помощью использования любых из основных видов высокочастотных фильтров. Например, применение слегка видоизмененного гауссова высокочастотного фильтра дает

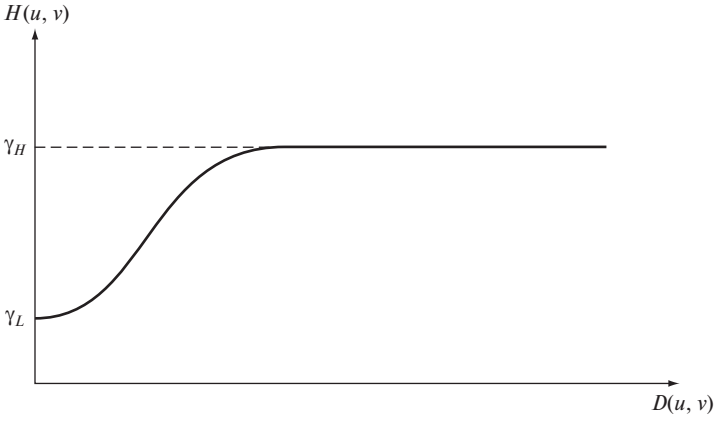


Рис. 4.61. Радиальный профиль центрально-симметричной передаточной функции гомоморфного фильтра. Вертикальная ось находится в центре частотного прямоугольника, а $D(u,v)$ — расстояние от центра

$$H(u,v) = (\gamma_H - \gamma_L)\left[1 - e^{-c(D^2(u,v)/D_0^2)}\right] + \gamma_L, \qquad (4.9\text{-}29)$$

где $D(u,v)$ задается формулой (4.8-2), а константа c управляет крутизной наклона передаточной функции фильтра в переходной области между γ_L и γ_H. Фильтр данного вида похож на фильтр усиления высоких частот, который обсуждался в предыдущем разделе.

Пример 4.22. Улучшение изображения при помощи гомоморфной фильтрации.

■ На рис. 4.62(*а*) представлен скан тела, полученный ПЭТ (позитронным эмиссионным томографом), имеющий размеры 1162×746 пикселей. Изображение несколько размыто, а многие детали в темных участках подавлены за счет нескольких выделяющихся пятен высокой яркости, которые определяют динамический диапазон отображения. (Эти пятна вызваны наличием опухолей в мозгу и в легких.) Изображение на рис. 4.62(*б*) получено при помощи гомоморфной фильтрации изображения (*а*), с использованием фильтра, заданного выражением (4.9-29) с $\gamma L = 0{,}25$, и $\gamma H = 2$, $c = 1$ и D_0=80. Профиль данного фильтра выглядит почти как показанный на рис. 4.61 со слегка более крутым наклоном.

Напомним, что фильтрация использует расширение изображения, так что фильтр имеет размеры $P \times Q$.

а *б*

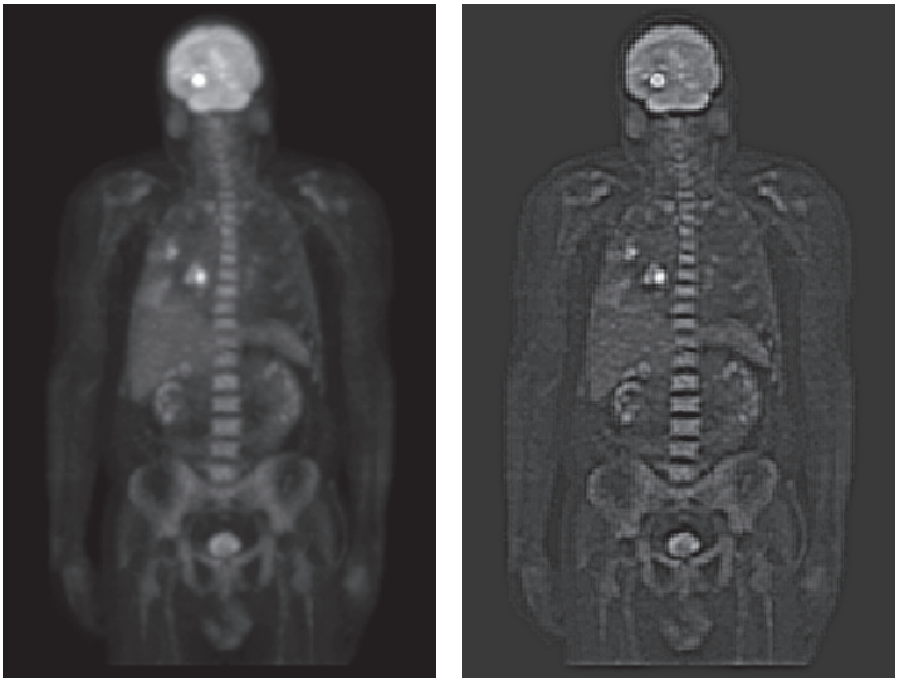


Рис. 4.62. (*а*) Скан тела, полученный ПЭТ. (*б*) Изображение, обработанное при помощи гомоморфной фильтрации. (Исходное изображение предоставил д-р Майкл Е. Кейси, CTI PET Systems)

Обратим внимание, насколько резче на обработанном изображении рис. 4.62(*б*) выглядят яркие пятна, мозг и скелет и как много новых деталей стало видно. Уменьшение эффекта влияния доминирующих пятен позволило расширить динамический диапазон в тенях и тем самым сделать детали малой яркости более заметными. Поскольку гомоморфная фильтрация усилила верхние частоты, то резкость «отражающей компоненты» (в данном случае контуров) тоже значительно повысилась. Обработанное изображение на рис. 4.62(*б*) заметно улучшено по сравнению с исходным. ■

4.10. Избирательная фильтрация

Рассматривавшиеся в двух предыдущих разделах фильтры действуют по всему частотному прямоугольнику. Но существуют приложения, в которых представляет интерес обработка отдельной полосы частот либо же небольшой области частотного прямоугольника. Фильтры, пропускающие определенную полосу частот, называют *полосовыми фильтрами* или *полосовыми пропускающими фильтрами*. Фильтры, задерживающие определенную полосу частот, называют *полосовыми заграждающими* (или *режекторными*) *фильтрами*; узкополосный заграждающий фильтр иногда называют *фильтром-пробкой*[23].

4.10.1. Режекторные и полосовые пропускающие фильтры

Фильтры этого типа легко создавать, используя концепции предыдущих двух разделов. В табл. 4.6 приведены выражения для нескольких режекторных фильтров — идеального, Баттерворта и гауссова; $D(u,v)$ означает расстояние от центра частотного прямоугольника согласно (4.8-2), D_0 — значение радиуса до середины полосы, а W — ширина полосы. На рис. 4.63(*а*) показан режекторный гауссов фильтр в виде изображения, где белое означает 1, а черное — 0.

Таблица 4.6. Режекторные фильтры. W означает ширину полосы, D — расстояние $D(u,v)$ от центра фильтра; D_0 — частота среза, а n — порядок фильтра Баттерворта. В формулах указано D вместо $D(u,v)$ для упрощения записи

Идеальный	Баттерворта	Гауссов
$H(u,v)=\begin{cases}0 & \text{при } D_0-\frac{W}{2}\le D\le D_0+\frac{W}{2}\\ 1 & \text{в остальных случаях}\end{cases}$	$H(u,v)=\frac{1}{1+\left[\frac{DW}{D^2-D_0^2}\right]^{2n}}$	$H(u,v)=1-e^{-\left[\frac{D^2-D_0^2}{DW}\right]^2}$

Полосовой пропускающий фильтр (H_{BP}) получается из режекторного (H_{BR}) тем же способом, как высокочастотный фильтр из низкочастотного:

[23] В оригинале употреблен принятый термин *notch filter*, от английского «notch» — «вырез, прорезь». Авторы пользуются этим названием и для заграждающих, и для пропускающих фильтров. — *Прим. перев.*

а *б*

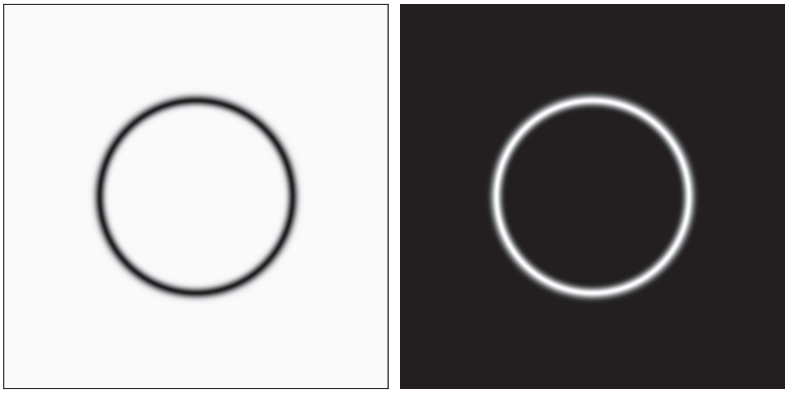


Рис. 4.63. (*а*) Режекторный гауссов фильтр. (*б*) Соответствующий полосовой пропускающий фильтр. На рисунке (*а*) для наглядности добавлена тонкая черная рамка; она не является частью данных

$$H_{\mathrm{BP}}(u,v) = 1 - H_{\mathrm{BR}}(u,v). \tag{4.10-1}$$

На рис. 4.63(*б*) в виде изображения представлен полосовой пропускающий гауссов фильтр.

4.10.2. Узкополосные фильтры

Узкополосные фильтры являются наиболее используемыми из избирательных фильтров. Такие фильтры задерживают (или пропускают) частоты в предварительно определенной окрестности центра частотного прямоугольника. Фильтры с нулевым сдвигом фазы должны быть симметричными относительно начала координат, таким образом, вырез в точке (u_0,v_0) обязан иметь симметричный вырез в точке $(-u_0,-v_0)$. Узкополосные режекторные фильтры строятся как произведение нескольких режекторных фильтров, центры которых совмещены с центром узкополосного фильтра. Основная форма следующая:

$$H_{\mathrm{NR}}(u,v) = \prod_{k=1}^{Q} H_k(u,v) H_{-k}(u,v), \tag{4.10-2}$$

где $H_k(u,v)$ и $H_{-k}(u,v)$ — режекторные фильтры с центрами в точках (u_k,v_k) и $(-u_k,-v_k)$ соответственно. Эти центры задаются относительно центра частотного прямоугольника $(M/2, N/2)$. Таким образом, вычисление расстояния от центра фильтра для каждого из фильтров осуществляется по формуле

$$D_k(u,v) = [(u - M/2 - u_k)^2 + (v - N/2 - v_k)^2]^{1/2} \tag{4.10-3}$$

и

$$D_{-k}(u,v) = [(u - M/2 + u_k)^2 + (v - N/2 + v_k)^2]^{1/2}. \tag{4.10-4}$$

Например, узкополосный режекторный фильтр Баттерворта порядка n, состоящий из трех пар вырезов, выражается формулой

$$H_{\mathrm{NR}}(u,v) = \prod_{k=1}^{3} \left[\frac{1}{1 + [D_{0k}/D_k(u,v)]^{2n}} \right] \left[\frac{1}{1 + [D_{0k}/D_{-k}(u,v)]^{2n}} \right], \tag{4.10-5}$$

где D_k и D_{-k} задаются выражениями (4.10-3) и (4.10-4). Константа D_{0k} одна и та же для каждой пары вырезов, но для различных пар может быть разной. Другие узкополосные фильтры строятся похожим образом в зависимости от выбранного исходного режекторного фильтра. Как и в случае фильтров, рассмотренных ранее, *узкополосный пропускающий фильтр* строится из узкополосного режекторного фильтра при помощи следующего выражения:

$$H_{\mathrm{NP}}(u,v) = 1 - H_{\mathrm{NR}}(u,v). \qquad (4.10\text{-}6)$$

Как показано в следующих примерах, одним из главных приложений узкополосных фильтров является выборочная модификация локальных областей ДПФ. Подобный способ обработки обычно осуществляется интерактивным образом непосредственно по ДПФ, полученному без расширения. В интерактивном случае преимущества работы с действительными координатами ДПФ (в отличие от необходимости «транслировать» расширенные значения в действительные) перевешивает все ошибки наложения (перехлеста), которые могут возникнуть из-за отсутствия расширения в процессе фильтрации. Более того, как будет показано в разделе 5.4.4, даже более мощные методы узкополосной фильтрации, чем рассмотренные здесь, основаны на нерасширенном ДПФ. Касательно вопроса изменения ДПФ в зависимости от параметров расширения см. задачу 4.22.

Пример 4.23. Подавление картины муара при помощи узкополосной фильтрации.

■ На рис. 4.64(*а*) представлено изображение сканированного газетного снимка, содержащее заметный муар, выглядящий как картина двумерного синуса; на рис. 4.64(*б*) показан его спектр. Из табл. 4.3 известно, что Фурье-преобразованием чистого синуса, являющегося периодической функцией, будет пара симметрично сопряженных импульсов. Симметричные «импульсоподобные» всплески на рис. 4.64(*б*) являются результатом почти периодичности картины муара. Эти всплески можно подавить использованием узкополосной фильтрации.

На рис. 4.64(*б*) представлен результат умножения ДПФ (рис. 4.64(*а*)) на узкополосный режекторный фильтр Баттерворта с $D_0 = 3$ и $n = 4$ для всех пар вырезов. Значение радиуса было выбрано (на основе визуального анализа спектра) так, чтобы полностью закрыть всплески энергии, а значение n выбиралось таким, чтобы обеспечить вырезам умеренно резкие переходы. Координаты центров вырезов находились интерактивно по спектру. На рис. 4.64(*г*) показан окончательный результат, полученный с подготовленным таким способом фильтром и выполненный согласно процедуре, изложенной в разделе 4.7.3. Улучшение заметно, учитывая при этом невысокое разрешение и незначительное ухудшение деталей исходного изображения. ■

Пример 4.24. Применение узкополосной фильтрации для улучшения изображения колец Сатурна, полученного зондом «Кассини».

■ На рис. 4.65(*а*) представлен снимок части колец, окружающих планету Сатурн. Изображение было получено зондом «Кассини» — первым космическим аппаратом, достигшим орбиты планеты. Вертикальная синусоидальная картина

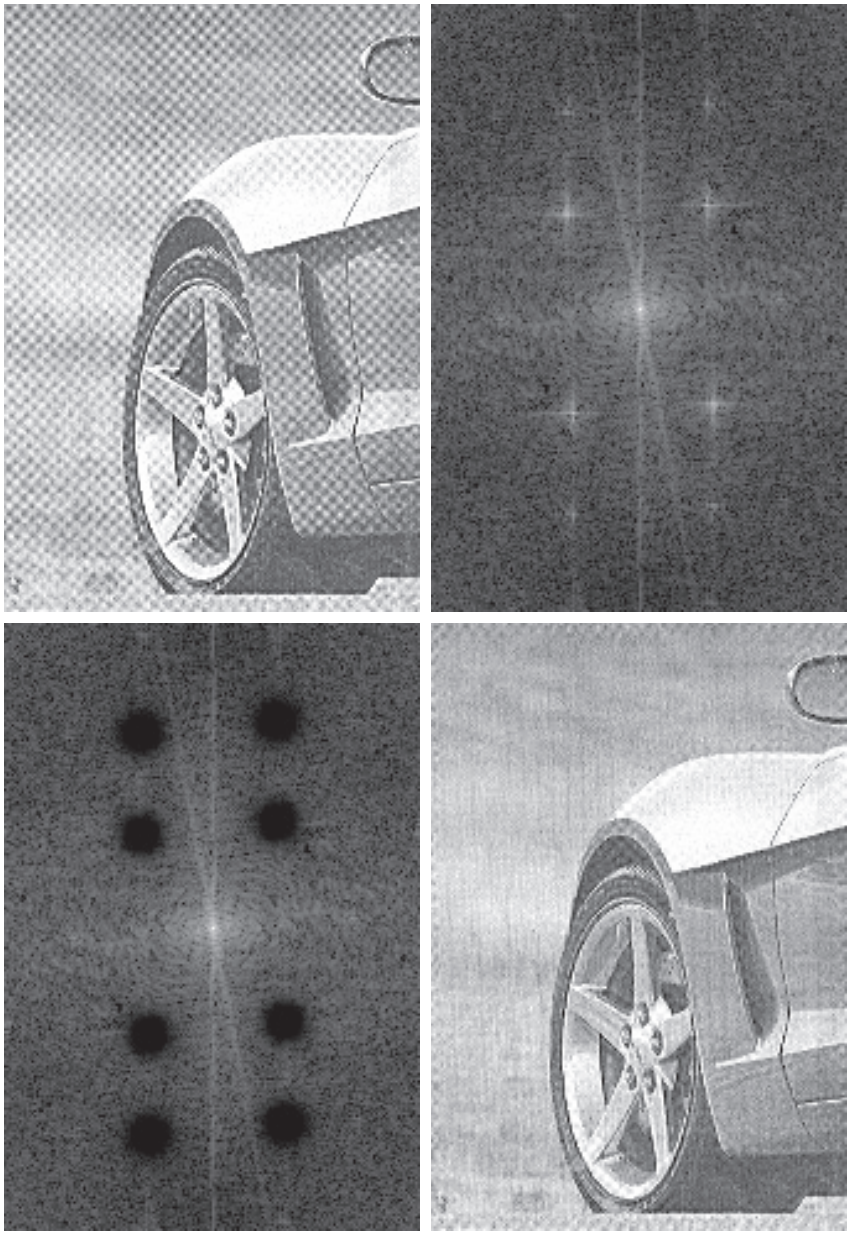
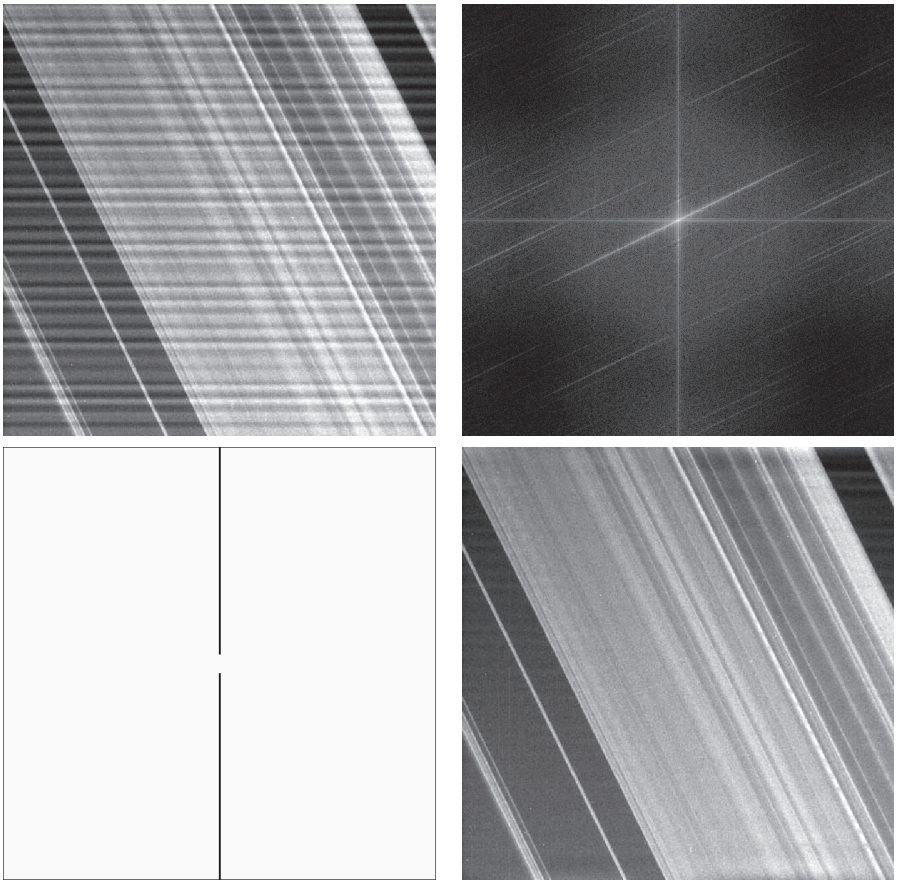


Рис. 4.64. (*а*) Изображение сканированного газетного снимка с муаром. (*б*) Спектр. (*в*) Фурье-преобразование, умноженное на узкополосный режекторный фильтр Баттерворта. (*г*) Фильтрованное изображение

возникла из-за помехи переменного тока, которая наложилась на видеосигнал с камеры непосредственно перед его оцифровкой. Это явилось неожиданной проблемой, испортившей несколько изображений программы полета. К счастью, этот тип помехи сравнительно легко скорректировать во время постобработки. Одним из подходов является использование узкополосной фильтрации.



а б
в г

Рис. 4.65. (*а*) Изображение (674×674 пикселя) колец Сатурна с почти периодической помехой. (*б*) Спектр. Всплески энергии на вертикальной оси вблизи начала координат соответствуют картине помехи. (*в*) Вертикальный узкополосный режекторный фильтр. (*г*) Результат фильтрации. Тонкая черная граница на рисунке (*в*) добавлена для ясности; она не является частью данных. (Исходное изображение предоставил д-р Р.А. Уэст, NASA/JPL

На рис. 4.65(*б*) показан Фурье-спектр. Тщательный анализ вертикального луча показал, что имеется серия маленьких всплесков энергии, которые соответствуют почти синусоидальной помехе. Простой подход состоит в использовании фильтра в виде узкого прямоугольного выреза от низшего по частоте всплеска и до конца вертикального луча. Такой фильтр показан на рис. 4.65(*в*); белый цвет представляет 1, а черный — 0. Рис. 4.65(*б*) отображает результат фильтрации искаженного изображения при помощи такого фильтра. Данный результат демонстрирует значительное улучшение по сравнению с исходным изображением.

Используя обратную версию того же фильтра — узкополосный пропускающий фильтр (рис. 4.66(*а*)), были выделены частоты в вертикальном луче. Затем

а б

Рис. 4.66. (*а*) Результат (спектр), полученный применением узкополосного пропускающего фильтра к ДПФ изображения рис. 4.65(*а*). (*б*) Пространственная картина, полученная вычислением обратного ДПФ от (*а*)

обратным ДПФ выделенных таким способом частот была получена сама пространственная картина искажений (рис. 4.66(*б*)). ■

4.11. Вопросы реализации

До сих пор внимание было сосредоточено на теоретических концепциях и примерах фильтрации в частотной области. К этому моменту уже должно быть ясно одно: вычислительные процедуры в этой области обработки изображений не являются тривиальными. Таким образом, важно получить хотя бы начальное представление о методах, с помощью которых вычисление преобразования Фурье может быть упрощено и ускорено. Настоящий раздел касается именно таких вопросов.

4.11.1. Разделимость двумерного ДПФ

Как отмечалось в табл. 4.2, двумерное ДПФ является разделимым на два одномерных преобразования. Можно записать выражение (4.5-15) в виде

$$F(u,v) = \sum_{x=0}^{M-1} e^{-i2\pi ux/M} \sum_{y=0}^{N-1} f(x,y) e^{-i2\pi vy/N} = \sum_{x=0}^{M-1} F(x,v) e^{-i2\pi ux/M}, \qquad (4.11\text{-}1)$$

где

$$F(x,v) = \sum_{y=0}^{N-1} f(x,y) e^{-i2\pi vy/N}. \qquad (4.11\text{-}2)$$

Видно, что для каждого значения x функция $F(x,v)$ ($v = 0, 1, 2, ..., N-1$) является просто одномерным ДПФ соответствующей строки двумерной функции $f(x,y)$, ($y = 0, 1, 2, ..., N-1$). Варьируя в (4.11-2) x от 0 до $M-1$, мы получаем набор ДПФ

для всех строк $f(x, y)$. Вычисления согласно (4.11-1) аналогично представляются одномерными ДПФ столбцов $F(x, v)$.

Таким образом, можно сделать вывод, что двумерное ДПФ от $f(x, y)$ может быть получено вычислением одномерных преобразований по каждой строке $f(x, y)$, а затем вычислением одномерных преобразований по каждому столбцу полученного промежуточного результата. Это является важным упрощением, поскольку в каждый момент времени нам нужно иметь дело всего только с одной переменной. Аналогичный подход применим и к вычислению двумерного обратного ДПФ при помощи одномерного обратного ДПФ. Более того, как будет показано в следующем разделе, одномерное обратное ДПФ может быть вычислено по алгоритму прямого ДПФ.

> Можно также выразить (4.11-1) и (4.11-2) в виде одномерного преобразования по столбцам, за которым следует преобразование по строкам. Конечный результат будет тем же самым.

4.11.2. Вычисление обратного ДПФ при помощи алгоритма прямого ДПФ

Взяв операцию комплексного сопряжения от обеих частей равенства (4.5-16) и умножив результат на MN, получим

$$MNf^*(x,y) = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F^*(u,v)e^{-i2\pi(ux/M+vy/N)}. \qquad (4.11\text{-}3)$$

> Умножение на MN обусловлено выбранной формой выражений (4.5-15) и (4.5-16). В случае иного распределения констант между прямым и обратным преобразованиями потребуется другая схема умножения на константу.

Правая часть этого преобразования есть не что иное, как ДПФ от $F^*(u,v)$. Таким образом, (4.11-3) показывает, что подставив $F^*(u,v)$ в алгоритм вычисления двумерного прямого преобразования Фурье, на выходе будем иметь $MNf^*(x,y)$. Вычислив комплексно сопряженные значения и разделив их на MN, получим в результате $f(x,y)$, что является обратным ДПФ от $F(u,v)$.

Вычисление двумерного обратного преобразования при помощи алгоритма двумерного прямого ДПФ, основанного на последовательности одномерных преобразований (как в предыдущем разделе) является источником частых ошибок, поскольку требуются операции комплексного сопряжения и умножения на константу, ни одна их которых не применяется в алгоритме вычисления прямого преобразования. Необходимо помнить тот основной принцип, что $F^*(u,v)$ просто подается на вход любого имеющегося алгоритма прямого ДПФ, результатом которого будет $MNf^*(x,y)$. Все, что остается сделать с этим результатом, чтобы получить $f(x,y)$, — взять комплексно сопряженные значения и разделить их на константу MN. Конечно, в случае если $f(x,y)$ — действительная, то $f^*(x,y) = f(x,y)$.

4.11.3. Быстрое преобразование Фурье

Работа в частотной области будет не слишком удобной, если вычислять операции (4.5-15) и (4.5-16) прямым образом. На выполнение этих преобразований «в лоб» требуется порядка $(MN)^2$ операций умножения и сложения. Для изображений среднего размера (скажем, 1024×1024 пикселя) это означает необходимость выполнения около 10^{12} умножений и сложений только для вычисления ДПФ, что не учитывает необходимости вычисления экспоненты, которая может быть вычислена один раз и помещена в таблицу. Такой объем вычислений затруднителен даже для суперкомпьютера. Без *быстрого преобразования Фурье* (БПФ), которое сокращает число необходимых вычислений до порядка $MN \log_2 MN$ операций умножения и сложения, можно с уверенностью сказать, что материал, изложенный в настоящей главе, мало пригоден к практическому использованию. Сокращение вычислений, обеспечиваемое БПФ, является впечатляющим. Так, вычисление двумерного БПФ изображения размерами 1024×1024 пикселей потребует всего порядка $2 \cdot 10^7$ операций умножения и сложения, что является значительным сокращением по сравнению с первоначальным числом 10^{12}, указанным выше.

Несмотря на то, что вопрос вычисления БПФ достаточно широко освещен в литературе по обработке сигналов, этот предмет настолько важен в нашей работе, что данная глава была бы неполной без рассмотрения хотя бы начального объяснения того, почему БПФ работает столь эффективно. Для достижения указанной цели мы выбрали алгоритм, называемый *методом последовательного удвоения*, который являлся оригинальным алгоритмом и привел к рождению целой индустрии. Данный алгоритм предполагает, что число отсчетов преобразуемой последовательности является степенью двойки, но это требование не является общим и существуют также другие подходы (см. [Brigham, 1988]). Из раздела 4.11.1 известно, что двумерное ДПФ может быть реализовано в виде последовательных проходов одномерных преобразований, так что достаточно сфокусироваться на вычислении БПФ только одной переменной.

При рассмотрении вывода алгоритма БПФ удобно выразить (4.4-6) в следующей форме:

$$F(u) = \sum_{x=0}^{M-1} f(x)(W_M)^{ux}, \tag{4.11-4}$$

$u = 0, 1, ..., M - 1$, где

$$W_M = e^{-i2\pi/M} \tag{4.11-5}$$

и предполагается, что M является степенью двойки, т. е. представимо в виде

$$M = 2^n, \tag{4.11-6}$$

причем n — целое положительное число. Отсюда следует, что M представимо в виде

$$M = 2K, \tag{4.11-7}$$

где K — также целое положительное число. Подставив (4.11-7) в (4.11-4), получим

$$F(u)=\sum_{x=0}^{2K-1} f(x)W_{2K}^{ux}=\sum_{x=0}^{K-1} f(2x)W_{2K}^{u(2x)}+\sum_{x=0}^{K-1} f(2x+1)W_{2K}^{u(2x+1)}. \qquad (4.11\text{-}8)$$

Используя (4.11-5), нетрудно показать, что $(W_{2K})^{2ux}=(W_K)^{ux}$, поэтому (4.11-8) может быть переписано в виде

$$F(u)=\sum_{x=0}^{K-1} f(2x)W_K^{ux}+\sum_{x=0}^{K-1} f(2x+1)W_K^{ux}W_{2K}^{u}. \qquad (4.11\text{-}9)$$

Положив по определению

$$F_{\text{even}}(u)=\sum_{x=0}^{K-1} f(2x)W_K^{ux}, \quad \text{для} \quad u=0,1,2,\ldots,K-1 \qquad (4.11\text{-}10)$$

и

$$F_{\text{odd}}(u)=\sum_{x=0}^{K-1} f(2x+1)W_K^{ux}, \quad \text{для} \quad u=0,1,2,\ldots,K-1, \qquad (4.11\text{-}11)$$

приводим выражение (4.11-9) к виду

$$F(u)=F_{\text{even}}(u)+F_{\text{odd}}(u)W_{2K}^{u}. \qquad (4.11\text{-}12)$$

Также, поскольку $(W_M)^{u+M}=(W_M)^u$ и $(W_{2M})^{u+M}=-(W_{2M})^u$, выражения (4.11-10)—(4.11-12) дают

$$F(u+K)=F_{\text{even}}(u)-F_{\text{odd}}(u)W_{2K}^{u}. \qquad (4.11\text{-}13)$$

Анализ выражений (4.11-10)—(4.11-13) обнаруживает ряд их интересных свойств. Как показывают выражения (4.11-12) и (4.11-13), M-точечное преобразование может быть вычислено с помощью разложения исходного выражения на две части. Для вычисления первой половины массива $F(u)$ нужно вычислить два $(M/2)$-точечных преобразования, заданных формулами (4.11-10) и (4.11-11). Подстановка полученных значений $F_{\text{even}}(u)$ и $F_{\text{odd}}(u)$ в (4.11-12) дает $F(u)$ для $u=0, 1, 2, ..., (M/2-1)$. Вторая половина массива прямо получается при подстановке этих значений в (4.11-13) *без* дополнительного вычисления преобразований.

Для того чтобы понять последствия применения такой процедуры с вычислительной точки зрения, обозначим через $m(n)$ и $a(n)$ соответственно число комплексных умножений и сложений, необходимых для ее выполнения. Как и ранее, полное число отсчетов равно 2^n, где n — положительное целое число. Предположим сначала, что $n=1$. Двухточечное преобразование требует вычисления значения $F(0)$; затем значение $F(1)$ может быть получено из (4.11-13). Для получения $F(0)$ сначала нужно вычислить $F_{\text{even}}(0)$ и $F_{\text{odd}}(0)$. Поскольку в данном случае $K=1$, то выражения (4.11-10) и (4.11-11) представляют собой одноточечные преобразования. Однако поскольку ДПФ единственного отсчета есть сам этот отсчет, то для получения $F_{\text{even}}(0)$ и $F_{\text{odd}}(0)$ не требуются ни умножения, ни сложения. Одно умножение $F_{\text{odd}}(0)$ на W_2^0 и одно сложение дает $F(0)$ по формуле (4.11-12). Затем $F(1)$ получается по формуле (4.11-13) при помощи еще одного сложения (вычитание рассматривается как операция аналогичная сложению). Поскольку величина $F_{\text{odd}}(0)W_2^0$ уже была сосчитана ранее, то полное число операций, необходимое для вычисления двухточечного преобразования, состоит из $m(1)=1$ умножений и $a(1)=2$ сложений.

Пусть теперь $n = 2$. В соответствии с предыдущими результатами 4-точечное преобразование может быть разделено на две части. Первая половина $F(u)$ требует вычисления двух двухточечных преобразований, задаваемых формулами (4.11-10) и (4.11-11) с $K = 2$. Как отмечено в предыдущем параграфе, вычисление двухточечного преобразования требует $m(1)$ умножений и $a(1)$ сложений, таким образом, вычисление по двум указанным формулам потребует всего $2m(1)$ умножений и $2a(1)$ сложений. Еще два сложения и умножения необходимы для получения $F(0)$ и $F(1)$ по формуле (4.11-12). Поскольку величины $F_{\text{odd}}(u)W_{2K}^{u}$ для $u = \{0,1\}$ уже были посчитаны, еще два сложения дадут $F(2)$ и $F(3)$. Таким образом, полное число операций $m(2) = 2m(1) + 2$ и $a(2) = 2a(1) + 4$.

Когда $n = 3$, вычисление $F_{\text{even}}(0)$ и $F_{\text{odd}}(0)$ сводится к вычислению двух 4-точечных преобразований. Это требует $2m(2)$ умножений и $2a(2)$ сложений. Для полного завершения вычислений необходимо еще четыре умножения и восемь сложений. Таким образом, полное число операций $m(3) = 2m(2) + 4$ и $a(3) = 2a(2) + 8$.

Продолжая рассуждать подобным образом, мы придем к следующим рекуррентным соотношениям для числа операций, необходимых для вычисления БПФ при любом положительном значении n:

$$m(n) = 2m(n-1) + 2^{n-1}, \quad n \geq 1 \tag{4.11-14}$$

и

$$a(n) = 2a(n-1) + 2^{n}, \quad n \geq 1; \tag{4.11-15}$$

кроме того, $m(0) = 0$, $a(0) = 0$, поскольку одноточечное преобразование не требует ни умножений, ни сложений.

Формулы (4.11-10)—(4.11-13) лежат в основе БПФ алгоритма с последовательным удвоением. Это название связано с методом вычисления двухточечного преобразования на основе двух одноточечных, 4-точечного — на основе двух двухточечных и т. д. для любого M, равного целой степени 2. В качестве упражнения (задача 4.41) мы предлагаем читателю показать, что

$$m(n) = \frac{1}{2} M \log_2 M \tag{4.11-16}$$

и

$$a(n) = M \log_2 M . \tag{4.11-17}$$

Преимущество в скорости, которое дает алгоритм БПФ по сравнению с методом прямого вычисления одномерного ДПФ, определяется выражением

$$C(M) = \frac{M^2}{M \log_2 M} = \frac{M}{\log_2 M}. \tag{4.11-18}$$

Поскольку предполагается, что $M = 2^n$, то (4.6-49) можно переписать относительно n:

$$C(n) = \frac{2^n}{n}. \tag{4.11-19}$$

График соответствующей функции представлен на рис. 4.67. Очевидно, что преимущество в скорости быстро возрастает с ростом числа n. Например, при

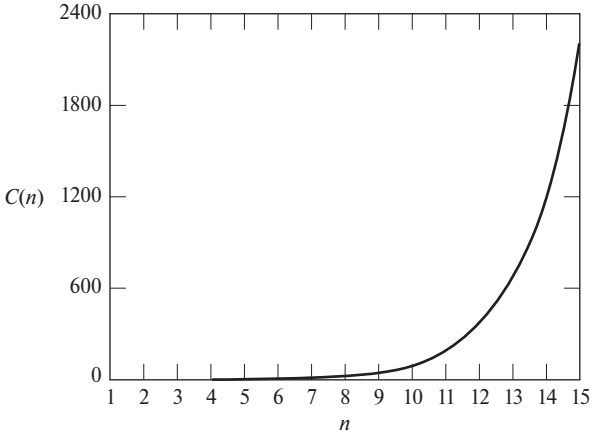


Рис. 4.67. Преимущество в скорости, которое дает алгоритм БПФ по сравнению с прямым методом вычисления одномерного ДПФ. Обратите внимание на быстрый рост этого показателя как функции от n

$n = 15$ (32768 отсчетов) БПФ превосходит в скорости метод прямого вычисления почти в 2200 раз. Поэтому можно ожидать, что на одном и том же компьютере вычисления с помощью БПФ будут совершаться почти в 2200 раз быстрее, чем прямое вычисление ДПФ.

Детальному рассмотрению БПФ посвящено так много прекрасных работ (см., например, [Brigham, 1988]), что мы не будем более останавливаться на этом вопросе. Фактически все полные пакеты программ для обработки сигналов и изображений включают в себя реализацию обобщенного БПФ алгоритма, который может также оперировать с количеством отсчетов, не равным целой степени 2 (за счет меньшей вычислительной эффективности). Бесплатные БПФ программы также легко доступны, в основном через Интернет.

4.11.4. Некоторые замечания по поводу построения фильтров

При рассмотрении в этой главе методов фильтрации в частотной области мы намеренно сосредоточили внимание на основных принципах, чтобы стала яснее суть используемых методов. Лучшего пути для этого, чем избранный способ изложения, мы не знаем. Материал главы может рассматриваться как фундамент для конструирования фильтров. Другими словами, если мы хотим подобрать фильтр для некоторой конкретной задачи, то частотные методы будут тем идеальным инструментом, который даст возможность экспериментировать быстро и с полным контролем над параметрами фильтра.

Когда для конкретного приложения уже найден фильтр, часто возникает вопрос о реализации соответствующей фильтрации непосредственно в пространственной области с использованием встроенных программ и/или аппаратных средств. Данный вопрос выходит за рамки настоящей книги. В работе [Petrou, Bosdogianni, 1999] представлена подходящая связь между двумерными пространственными фильтрами и соответствующими цифровыми фильтрами. По поводу разработки двумерных цифровых фильтров см. [Lu, Antoniou, 1992].

Заключение

Представленные в настоящей главе сведения являются последовательным изложением материала от дискретизации к преобразованию Фурье и затем до вопросов фильтрации в частотной области. Некоторые из концепций, такие как теорема отсчетов, не имеют почти никакого смысла, если не рассматриваются в контексте частотной области. То же самое справедливо и в отношении вопросов наложения спектров. Таким образом, изложенный материал служит прочной базой для понимания основ цифровой обработки сигналов. Мы специально постарались вести изложение, начиная с самых основных принципов, чтобы любой читатель с умеренным уровнем математической подготовки был способен не только впитать материал, но также и использовать его.

Второй важной целью главы было рассмотрение дискретного преобразования Фурье и вопросов его использования для фильтрации в частотной области. Для этой цели нам потребовалось рассмотреть теорему о свертке. Этот вопрос служит базой для построения линейных систем и лежит в основе многих методов восстановления, рассматриваемых в главе 5. Набор рассмотренных фильтров достаточно репрезентативен для множества, используемого на практике. Однако ключевым вопросом представления этих фильтров было стремление показать, насколько просто формулировать и реализовывать фильтры в частотной области. Хотя окончательная реализация найденного решения, как правило, базируется на пространственных фильтрах, тем не менее невозможно переоценить роль интуиции, возникающей при работе в частотной области и служащей затем руководством для выбора пространственных фильтров.

Хотя большинство примеров фильтрации в данной главе касаются вопросов улучшения изображений, тем не менее сами процедуры являются общими и широко используются в последующих главах книги.

Ссылки и литература для дальнейшего изучения

В качестве дополнительного чтения по вопросам, затронутым в разделе 4.1, мы рекомендуем [Hubbard, 1998]. Книги [Bracewell, 2000, 1995] могут служить хорошим введением в теорию непрерывного преобразования Фурье и его обобщения на двумерный случай для обработки изображений. Две упомянутые книги, а также [Lim, 1990], [Castleman, 1996], [Petrou, Bosdogianni, 1999], [Brigham, 1988] и [Smith, 2003] содержат исчерпывающее изложение вопросов, составляющих основу всех рассмотрений разделов 4.2—4.6. В качестве обзора ранних работ, касающихся вопросов картин муара, см. [Oster, Nishijima, 1963]. Уровень науки в этой области спустя тридцать лет рассмотрен в [Creath, Wyant, 1992]. Вопросы дискретизации, наложения спектров и восстановления изображений, рассмотренные в разделе 4.5, также являются предметом значительного интереса в компьютерной графике, как это иллюстрируется в [Shirley, 2002].

В качестве дополнительных основ по материалу разделов 4.7—4.11 см. [Castleman, 1996], [Pratt, 2001] и [Hall, 1979]. Для более глубокого знакомства с датчиками изображений на зонде «Кассини» (раздел 4.10.2) см. [Porco, West et al. 2004]. По-прежнему представляет интерес вопрос о том, как эффективно управлять возникающими в результате фильтрации артефактами (такими как звон); по этому поводу см. [Baker, Reeves, 2000]. По вопросам нерезкого маскирования и высокочастотной фильтрации с подъемом частотной характеристики см. [Schowengerdt, 1983]. При изложении материала, относящегося к гомоморфной фильтрации (раздел 4.9.5), мы основываемся на работе [Stockham, 1972]; см. также книги [Oppenheim, Schafer, 1975] и [Pitas, Venetsanopoulos, 1990]. В работе [Brinkman et al., 1998] методы нерезкого маскирования и гомоморфной фильтрации скомбинированы для улучшения магнитно-резонансных изображений. Вопросы конструирования цифровых фильтров (раздел 4.6.7) на основе представленного в этой главе частотного подхода рассмотрены в [Lu, Antoniou, 1992] и [Petrou, Bosdogianni, 1999].

Как указано в разделе 4.1.1, открытие быстрого преобразование Фурье (раздел 4.11.3) стало тем краеугольным камнем, на котором держится популярность ДПФ как главного инструмента для обработки сигналов. Наше изложение БПФ в разделе 4.11.3 основано на работе [Cooley, Tuckey, 1965] и на книге [Brigham, 1988], которая также содержит обсуждение ряда вопросов реализации БПФ, включая рассмотрение других оснований степени, отличных от 2. Честь открытия быстрого преобразование Фурье[24] часто приписывают Кули и Тьюки [Cooley, Tuckey, 1965]. Однако с открытием БПФ связана интересная история, достойная того, чтобы упомянуть здесь о ней. В ответ на статью Кули и Тьюки Радник опубликовал работу [Rudnick, 1966], в которой утверждал, что он использует аналогичную технику с числом операций, также пропорциональным $M \log_2 M$, которая основана на методе, изложенном в работе Даниэльсона и Ланцоша [Danielson, Lanczos, 1942]. Эти авторы в свою очередь ссылались на Рунге [Runge, 1903, 1904] как на первоисточник. Две последние работы вместе с лекциями Рунге и Кёнига [Runge, König, 1924] содержат изложение всех существенных вопросов, относящихся к вычислению БПФ. Похожие методы рассматривались также в работах [Yates, 1937], [Stumpff, 1939], [Good, 1958] и [Thomas, 1963]. В работе [Cooley, Lewis, Welch, 1967a] приведены исторический обзор и интересное сравнение результатов, предшествующих работе Кули и Тьюки 1965 г.

Алгоритм БПФ, изложенный в разделе 4.11.3, взят из оригинальной статьи Кули и Тьюки [Cooley, Tuckey, 1965]. Для дополнительного чтения см. [Brigham, 1988] и [Smith, 2003]. Касательно вопросов построения цифровых фильтров, основанных на терминах частотной области, рассматривавшихся в настоящей главе (раздел 4.11.4), см. [Lu, Antoniou, 1992] и [Petrou, Bosdogianni, 1999]. По поводу программной реализации многих приложений, рассмотренных в разделах 4.7—4.11, см. [Gonzalez, Woods, Eddins, 2004].

---

[24] Имеются сведения, что метод БПФ был предложен в 1805 году Гауссом и переоткрыт в 1965 г. — *Прим. перев.*

Задачи

Детальное рассмотрение задач, помеченных звездочкой, можно найти на интернет-сайте книги. Также там содержатся примерные проекты, базирующиеся на материалах данной главы.

4.1 Повторите пример 4.1, используя функцию $f(t) = 2A$ для $-W/4 \le t \le W/4$ и $f(t) = 0$ для остальных значений t. Объясните причину всех различий между Вашими результатами и результатами примера.

★4.2 Покажите, что $\tilde{F}(\mu)$ в выражении (4.4-2) является периодической в обоих направлениях с периодом $1/\Delta T$.

★4.3 Можно показать ([Bracewell, 2000]), что $1 \Leftrightarrow \delta(\mu)$ и $\delta(t) \Leftrightarrow 1$. Используйте первое из этих свойств, а также свойство сдвига из табл. 4.3, чтобы показать, что преобразование Фурье непрерывной функции $f(t) = \cos(2\pi nt)$, где n — действительное число, будет $F(\mu) = (1/2)[\delta(\mu+n) + \delta(\mu-n)]$.

4.4 Рассмотрите непрерывную функцию $f(t) = \cos(2\pi nt)$.

★ **(а)** Чему равен период $f(t)$?

★ **(б)** Чему равна частота $f(t)$?

Фурье-преобразование $F(\mu)$ от $f(t)$ является действительным (задача 4.3), и поскольку преобразование дискретизованных данных состоит из периодических копий $F(\mu)$, то преобразование от дискретизованной функции $\tilde{F}(\mu)$ также будет действительным. Нарисуйте диаграмму, аналогичную рис. 4.6, и ответьте на следующие вопросы, основываясь на диаграмме (предположите, что дискретизация начинается с точки $t = 0$).

★ **(в)** Как в общих чертах будут выглядеть дискретизованная функция и ее Фурье-преобразование, если $f(t)$ дискретизована с частотой, превышающей частоту Найквиста?

(г) Как в общих чертах будут выглядеть дискретизованная функция и ее Фурье-преобразование, если $f(t)$ дискретизована с частотой, меньшей частоты Найквиста?

(д) Как в общих чертах будут выглядеть дискретизованная функция, если $f(t)$ дискретизована с частотой Найквиста с отсчетами, взятыми в точках $t = 0, \Delta T, 2\Delta T, ...$?

★4.5 Докажите справедливость одномерной теоремы о свертке непрерывных переменных, сформулированной в виде соотношений (4.2-21) и (4.2-22).

4.6 Дополните последовательность действий, приводящих от (4.3-11) к (4.3-12).

4.7 Как показано на рисунке ниже, Фурье-преобразованием треугольной функции (слева) является квадрат sinc-функции (справа). Приведите довод, который докажет, что Фурье-преобразование треугольной функции может быть получено из Фурье-преобразования прямоугольной функции. (*Подсказка*: треугольная функция может быть получена как свертка двух одинаковых прямоугольных функций.)

4.8 **(а)** Покажите, что выражения (4.4-4) и (4.4-5) составляют пару преобразований Фурье.

★ **(б)** Повторите вычисления для формул (4.4-6) и (4.4-7). Вам потребуется следующее свойство ортогональности для экспонент в обеих частях данной задачи:

$$\sum_{x=0}^{M-1} e^{i2\pi rx/M} e^{-i2\pi ux/M} = \begin{cases} M & \text{если } r = u; \\ 0 & \text{в противном случае.} \end{cases}$$

4.9 Докажите справедливость равенств (4.4-8) и (4.4-9).

★**4.10** Докажите теорему о свертке для одной дискретной переменной (см. соотношения (4.2-21), (4.2-22) и (4.4-10)). Вам потребуется воспользоваться свойством сдвига $f(x)e^{i2\pi u_0 x/M} \Leftrightarrow F(u-u_0)$ и, наоборот, $f(x-x_0) \Leftrightarrow F(u)e^{-i2\pi u_0 x/M}$.

★**4.11** Напишите выражение для двумерной дискретной свертки.

4.12 Рассмотрите изображение шахматной доски, на котором каждая клетка имеет размеры 0,5×0,5 мм. Предположим, что изображение простирается в бесконечность по обеим координатам. Чему будет равна минимальная частота дискретизации (в отсчетах/мм), необходимая, чтобы избежать наложения спектров?

4.13 Из обсуждений в разделе 4.5.6 известно, что уменьшение изображения может привести к наложению спектров. Справедливо ли подобное утверждение для увеличения? Объясните.

★**4.14** Докажите, что одномерные непрерывное и дискретное преобразования Фурье являются линейными операциями (см. раздел 2.6.2 касательно определения линейности).

4.15 Имеется программа вычисления пары двумерных преобразований Фурье. Однако из ее описания неизвестно, в какое из двух преобразований (прямое или обратное) включен член $1/MN$, или же он разделен на две константы $1/\sqrt{MN}$ перед каждым из преобразований. Как можно определить, в которое из преобразований (или в оба) включен этот член?

4.16 Докажите, что спектры непрерывного и дискретного двумерных преобразований Фурье инвариантны к сдвигу, а при повороте исходного изображения поворачиваются на тот же угол.

4.17 Из задачи 4.3 можно предположить, что $1 \Leftrightarrow \delta(\mu,v)$ и $\delta(t,z) \Leftrightarrow 1$. Воспользуйтесь первым из этих свойств, а также свойством сдвига из табл. 4.3, чтобы показать, что преобразованием Фурье непрерывной функции $f(t,z) = A\cos(2\pi\mu_0 t + 2\pi v_0 z)$ будет

$$F(\mu,v) = \frac{1}{2}[\delta(\mu+\mu_0, v+v_0) + \delta(\mu-\mu_0, v-v_0)].$$

4.18 Покажите, что ДПФ дискретной функции $f(x,y) = 1$ равно

$$\Im\{1\} = \delta(u,v) = \begin{cases} MN & \text{если } u = v = 0, \\ 0 & \text{в остальных случаях.} \end{cases}$$

4.19 Покажите, что ДПФ дискретной функции $f(x,y) = \cos(2\pi u_0 x + 2\pi v_0 y)$ будет

$$F(\mu,v) = \frac{1}{2}[\delta(u+Mu_0, v+Nv_0) + \delta(u-Mu_0, v-Nv_0)],$$

где u и v являются целыми кратными M и N соответственно.

4.20 Следующие задачи связаны со свойствами, приведенными в табл. 4.1.

★ **(а)** Докажите справедливость свойства 1.

★ **(б)** Докажите справедливость свойства 3.

(в) Докажите справедливость свойства 6.

★ **(г)** Докажите справедливость свойства 7.

(д) Докажите справедливость свойства 9.

(е) Докажите справедливость свойства 10.

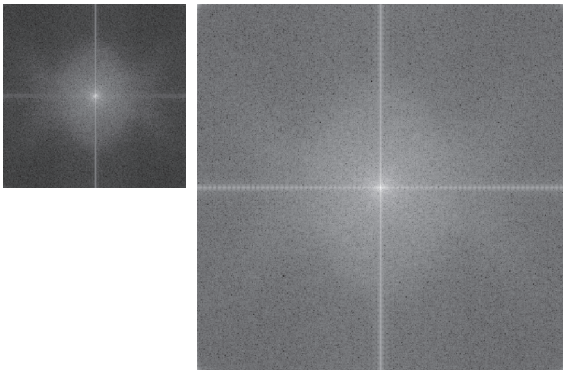
★ **(ж)** Докажите справедливость свойства 11.

(з) Докажите справедливость свойства 12.

(и) Докажите справедливость свойства 13.

★**4.21** Необходимость дополнения нулями изображения при фильтрации в частотной области детально обсуждалась в разделе 4.6.6. В разделе указывалось, что нули должны быть добавлены в конец строк и столбцов изображения (см. следующее изображение слева). Как Вы думаете, изменится ли результат, если вместо этого мы окружим изображение состоящим из нулей бордюром (см. изображение справа), не изменив общего числа нулей? Ответ объясните.

4.22 Два представленных Фурье-спектра являются спектрами одного и того же изображения. Спектр слева соответствует исходному изображению, а спектр справа получен после расширения изображения нулями. Объясните значительный рост уровня сигнала вдоль вертикальной и горизонтальной осей в правом спектре.



4.23 Как указано в табл. 4.2, $F(0,0)$ — постоянная составляющая ДПФ — пропорциональна величине средней яркости соответствующего пространственного изображения. Пусть изображение имеет размеры $M \times N$ и оно расши-

рено нулями до размера $P \times Q$, где P и Q заданы выражениями (4.6-31) и (4.6-32). Пусть $F_p(0,0)$ — постоянная составляющая ДПФ расширенного изображения.

★ **(а)** Каково соотношение средних яркостей исходного и расширенного изображений?

(б) Верно ли равенство $F_p(0,0) = F(0,0)$? Подтвердите свой ответ математически.

4.24 Докажите свойства периодичности (строка 8 в табл. 4.2).

4.25 Следующие задачи связаны с табл. 4.3.

★ **(а)** Докажите справедливость дискретной теоремы о свертке (строка 6) для одномерного случая.

(б) Повторите (а) для двумерного случая.

(в) Докажите страведливость выражения в строке 9.

(г) Докажите справедливость выражения в строке 13.

(Замечание: задачи 4.18, 4.19 и 4.31 также связаны с табл. 4.3.)

4.26 **(а)** Покажите, что лапласиан непрерывной функции $f(t,z)$ непрерывных переменных t и z удовлетворяет следующему выражению Фурье-пары (см. уравнение (3.6-3) для определения лапласиана):

$$\nabla^2 f(t,z) \Leftrightarrow -4\pi^2(\mu^2 + v^2)F(\mu,v).$$

(Совет: проанализируйте строку 12 табл. 4.3 и см. задачу 4.25(*г*).)

★ **(б)** Предыдущее аналитическое выражение справедливо лишь для непрерывных переменных. Однако оно может являться основой для реализации лапласиана в дискретной частотной области при использовании фильтра $M \times N$

$$H(u,v) = -4\pi^2(u^2 + v^2)$$

для $u = 0, 1, 2, ..., M - 1$ и $v = 0, 1, 2, ..., N - 1$. Объясните, как следует реализовывать такой фильтр.

(в) Как видно из примера 4.20, результат действия лапласиана в частотной области аналогичен результату действия пространственной маски с центральным коэффициентом, равным −8. Объясните, по какой причине результат лапласиана в частотной области не похож на результат действия пространственной маски с центральным коэффициентом −4. См. раздел 3.6.2 касательно лапласиана в пространственной области.

★**4.27** Рассмотрите пространственную маску размерами 5×5, которая усредняет 12 ближайших соседей точки (x,y), но саму центральную точку не включает.

(а) Найдите эквивалентный фильтр $H(u,v)$ в частотной области.

(б) Покажите, что результатом является низкочастотный фильтр.

4.28 Основываясь на выражении (3.6-4), один из подходов вычисления производных второго порядка для двумерного дискретного случая заключается в вычислении разностей вида $f(x+1, y) + f(x-1, y) - 2f(x,y)$ и $f(x, y+1) + f(x, y-1) - 2f(x,y)$.

(а) Найдите эквивалентный фильтр $H(u,v)$ в частотной области.

(б) Покажите, что результатом является высокочастотный фильтр.

4.29 Найдите эквивалентный фильтр $H(u,v)$, который реализует в частотной области пространственную операцию, выполняемую маской лапласиана на рис. 3.37(*б*).

★4.30 Подумайте о возможности применения Фурье-преобразования в вычислении (или частичном вычислении) модуля градиента (выражение (3.6-11)) для дифференцирования изображения. Если это возможно, предложите метод. Если нет, объясните почему.

4.31 Непрерывный гауссов низкочастотный фильтр в непрерывной частотной области имеет передаточную функцию

$$H(\mu, v) = e^{-(\mu^2+v^2)}.$$

Покажите, что соответствующим фильтром в пространственной области является

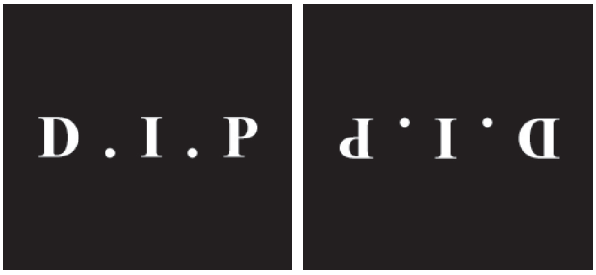
$$h(t, z) = \pi e^{-\pi(t^2+z^2)}.$$

4.32 Согласно (4.9-1) можно получить передаточную функцию высокочастотного фильтра $H_{\text{НР}}$ из передаточной функции низкочастотного фильтра при помощи выражения

$$H_{\text{НР}} = 1 - H_{\text{LP}}.$$

Используя условия задачи 4.31, определите форму *пространственной маски* гауссового фильтра высоких частот.

4.33 Рассмотрим представленные изображения. Изображение справа получено следующим образом. (а) Изображение слева умножено на $(-1)^{x+y}$; (б) вычислено ДПФ; (в) взято комплексное сопряжение Фурье-образа; (г) вычислено обратное ДПФ; (д) вещественная часть полученного результата умножена на $(-1)^{x+y}$. Объясните (математически), почему полученное изображение выглядит так, как на рисунке справа.



4.34 В чем причина возникновения почти периодических ярких точек на горизонтальной оси рис. 4.41(*б*)?

★4.35 Каждый из фильтров на рис. 4.53 имеет сильный пик в центре. Объясните причину этих пиков.

4.36 Рассмотрим представленные изображения. Изображение справа получено в результате низкочастотной фильтрации с помощью гауссова фильтра низких частот и последующего применения к результату высокочастотной фильтрации с помощью гауссова фильтра высоких частот. Размер изображений 420×344, и $D_0 = 25$ для каждого из фильтров.

(а) Объясните, почему центральная часть кольца выглядит как сплошная яркая область, в то время как превалирующая часть изображения после фильтрации состоит из контуров вдоль внешней границы объектов (например костей пальцев и запястья) с темными областями

посередине. Другими словами, разве не следует ожидать потемнения области постоянной яркости внутри кольца в результате высокочастотной фильтрации, коль скоро высокочастотные фильтры уничтожают постоянную составляющую изображения?

(б) Как Вы думаете, изменится ли результат, если поменять порядок применения фильтров на противоположный?

(Исходное изображение предоставил д-р Томас Р. Гест, отделение анатомии медицинской школы университета шт. Мичиган.)

4.37 Пусть дано изображение размером $M \times N$ и проводится эксперимент, состоящий в последовательном применении к изображению процедур низкочастотной фильтрации с использованием одного и того же гауссова фильтра низких частот с заданной частотой среза D_0. Ошибкой округления можно пренебречь. Пусть $k_{\min}$ — наименьшее положительное число, представимое в вычислительной машине, на которой проводится эксперимент.

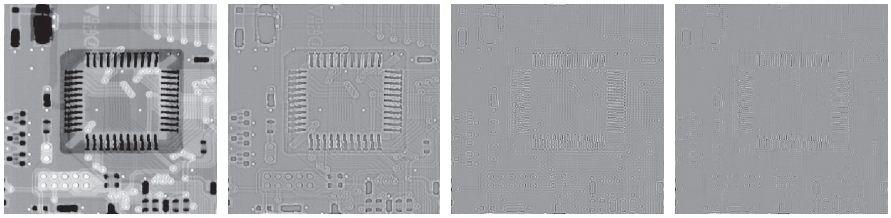
★ **(а)** Обозначим через K число произведенных в эксперименте процедур фильтрации. Можете ли Вы предсказать без проведения эксперимента, каков будет его результат (изображение) при достаточно большом значении K?

(б) Получите выражение для *минимального* значения K, гарантирующего получение предсказанного результата.

4.38 Рассмотрим представленную последовательность изображений. Изображение слева представляет собой фрагмент рентгеновского снимка промышленной печатной платы. Следующие за ним изображения суть результаты 1-кратного, 10-кратного и 100-кратного повторения процедуры фильтрации с использованием гауссова высокочастотного фильтра с $D_0 = 30$. Размер изображений — 330×334, диапазон яркости — 8 бит на элемент. Перед воспроизведением изображения были подвергнуты масштабированию, но это не сказывается на постановке задачи.

(а) Вид изображений наводит на мысль, что после некоторого количества повторений результаты перестают меняться. Установите, имеет ли в действительности это место в рассматриваемом случае. Ошибкой округления можно пренебречь. Пусть $k_{\min}$ обозначает наименьшее положительное число, представимое в вычислительной машине, на которой проводится предлагаемый эксперимент.

(б) Если при решении части (а) задачи установлено, что изменения прекратятся после некоторого числа итераций, то найдите минимальное значение этого числа.



(Исходное изображение предоставил Джозеф Е. Пассенте, Lixi, Inc.)

4.39 Как показано на рис. 4.59, комбинирование фильтрации с усилением высоких частот и эквализации гистограммы является эффективным способом для повышения резкости и улучшения контраста.

(а) Покажите, имеет ли значение очередность применения этих двух процессов.

(б) Если имеет, дайте обоснование применения первым одного или другого процесса.

4.40 Используйте гауссов высокочастотный фильтр для построения гомоморфного фильтра, который имел бы ту же общую форму, что и фильтр на рис. 4.61.

★4.41 Докажите справедливость равенств (4.11-16) и (4.11-17). (*Указание*: используйте метод математической индукции.)

4.42 Предположим, что имеется набор изображений, полученных в результате астрономических наблюдений. Каждое изображение состоит из множества ярких широко разбросанных точек, соответствующих звездам в малонаселенной части Вселенной. Проблема заключается в том, что звезды едва различимы на фоне дополнительного освещения, возникающего в результате рассеяния света в атмосфере. Исходя из модели, согласно которой изображения представляют собой суперпозицию постоянной яркостной составляющей и множества импульсов, предложите основанную на гомоморфной фильтрации процедуру для выявления составляющих изображения, которые непосредственно связаны со звездами.

4.43 Опытному медицинскому эксперту поручено просмотреть некоторую группу изображений, полученных при помощи электронного микроскопа. Для того чтобы облегчить себе задачу, эксперт решает воспользоваться методами цифровой обработки изображений. С этой целью он исследует ряд характерных изображений и сталкивается со следующими трудностями. (1) Наличие на изображениях отдельных ярких точек, не представляющих интерес. (2) Недостаточная резкость изображений. (3) Недостаточный уровень контрастности некоторых изображений. И, наконец, (4) сдвиг среднего уровня яркости, который для корректного проведения некоторых измерений яркости должен принимать значение V. Эксперт хочет преодолеть эти трудности и затем выделить белым все точки изображения, яркость которых находится в диапазоне от I_1 до I_2, сохранив яркость всех остальных точек без изменения. Предложите последовательность шагов обработки, придерживаясь которой эксперт достигнет поставленных целей. Вы можете использовать как методы главы 3, так и методы главы 4.

ГЛАВА 5

ВОССТАНОВЛЕНИЕ И РЕКОНСТРУКЦИЯ ИЗОБРАЖЕНИЙ

Вещи, которые мы созерцаем, сами по себе не таковы, как мы их созерцаем... Каковы предметы сами по себе и обособленно от восприимчивости нашей чувственности, нам совершенно неизвестно. Мы не знаем ничего, кроме свойственного нам способа воспринимать их.

Иммануил Кант

Введение

Как и при улучшении изображений, основной целью восстановления является повышение качества изображения в некотором заранее предопределенном смысле. Несмотря на пересечение областей применения методов обоих классов, улучшение изображений является в большей степени субъективной процедурой, в то время как процесс восстановления имеет в основном объективный характер. При восстановлении делается попытка реконструировать или воссоздать изображение, которое было до этого искажено, используя априорную информацию о явлении, которое вызвало ухудшение изображения. Поэтому методы восстановления основаны на моделировании процессов искажения и применении обратных процедур для воссоздания исходного изображения.

Этот подход обычно включает разработку критериев качества, которые дают возможность объективно оценить полученный результат. Напротив, методы улучшения изображений в основном представляют собой эвристические процедуры, предназначенные для такого воздействия на изображение, которое позволит затем использовать преимущества, связанные с психофизическими особенностями зрительной системы человека. Например, процедура усиления контраста рассматривается как метод улучшения, поскольку в результате ее применения изображение в первую очередь становится более приятным для глаза, тогда как процедура обработки смазанного изображения, основанная на применении обратного оператора, рассматривается как метод восстановления.

Материал этой главы носит сугубо вводный характер. Задача восстановления рассматривается лишь с момента получения уже искаженного *цифрового* изображения; поэтому вопросы, касающиеся природы искажений, вносимых чувствительными элементами, цифровыми преобразователями и воспроизводящими устройствами, затрагиваются лишь поверхностно. Эти вопросы, не-

смотря на их важность в общем контексте применения методов восстановления изображений, находятся за рамками настоящего обсуждения.

Как обсуждается в главах 3 и 4, некоторые методы восстановления удобно формулируются в пространственной области, в то время как для формулировки других больше подходит частотная область. Например, пространственная обработка применима в случае, когда единственным источником искажений является аддитивный шум. С другой стороны, задача восстановления смазанных изображений, например, трудно поддается решению в пространственной области с использованием масок фильтров малого размера. В этом случае правильным подходом является использование частотных фильтров, полученных на основе различных критериев оптимальности; такие фильтры учитывают также и наличие шума. Так же как и в главе 4, некоторый фильтр восстановления, решающий конкретную задачу в частотной области, часто используется в качестве основы для построения другого фильтра, более удобного для реализации вычислительных процедур, использующих программно-аппаратных средства.

В разделе 5.1 вводится линейная модель процесса искажения/восстановления изображения. Раздел 5.2 служит рассмотрению различных моделей шума, часто встречающихся на практике. В разделе 5.3 разрабатываются несколько методов пространственной фильтрации для уменьшения уровня шума на изображении, — процесс, часто называемый *подавлением шума на изображении*. Раздел 5.4 отведен частотным методам сокращения шума. В разделе 5.5 вводятся линейные, пространственно-инвариантные модели искажения изображения, раздел 5.6 отведен методам оценивания искажающих функций. Разделы с 5.7 по 5.10 содержат рассмотрение основных подходов к восстановлению изображения. Глава завершается (раздел 5.11) введением в задачу реконструкции изображения по проекциям. Основное применение этой концепции — компьютерная томография (КТ), являющаяся коммерчески наиболее важным приложением обработки изображений в здравоохранении.

5.1. Модель процесса искажения/восстановления изображения

Как показано на рис. 5.1, принятая в этой главе модель процесса искажения предполагает действие некоторого искажающего оператора H на исходное изображение $f(x,y)$, что после добавления аддитивного шума дает искаженное изображение $g(x,y)$. Задача восстановления состоит в построении некоторого приближения $\hat{f}(x,y)$ исходного изображения по заданному (искаженному) изображению $g(x,y)$, некоторой информации относительно искажающего оператора H и некоторой информации относительно аддитивного шума $\eta(x,y)$. Мы хотим, чтобы наше приближение было как можно ближе к исходному изображению, и, в принципе, чем больше мы знаем об операторе H и о функции η, тем ближе будет функция $\hat{f}(x,y)$ к функции $f(x,y)$. В основе подхода, применяемого на протяжении большей части главы, лежит использование операторов (фильтров), восстанавливающих изображение.

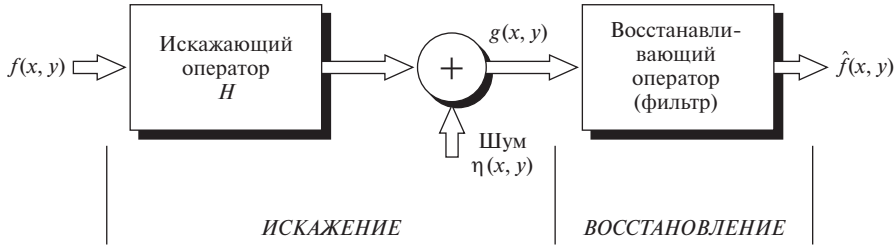


Рис. 5.1. Модель процесса искажения/восстановления изображения

В разделе 5.5 будет показано, что если H — линейный трансляционно-инвариантный оператор, то искаженное изображение может быть представлено в пространственной области в виде

$$g(x,y) = h(x,y) \star f(x,y) + \eta(x,y), \tag{5.1-1}$$

где $h(x,y)$ — функция, представляющая искажающий оператор в пространственной области[1], а символ «★» используется для обозначения пространственной свертки, как и в главе 4. Из материала раздела 4.6.6 нам известно, что свертка в пространственной области аналогична умножению в частотной области, поэтому задающее модель равенство (5.1-1) может быть эквивалентным образом записано в частотной области:

$$G(u,v) = H(u,v)F(u,v) + N(u,v), \tag{5.1-2}$$

где обозначенные заглавными буквами функции суть Фурье-образы соответствующих функций в (5.1-1). Эти два равенства составляют основу для большей части материала настоящей главы, касающихся восстановления.

В следующих трех разделах предполагается, что H — тождественный оператор, и мы имеем дело только с искажениями, вызванными наличием шума. Начиная с раздела 5.6 мы рассматриваем ряд важных искажающих операторов и некоторые методы восстановления при наличии как искажающего оператора H, так и шума η.

5.2. Модели шума

Основные источники шума на цифровом изображении — это сам процесс его получения, а также процесс передачи. Работа сенсоров зависит от различных факторов, таких как внешние условия в процессе видеосъемки и качество сенсоров. Например, в процессе получения изображения с помощью фотокамеры с ПЗС-матрицей основными факторами, влияющими на величину шума, являются уровень освещенности и температура сенсоров. В процессе передачи изображения могут искажаться помехами, возникающими в каналах связи. Например, при передаче изображения с использованием беспроводной связи оно может быть искажено в результате разряда молнии или других возмущений в атмосфере.

---

[1] Функция $h(x, y)$ называется ядром оператора H. — *Прим. перев.*

5.2.1. Пространственные и частотные свойства шума

Для дальнейшего рассмотрения важными являются параметры, определяющие пространственные характеристики шума, а также вопрос, коррелирует ли шум с изображением. Под частотными характеристиками понимаются свойства спектра шума в смысле преобразования Фурье (в отличие от частот электромагнитного спектра). Например, шум, спектр которого является постоянной величиной, называется обычно *белым шумом*. Происхождение этого термина связано с физическими свойствами белого света, который содержит практически все частоты видимого спектра в равных пропорциях. Исходя из рассмотрения главы 4, легко показать, что Фурье-спектр функции, содержащей все частоты в равных пропорциях, является постоянной величиной.

За исключением периодического в пространстве шума (раздел 5.2.3), в этой главе мы предполагаем, что шум не зависит от пространственных координат и не коррелирует с самим изображением (т. е. между значениями элементов изображения и значениями шумовой составляющей нет корреляции). Хотя в ряде случаев такие предположения по меньшей мере не вполне справедливы (примером чего могут служить изображения, полученные в ситуации с небольшим числом квантов, например рентгеновские и ЯМР-изображения), трудности, возникающие при рассмотрении пространственно-зависимого и коррелированного шума, лежат за пределами нашего обсуждения.

5.2.2. Функции плотности распределения вероятностей для некоторых важных типов шума

В рамках сделанных в предыдущем параграфе предположений мы будем иметь дело с описанием поведения шума в *пространственной области*, которое основано на статистических свойствах значений интенсивности компоненты шума в модели на рис. 5.1. Эти значения яркости могут рассматриваться как случайные величины, характеризующиеся функцией плотности распределения вероятностей. Ниже даны примеры функций плотности распределения вероятностей (probability density function, PDF), которые наиболее часто встречаются в приложениях, связанных с обработкой изображений.

По вопросам основ теории вероятности обратитесь к интернет-сайту книги.

Гауссов шум

Математическая простота, характерная для работы с моделями гауссова шума (также называемого *нормальным шумом*) как в пространственной, так и в частотной области, обусловила широкое распространение этих моделей на практике. На самом деле эта простота оказывается столь привлекательной, что зачастую гауссовы модели используются даже в тех ситуациях, когда их применение оправдано в лучшем случае лишь частично.

Функция плотности распределения вероятностей гауссовой случайной величины z задается выражением

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\bar{z})^2/2\sigma^2}, \qquad (5.2\text{-}1)$$

где z представляет собой значение яркости, $\bar{z}$ — среднее[2] значение случайной величины z, σ — ее среднеквадратическое отклонение. Квадрат среднеквадратического отклонения σ^2 называется *дисперсией* величины z. График этой функции представлен на рис. 5.2(*а*). Когда плотность распределения случайной величины z описывается функцией (5.2-1), то приблизительно 70 % ее значений попадают в диапазон $(\bar{z}-\sigma, \bar{z}+\sigma)$ и примерно 95 % — в диапазон $(\bar{z}-2\sigma, \bar{z}+2\sigma)$.

Шум Рэлея

Функция плотности распределения вероятностей шума Рэлея задается выражением

$$p(z)=\begin{cases}\dfrac{2}{b}(z-a)e^{-(z-a)^2/b} & \text{при } z \geq a \\ 0 & \text{при } z < a.\end{cases} \tag{5.2-2}$$

Среднее и дисперсия для этого распределения имеют вид

$$\bar{z}=a+\sqrt{\pi b/4}\,, \tag{5.2-3}$$

и

$$\sigma^2=\frac{b(4-\pi)}{4}. \tag{5.2-4}$$

График плотности распределения вероятностей шума Рэлея представлен на рис. 5.2(*б*). Обратите внимание на местоположение начала координат и на

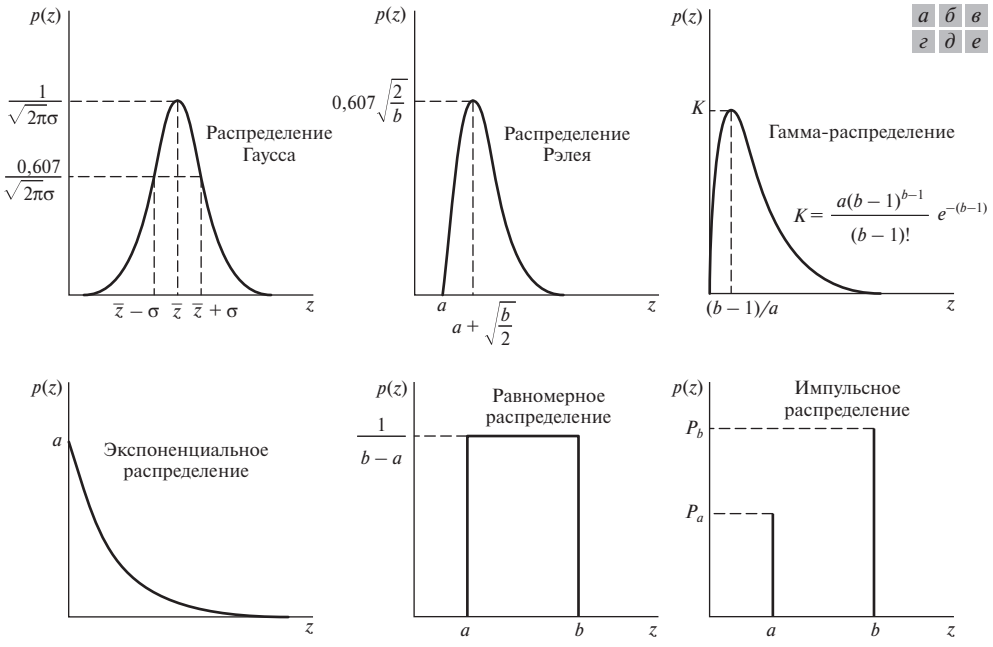


Рис. 5.2. Некоторые важные функции плотности распределения вероятностей

[2] Далее для обозначения среднего используется символ $\bar{z}$ вместо m, чтобы устранить возможную путаницу с размерами окрестности m и n.

то обстоятельство, что график имеет асимметричную (перекошенную вправо) форму. Распределение Рэлея бывает полезно для приближения асимметричных гистограмм.

Шум Эрланга (гамма-шум)

Функция плотности распределения вероятностей шума Эрланга задается выражением

$$p(z)=\begin{cases}\dfrac{a^b z^{b-1}}{(b-1)!}e^{-az} & \text{при } z\geq 0\\ 0 & \text{при } z<0.\end{cases} \tag{5.2-5}$$

где $a > 0$, b — положительное целое число и символ «!» обозначает факториал. Среднее и дисперсия для этого распределения имеют вид

$$\bar{z}=\frac{b}{a} \tag{5.2-6}$$

и

$$\sigma^2=\frac{b}{a^2}. \tag{5.2-7}$$

На рис. 5.2(*в*) представлен график плотности этого распределения. Выражение (5.2-5) часто называют *гамма-распределением*, хотя, строго говоря, это название относится к распределению более общего вида, когда b не является целым, а в знаменателе стоит гамма-функция $\Gamma(b)$. Рассматриваемый частный случай правильнее называть *распределением Эрланга*.

Экспоненциальный шум

Функция плотности распределения вероятностей *экспоненциального* шума задается выражением

$$p(z)=\begin{cases}ae^{-az} & \text{при } z\geq 0\\ 0 & \text{при } z<0.\end{cases} \tag{5.2-8}$$

где $a > 0$. Среднее и дисперсия для этого распределения имеют вид

$$\bar{z}=\frac{1}{a} \tag{5.2-9}$$

и

$$\sigma^2=\frac{1}{a^2}. \tag{5.2-10}$$

Заметим, что это распределение является частным случаем распределения Эрланга с $b = 1$. На рис. 5.2(*г*) представлен график плотности этого распределения.

Равномерный шум

Функция плотности распределения вероятностей *равномерного* шума задается выражением

$$p(z)=\begin{cases}\dfrac{1}{b-a} & \text{при } a\leq z\leq b;\\ 0 & \text{в остальных случаях.}\end{cases} \tag{5.2-11}$$

Среднее значение для этого распределения равно

$$\bar{z} = \frac{a+b}{2}, \tag{5.2-12}$$

а дисперсия

$$\sigma^2 = \frac{(b-a)^2}{12}. \tag{5.2-13}$$

На рис. 5.2(*д*) представлен график плотности этого распределения.

Импульсный шум

Функция плотности распределения вероятностей (*биполярного*) *импульсного* шума задается выражением[3]

$$p(z) = \begin{cases} P_a & \text{при } z = a; \\ P_b & \text{при } z = b; \\ 1 - P_a - P_d & \text{в остальных случаях}. \end{cases} \tag{5.2-14}$$

Если $b > a$, то пиксель с яркостью b выглядит как светлая точка на изображении. Пиксель с яркостью a выглядит, наоборот, как темная точка. Если одно из значений вероятности (P_a или P_b) равно нулю, то импульсный шум называется *униполярным*. Если ни одна из вероятностей не равна нулю и в особенности если они приблизительно равны по величине, импульсный шум походит на крупицы соли и перца, случайно рассыпанные по изображению. По этой причине импульсный шум называют также шумом типа «*соль и перец*». Мы будем использовать оба названия как взаимозаменяемые[4]. Также для обозначения этого типа шумов используются термины *шум выпадения* и *пиковый шум*.

Значения импульсов шума могут быть как положительные, так и отрицательные. При оцифровке изображения обычно происходит масштабирование (и ограничение) значений яркости. Поскольку величина связанных с импульсным шумом искажений, как правило, велика по сравнению с величиной полезного сигнала, импульсный шум после оцифровки принимает экстремальные значения, что соответствует появлению абсолютно черных и белых точек на изображении. Поэтому обычно предполагается, что значения a и b являются «интенсивными» в том смысле, что они равны минимальному и максимальному значениям, которые в принципе могут присутствовать в оцифрованном изображении. В результате отрицательные импульсы выглядят как черные точки на изображении (перец). По тем же причинам положительные импульсы выглядят как белые точки (соль). Для 8-битовых изображений это обычно означает, что $a = 0$ (черное) и $b = 255$ (белое). На рис. 5.2(*е*) представлен график функции плотности вероятностей значений импульсного шума.

---

[3] Это выражение для самой вероятности $P(z)$. Плотность распределения вероятностей может быть записана в виде $P_a\delta(z-a) + P_b\delta(z-b)$, где $\delta(\cdot)$ — δ-функция. Как правило, процесс добавления импульсного шума к изображению состоит в том, что значение яркости каждой точки изображения с вероятностью $P_s = P_a + P_b \le 1$ *заменяется* на случайное значение шума. Поэтому для биполярного импульсного шума, например, яркость изображения с вероятностью P_a заменяется на значение a, с вероятностью P_b — на значение b и с вероятностью $(1 - P_s)$ остается неизменной. — *Прим. перев.*

[4] Английский термин «соль и перец» (salt-and-pepper) практически не употребляется в русскоязычной литературе. — *Прим. перев.*

Рассмотренные распределения в совокупности представляют собой набор средств, которые позволяют моделировать искажения, связанные с широким диапазоном встречающихся на практике шумов. Так например, гауссов шум возникает на изображении в результате воздействия таких факторов, как шум в электронных цепях, а шум сенсоров — из-за недостатка освещения и/или высокой температуры. Распределение Рэлея полезно при моделировании шума, который возникает на снимках, снятых с большого расстояния. Экспоненциальное и гамма-распределения отвечают шуму на изображениях, получаемых с использованием лазеров. С импульсным шумом мы сталкиваемся в ситуациях, когда во время оцифровки изображения из-за помех в сети питания возникают переходные процессы, приводящие к появлению экстремальных значений, о которых говорилось в предыдущем абзаце. Равномерное распределение, пожалуй, в наименьшей степени подходит для описания встречающихся на практике явлений. Однако это распределение весьма полезно как основа для создания различных генераторов случайных чисел, используемых при моделировании [Peebles, 1993] и [Gonzalez, Woods, Eddins, 2004].

Пример 5.1. Изображения с шумом и их гистограммы.

■ На рис. 5.3 представлено тестовое изображение, весьма подходящее для иллюстрации только что рассмотренных моделей шума. Изображение удобно тем, что оно состоит из простых областей постоянной яркости, которая принимает всего три значения и при этом охватывает весь диапазон от черного до почти белого. Это упрощает визуальный анализ свойств различных шумовых составляющих, добавляемых к изображению.

На рис. 5.4 представлены изображения, полученные в результате добавления к тестовому изображению шумов тех шести типов, которые обсуждались выше. Под каждым изображением приведена гистограмма, посчитанная по этому изображению. Параметры шума были подобраны в каждом отдельном случае таким образом, чтобы части гистограммы, соответствующие трем уровням яркости на тестовом изображении, начали сливаться. При этом шум хорошо заметен, но не затеняет основную структуру тестового изображения.

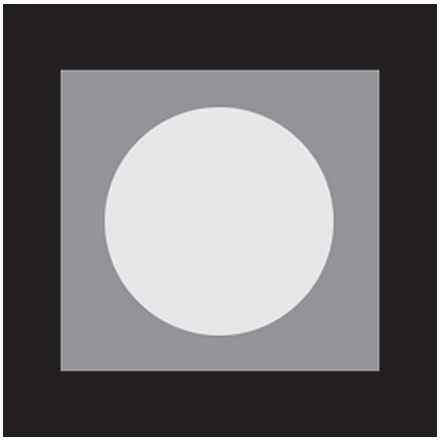


Рис. 5.3. Тестовое изображение, используемое при иллюстрации свойств шумов, распределенных в соответствии с функциями на рис. 5.2

При сравнении гистограмм на рис. 5.4 с функциями плотности распределения вероятностей на рис. 5.2 мы убеждаемся в том, что они соответствуют друг другу. В примере с импульсным шумом на гистограмме имеется дополнительный пик в «белом» конце диапазона яркостей, поскольку составляющие шума принимают значения, соответствующие абсолютно черным и абсолютно белым точкам, а наиболее яркая часть тестового изображения (внутри круга) является светло-серой. За исключением небольшого различия в общем уровне яркости, первые пять изображений на рис. 5.4 визуально мало различимы, хотя их гистограммы существенно отличаются друг от друга. Только внешний вид изображения, искаженного импульсным шумом, указывает на тип шума, который привел к искажению. ■

5.2.3. Периодический шум

Причиной появления периодического шума обычно являются электрические или электромеханические помехи во время получения изображения. Периодический шум является тем единственным видом пространственно-зависимого шума, который мы будем рассматривать в этой главе. Как обсуждается в разделе 5.4, такой шум может быть существенно уменьшен при помощи частотной фильтрации. Рассмотрим, например, изображение на рис. 5.5(*а*). Это изображение сильно искажено пространственным синусоидальным шумом различных частот. Преоб-

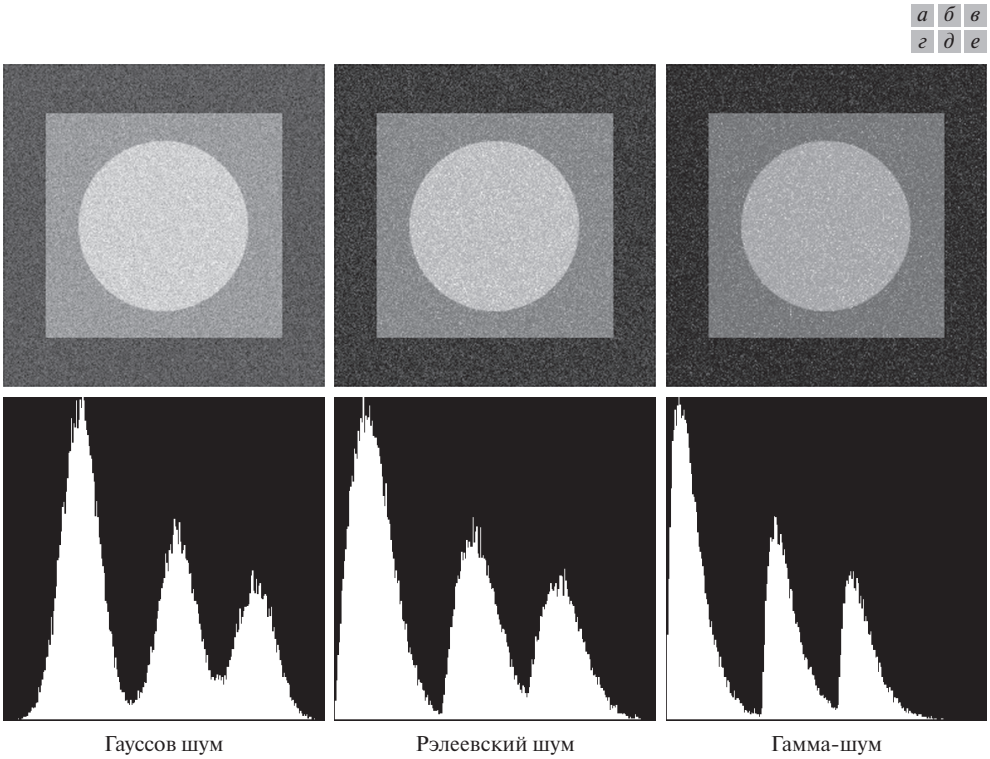
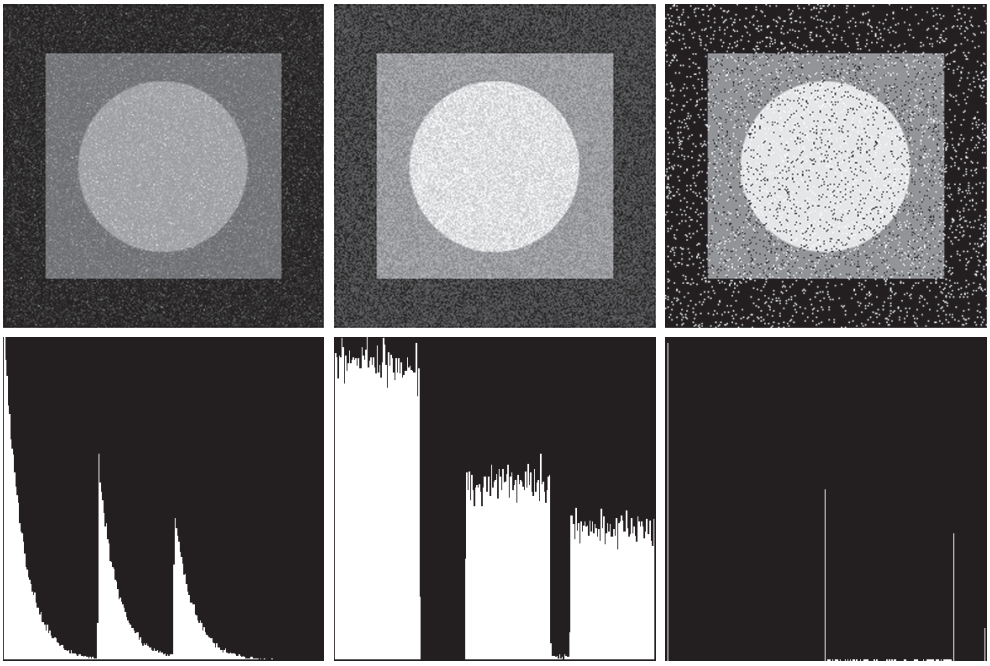


Рис. 5.4. Изображения и гистограммы, полученные в результате добавления гауссова, рэлеевского и гамма-шума к изображению на рис. 5.3

ж з и
к л м



Экспоненциальный шум Равномерный шум Импульсный шум

Рис. 5.4. (Продолжение.) Изображения и гистограммы, полученные в результате добавления экспоненциального, равномерного и импульсного шума к изображению на рис. 5.3

разование Фурье синусоиды в чистом виде представляет собой пару сопряженных импульсов[5], расположенных в центрально-симметричных точках частотной области, которые отвечают частотам синусоидальной волны (см. табл. 4.3). Поэтому если амплитуды синусоидальных волн в пространственной области достаточно велики, можно ожидать, что в спектре изображения будут видны пары импульсов, по одной для каждой синусоидальной волны в исходном изображении. Рис. 5.5(*б*) показывает, что это имеет место в действительности, причем в данном случае соответствующие импульсы расположены приблизительно на окружности, что связано с конкретным набором значений частот. В разделе 5.4 этот и другие примеры периодического шума будут рассмотрены значительно подробнее.

5.2.4. Построение оценок для параметров шума

Оценка параметров периодического шума обычно осуществляется путем анализа Фурье-спектра изображения. Как указано в предыдущем параграфе, периодический шум приводит к появлению пиков в частотной области, которые часто обнаруживаются даже при визуальном анализе. Другой подход состоит в том, что-

[5] Следует быть осторожным, дабы не путать термин *импульс* в частотной области с употреблением того же термина в импульсном шуме.

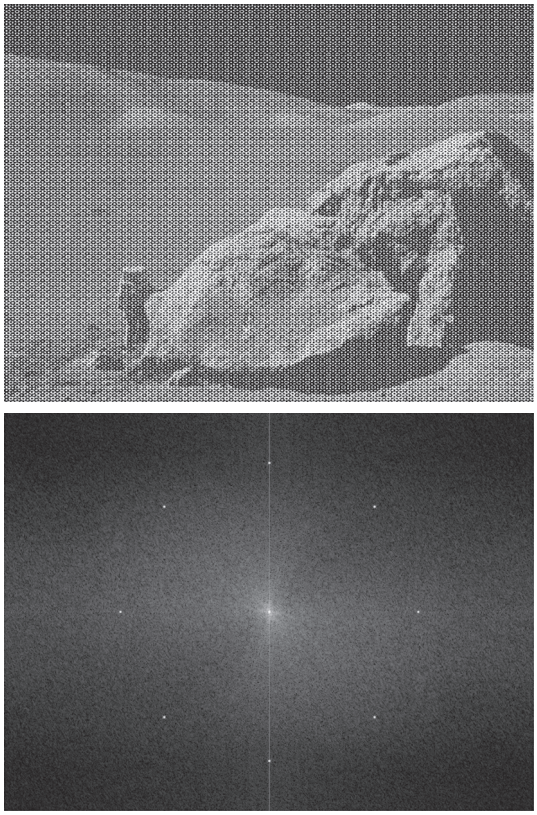


Рис. 5.5. (*а*) Изображение, искаженное синусоидальным шумом. (*б*) Спектр (каждая пара сопряженных импульсов соответствует одной синусоидальной волне). (Изображение предоставлено NASA)

бы попытаться сделать вывод о периодичности шумовых составляющих прямо на основе исходного изображения, однако это приводит к успеху лишь в простейших случаях. Анализ в автоматическом режиме возможен в тех ситуациях, когда шумовые пики ярко выражены или когда имеется информация относительно обычного месторасположения соответствующих частотных составляющих.

Параметры функции плотности распределения вероятностей шума могут быть частично известны исходя из технических характеристик сенсоров, однако часто необходимо оценить эти параметры для конкретной системы, использованной при получении изображения. Если эта система находится в вашем распоряжении, то один из простых способов изучения ее характеристик, связанных с шумом, заключается в том, чтобы получить набор изображений однородных тестовых объектов. В случае оптической системы, например, таким объектом будет являться большая сплошная равномерно освещенная серая область. Полученные таким путем изображения обычно достаточно хорошо характеризуют шум системы.

В тех случаях, когда доступны только изображения, ранее сформированные неизвестной системой, рассмотрение небольших участков изображения примерно постоянной фоновой яркости часто дает возможность оценить параметры

функции плотности распределения вероятностей шума. Например, на рис. 5.6 показаны вертикальные полосы, вырезанные из тех изображений на рис. 5.4, которые отвечают гауссову, рэлеевскому и равномерному шуму. Представленные гистограммы посчитаны на основе данных, отвечающих этим небольшим полосам. Гистограммам на рис. 5.6 соответствуют средние (из трех) части гистограмм на рис. 5.4(*г*), (*д*) и (*л*). Сравнение показывает, что вид гистограмм на рис. 5.6 весьма близок к виду соответствующих частей гистограмм на рис. 5.4. Высота гистограмм различается из-за масштабирования, но их форма, без сомнения, одинакова.

Естественно использовать данные, отвечающие небольшим участкам изображения, для вычисления среднего значения и дисперсии яркости. Рассмотрим некоторую полосу (фрагмент изображения) S, и пусть $p_S(z_i)$, $i = 0, 1, 2, ..., L - 1$, обозначают оценки вероятности (нормированные значения гистограммы) значений яркостей пикселей в S, где L — число возможных уровней яркостей всего изображения (т. е. 256 для 8-битового изображения). Так же, как и в главе 3, среднее и дисперсия значений пикселей в S оцениваются следующим образом:

$$\bar{z} = \sum_{i=0}^{L-1} z_i p(z_i) \tag{5.2-15}$$

и

$$\sigma^2 = \sum_{i=0}^{L-1} (z_i - \bar{z})^2 p_S(z_i). \tag{5.2-16}$$

Вид гистограммы определяет, какая из функций плотности распределения вероятностей является наиболее подходящей. Если форма гистограммы приблизительно гауссова, то все что необходимо — это определить среднее и дисперсию, поскольку гауссова функция плотности распределения полностью определяется этими двумя параметрами. Для распределений других типов, которые обсуждались в разделе 5.2.2, мы рассматриваем выражения для среднего и дисперсии как уравнения для параметров a, b и, разрешая уравнения, находим параметры распределения. Обработка импульсного шума осуществляется по-другому, поскольку в этом случае требуется оценить фактическую вероятность появления черных и белых точек на изображении. Для получения такой оценки необходимо, чтобы были видны как черные, так и белые точки. Таким образом для вычисления гистограммы пригодна только такая область изображения,

а *б* *в*

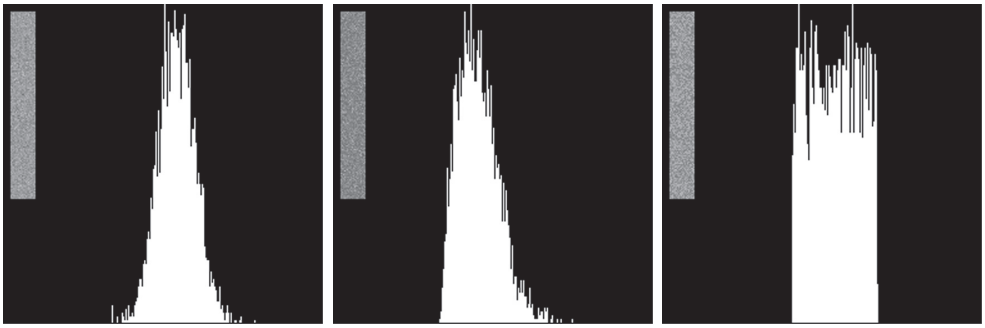


Рис. 5.6. Гистограммы, посчитанные по небольшим полосам (показаны в виде вставок) изображений на рис. 5.4, содержащих (*а*) гауссов, (*б*) рэлеевский и (*в*) равномерный шум

в которой значения яркости лежат в средней части диапазона и относительно постоянны. Высоты пиков, соответствующих черным и белым точкам, дают оценку вероятностей P_a и P_b в (5.2-14)[6].

5.3. Подавление шумов — пространственная фильтрация

Когда искажение изображения обусловлено исключительно наличием шума, равенства (5.1-1) и (5.1-2) приобретают вид

$$g(x,y) = f(x,y) + \eta(x,y) \tag{5.3-1}$$

и

$$G(u,v) = F(u,v) + N(u,v). \tag{5.3-2}$$

Слагаемое, описывающее шум, неизвестно, поэтому просто вычесть его из функции $g(x,y)$ или $G(u,v)$ невозможно. Обычно в случае периодического шума спектр $G(u,v)$ дает возможность оценить величину $N(u,v)$, как было указано в разделе 5.2.3. Тогда в целях построения приближения исходного изображения величина $N(u,v)$ может быть вычтена из функции $G(u,v)$. Однако этот случай является скорее исключением, чем правилом.

В тех ситуациях, когда на изображении присутствует только аддитивный случайный шум, пространственная фильтрация является лучшим из возможных методов восстановления. Пространственная фильтрация детально обсуждается в главе 3. За исключением вычислительной процедуры, характерной для использования какого-то конкретного фильтра, механизм применения всех нижеследующих фильтров в точности такой же, как обсуждался в разделах 3.5—3.6.

5.3.1. Усредняющие фильтры

В этом параграфе мы кратко обсудим возможности подавления шумов пространственными фильтрами, введенными в разделе 3.5, а также построим некоторые другие фильтры, эффективность которых во многих случаях превосходит эффективность фильтров, рассмотренных в том разделе.

Фильтр, основанный на вычислении среднего арифметического

Такой фильтр, называемый *среднеарифметическим*, является простейшим среди усредняющих фильтров. Пусть S_{xy} обозначает прямоугольную окрестность (множество координат точек изображения) размерами $m \times n$ с центром в точке (x,y). Процедура фильтрации предполагает вычисление среднего арифметического значения искаженного изображения $g(x,y)$ по окрестности S_{xy}. Значение восстановленного изображения $\hat{f}$ в произвольной точке (x,y) представляет собой среднее арифметическое значений в точках, принадлежащих окрестности S_{xy}. Другими словами,

Предполагается, что размеры m и n — нечетные числа.

[6] См. также прим. 3. — *Прим. перев.*

$$\hat{f}(x,y)=\frac{1}{mn}\sum_{(s,t)\in S_{xy}} g(s,t). \tag{5.3-3}$$

Эта операция может быть реализована в виде пространственного фильтра размерами $m\times n$, в котором все коэффициенты равны $1/mn$. Усредняющий фильтр сглаживает локальные вариации яркости на изображении, и в результате этого сглаживания происходит уменьшение шума.

Фильтр, основанный на вычислении среднего геометрического

Изображение, восстановленное с использованием *среднегеометрического* фильтра, задается выражением

$$\hat{f}(x,y)=\left[\prod_{(s,t)\in S_{xy}} g(s,t)\right]^{\frac{1}{mn}}. \tag{5.3-4}$$

Здесь значение восстановленного изображения в каждой точке (x,y) является корнем степени mn из произведения значений в точках окрестности S_{xy}. Как показывает пример 5.2, применение среднегеометрического фильтра приводит к сглаживанию, сравнимому с тем, которое достигается при использовании среднеарифметического фильтра, но при этом теряется меньше деталей изображения.

Фильтр, основанный на вычислении среднего гармонического

Результат обработки с применением *среднегармонического* фильтра дается выражением

$$\hat{f}(x,y)=\frac{mn}{\sum\limits_{(s,t)\in S_{xy}} \frac{1}{g(s,t)}}. \tag{5.3-5}$$

Среднегармонический фильтр хорошо работает в случае униполярного «белого» импульсного шума (т.е. когда значение шума соответствует появлению белых точек на изображении), но не работает в случае униполярного «черного» импульсного шума (когда значение шума соответствует появлению черных точек). Этот фильтр также хорошо работает для других типов шума, таких как гауссов шум.

Фильтр, основанный на вычислении среднего контрагармонического

Обработка с применением фильтра среднего *контрагармонического* описывается выражением

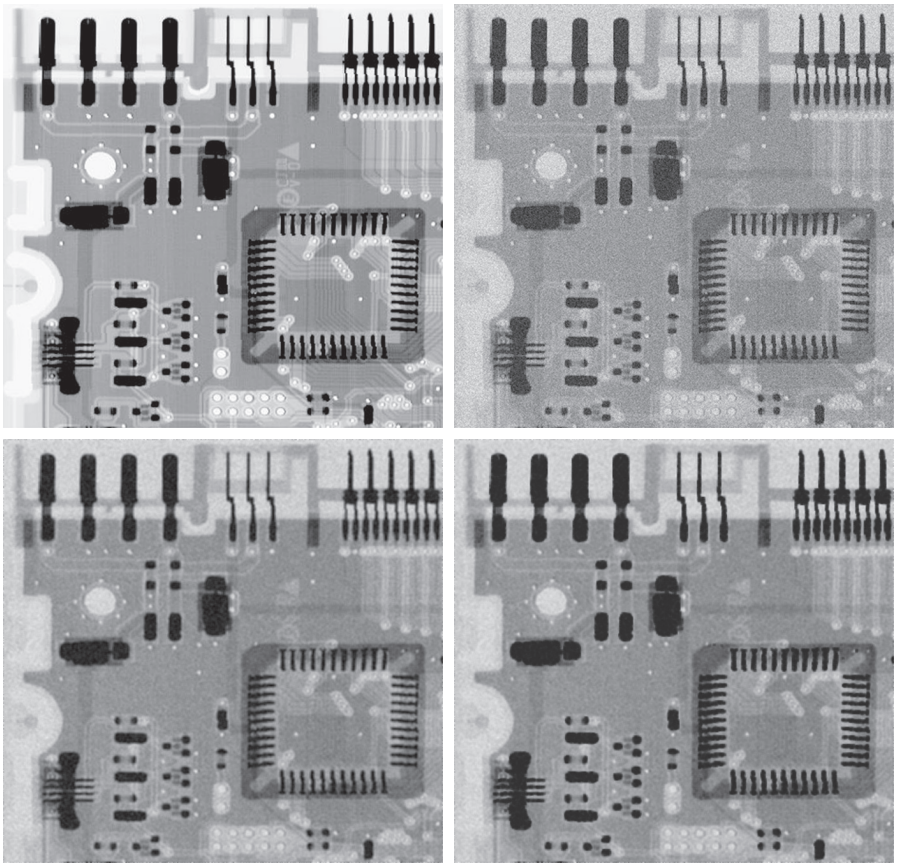
$$\hat{f}(x,y)=\frac{\sum\limits_{(s,t)\in S_{xy}} g(s,t)^{Q+1}}{\sum\limits_{(s,t)\in S_{xy}} g(s,t)^{Q}}, \tag{5.3-6}$$

где Q называется *порядком* фильтра. Этот фильтр хорошо приспособлен для уменьшения или почти полного устранения импульсного шума. При положительных значениях Q фильтр устраняет «черную» часть импульсного шума. При отрицательных значениях Q фильтр устраняет «белую» часть импульсного шума. Обе части шума не могут быть устранены одновременно. Заметим, что

контрагармонический фильтр при $Q = 0$ сводится к среднеарифметическому фильтру, а при $Q = -1$ сводится к среднегармоническому фильтру.

Пример 5.2. Восстановление с помощью усредняющих фильтров.

■ На рис. 5.7(*а*) приведено 8-битовое рентгеновское изображение монтажной платы, а на рис. 5.7(*б*) приведено то же изображение, но искаженное путем добавления гауссова шума с нулевым средним и дисперсией 400. Для изображения такого типа данный уровень шума является значительным. На рис. 5.7(*в*) и (*г*) представлены результаты фильтрации изображения с шумом при использовании среднеарифметического и среднегеометрического фильтров размерами 3×3. Хотя оба фильтра дали приемлемый результат в плане уменьшения шума на изображении, применение среднегеометрического фильтра привело к меньшему размыванию изображения, чем применение среднеарифметического фильтра. Например, штырьки разъема в верхней части изображения вы-



а б
в г

Рис. 5.7. (*а*) Рентгеновский снимок. (*б*) Изображение, искаженное аддитивным гауссовым шумом. (*в*) Результат фильтрации с использованием среднеарифметического фильтра размерами 3×3. (*г*) Результат фильтрации с использованием среднегеометрического фильтра тех же размеров. (Исходное изображение предоставил Джозеф Е. Пассенте, Lixi Inc.)

а б
в г

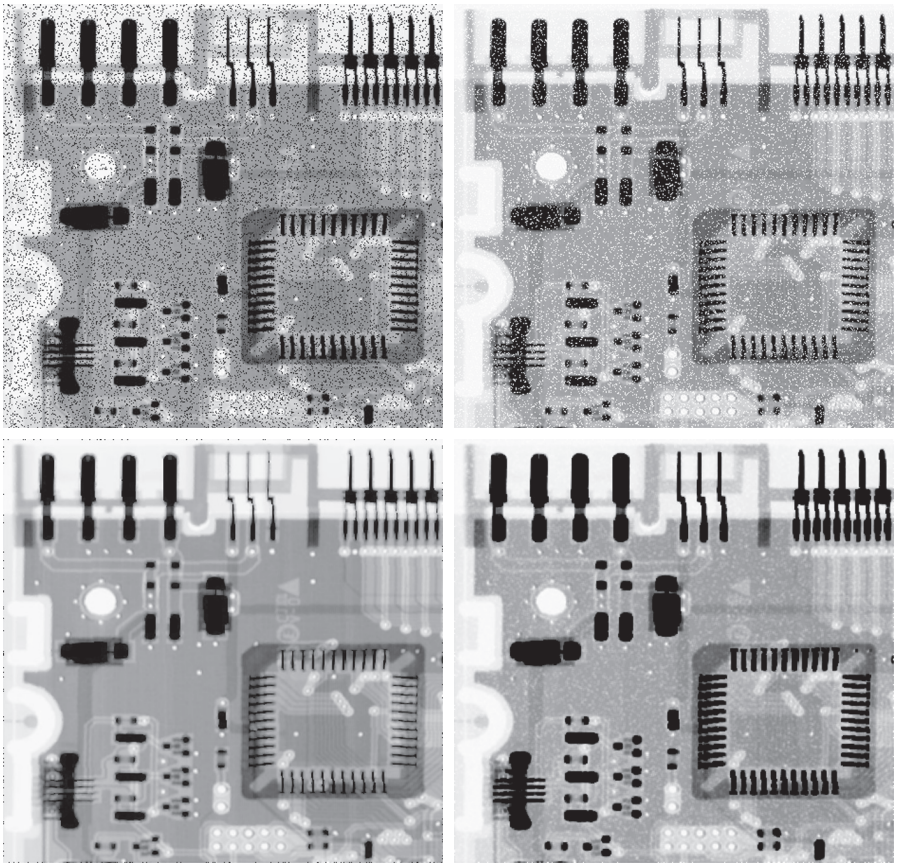
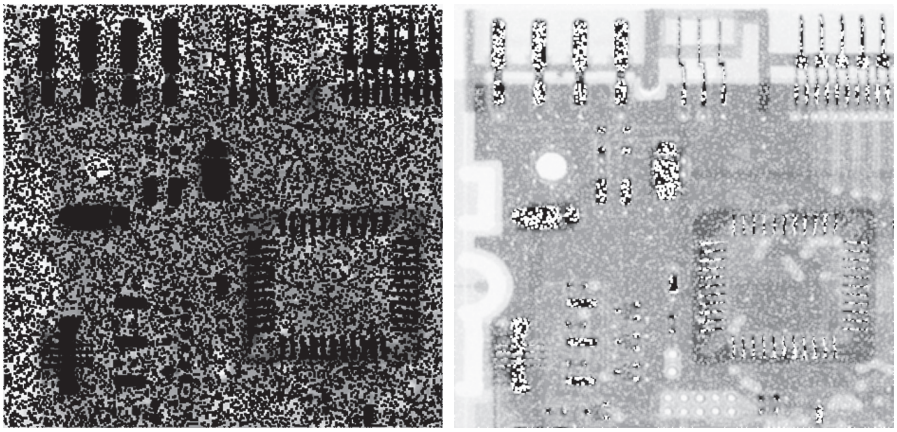


Рис. 5.8. (*а*) Изображение, искаженное униполярным «черным» импульсным шумом с вероятностью 0,1. (*б*) Изображение, искаженное униполярным «белым» импульсным шумом с той же вероятностью. (*в*) Результат фильтрации изображения (*а*) с использованием 3×3 контрагармонического фильтра порядка $Q = 1{,}5$. (*г*) Результат фильтрации изображения (*б*) с использованием аналогичного фильтра порядка $Q = -1{,}5$

глядят на рис. 5.7(*г*) намного более резкими, чем на рис. 5.7(*в*). Это справедливо и для других частей изображения.

На рис. 5.8(*а*) представлено изображение той же платы, но искаженное униполярным «черным» импульсным шумом с вероятностью 0,1. Аналогично на рис. 5.8(*б*) представлено изображение, искаженное униполярным «белым» импульсным шумом с той же вероятностью. На рис. 5.8(*в*) представлен результат фильтрации изображения на рис. 5.8(*а*) с использованием контрагармонического фильтра порядка $Q = 1{,}5$, а на рис. 5.8(*г*) представлен результат фильтрации изображения на рис. 5.8(*б*) с использованием аналогичного фильтра порядка $Q = -1{,}5$. В обоих случаях применение фильтров дало хороший результат в плане уменьшения шума. Результаты работы фильтра положительного порядка лучше в фоновой области, издержки выражаются в небольшом утончении и размывании темных областей. Обратное имеет место для фильтра отрицательного порядка.



а б

Рис. 5.9. Результаты неправильного выбора знака порядков для контрагармонических фильтров. (*а*) Результат фильтрации изображения рис. 5.8(*а*) с использованием контрагармонического фильтра размерами 3×3 порядка $Q = -1{,}5$. (*б*) Результат фильтрации изображения рис. 5.8(*б*) с $Q = 1{,}5$

Вообще среднеарифметические и среднегеометрические фильтры (особенно последние) подходят для фильтрации случайных шумов типа гауссова или равномерного. Контрагармонические фильтры подходят для фильтрации импульсного шума, но их применение затруднено тем, что необходимо заранее знать, является ли шум «черным» или «белым», поскольку необходимо выбрать правильный знак порядка фильтра Q. Как показывает рис. 5.9, неправильный выбора знака может привести к катастрофическим результатам. Некоторые из фильтров, рассматриваемых в следующем параграфе, свободны от этого недостатка. ■

5.3.2. Фильтры, основанные на порядковых статистиках

Фильтры, основанные на порядковых статистиках, были введены в разделе 3.5.2. В этом параграфе мы расширим рамки обсуждения и введем дополнительно еще некоторые фильтры подобного рода. Как отмечалось в разделе 3.5.2, фильтры, основанные на порядковых статистиках, представляют собой пространственные фильтры, вычисление отклика которых требует предварительного упорядочивания (ранжирования) значений пикселей, заключенных внутри обрабатываемой фильтром области изображения. Результат упорядочивания определяет отклик фильтра.

Медианные фильтры

Наиболее известным из фильтров, основанных на порядковых статистиках, является *медианный* фильтр. Действие этого фильтра, как следует из его названия, состоит в замене значения в точке изображения на медиану значений яркости в окрестности этой точки:

$$\hat{f}(x,y) = \underset{(s,t)\in S_{xy}}{\text{med}} \{g(s,t)\}. \qquad (5.3\text{-}7)$$

При вычислении медианы значение в самой точке (т. е. в центре окрестности) также учитывается. Широкая популярность медианных фильтров обусловлена тем, что они прекрасно приспособлены для подавления некоторых видов случайных шумов и при этом приводят к меньшему размыванию по сравнению с линейными сглаживающими фильтрами того же размера. Медианные фильтры особенно эффективны при наличии как биполярного, так и униполярного импульсного шума. На самом деле, как показывает пример 5.3, применение медианных фильтров дает отличные результаты для изображений, которые искажены шумом этого типа. Процедура вычисления медианы и реализация медианной фильтрации детально обсуждались в разделе 3.6.2.

Фильтры, основанные на выборе максимального и минимального значения

Хотя медианные фильтры, безусловно, принадлежат к числу наиболее часто используемых в обработке изображений фильтров, основанных на порядковых статистиках, это отнюдь не единственный пример таких фильтров. Медиана представляет собой квантиль уровня 0,5 упорядоченного набора чисел, однако использование иных статистических характеристик предоставляет много других возможностей. Например, использование квантиля уровня 1,0 приводит к фильтру, основанному на выборе максимального значения (или *фильтру максимума*), который задается выражением

$$\hat{f}(x,y) = \max_{(s,t)\in S_{xy}} \{g(s,t)\}. \tag{5.3-8}$$

Такой фильтр полезен при обнаружении наиболее ярких точек на изображении. Кроме того, поскольку униполярный «черный» импульсный шум принимает минимальные значения, применение этого фильтра приводит к уменьшению такого шума, так как в процессе фильтрации из окрестности S_{xy} выбирается максимальное значение.

Касательно квантилей см. вторую заметку раздела 10.3.5 на стр. 868

Использование квантиля уровня 0 приводит к фильтру, основанному на выборе минимального значения (или *фильтру минимума*):

$$\hat{f}(x,y) = \min_{(s,t)\in S_{xy}} \{g(s,t)\}. \tag{5.3-9}$$

Такой фильтр полезен при обнаружении наиболее темных точек на изображении. Кроме того, применение этого фильтра приводит к уменьшению униполярного «белого» импульсного шума вследствие операции выбора минимума.

Фильтр срединной точки

Применение фильтра *срединной точки* заключается просто в вычислении среднего между максимальным и минимальным значениями в соответствующей окрестности:

$$\hat{f}(x,y) = \frac{1}{2}\left[\max_{(s,t)\in S_{xy}} \{g(s,t)\} + \min_{(s,t)\in S_{xy}} \{g(s,t)\}\right]. \tag{5.3-10}$$

Отметим, что этот фильтр объединяет в себе методы порядковых статистик и усреднения. Он лучше всего работает при наличии таких случайно распределенных шумов, как гауссов или равномерный.

Фильтр, основанный на вычислении усеченного среднего

Предположим, что мы удалили $d/2$ наименьших и $d/2$ наибольших значений яркости из множества всех значений функции $g(s,t)$ в окрестности S_{xy}. Пусть $g_r(s,t)$ представляет собой оставшиеся элементы изображения, количество которых равно $(mn - d)$. Фильтр, действие которого заключается в усреднении оставшихся значений, называется фильтром *усеченного среднего*

$$\hat{f}(x,y) = \frac{1}{mn-d} \sum_{(s,t)\in Sxy} g_r(s,t), \tag{5.3-11}$$

причем значение d может изменяться в диапазоне от 0 до $mn - 1$. В случае $d = 0$ фильтр усеченного среднего сводится к среднеарифметическому фильтру, который рассматривался в предыдущем параграфе. В случае $d = mn - 1$ фильтр превращается в медианный фильтр. Использование фильтра усеченного среднего с другими значениями d полезно в тех случаях, когда мы имеем дело с несколькими видами шума одновременно, например с комбинацией импульсного и гауссова шума.

Пример 5.3. Восстановление с помощью фильтров, основанных на порядковых статистиках.

■ На рис. 5.10(*а*) представлено изображение платы, искаженное биполярным импульсным шумом с вероятностями $P_a = P_b = 0{,}1$. На рис. 5.10(*б*) представлен результат обработки с использованием медианного фильтра размерами 3×3. Улучшение изображения по сравнению с изображением на рис. 5.10(*а*) весьма значительное, однако по-прежнему можно видеть остатки шума в некотором количестве точек. Второй проход (по изображению на рис. 5.10(*б*)) медианным фильтром удаляет большинство этих точек, оставляя лишь едва заметную их часть. Последние удаляются при третьем проходе. Эти результаты наглядно демонстрируют силу медианной фильтрации при обработке изображений, содержащих шум типа импульсного. Следует иметь в виду, что повторные применения медианной фильтрации приводят к размыванию изображения, поэтому желательно, чтобы число проходов было как можно меньше.

На рис. 5.11(*а*) представлен результат применения фильтра максимума к изображению на рис. 5.8(*а*), искаженному униполярным «черным» импульсным шумом. Фильтрация дала разумный в смысле устранения шума результат, однако необходимо отметить, что некоторые темные точки на границах темных объектов также оказались удалены (эти точки приобрели значения яркости из светлой части диапазона). На рис. 5.11(*б*) представлен результат применения фильтра минимума к изображению на рис. 5.8(*б*). В рассматриваемом случае результат работы фильтра минимума по устранению шума лучше, чем аналогичный результат для фильтра максимума, но фильтр минимума удалил ряд белых точек около границ светлых объектов. Вследствие этого светлые объекты стали меньше по размерам, а некоторые темные объекты (как, например, штырьки разъема в верхней части изображения) стали больше, поскольку белые точки вокруг этих объектов заменились на темные.

Продемонстрируем теперь работу фильтра усеченного среднего. На рис. 5.12(*а*) представлено изображение платы, искаженное в данном случае добавлением равномерного шума с нулевым средним и дисперсией 800. На это

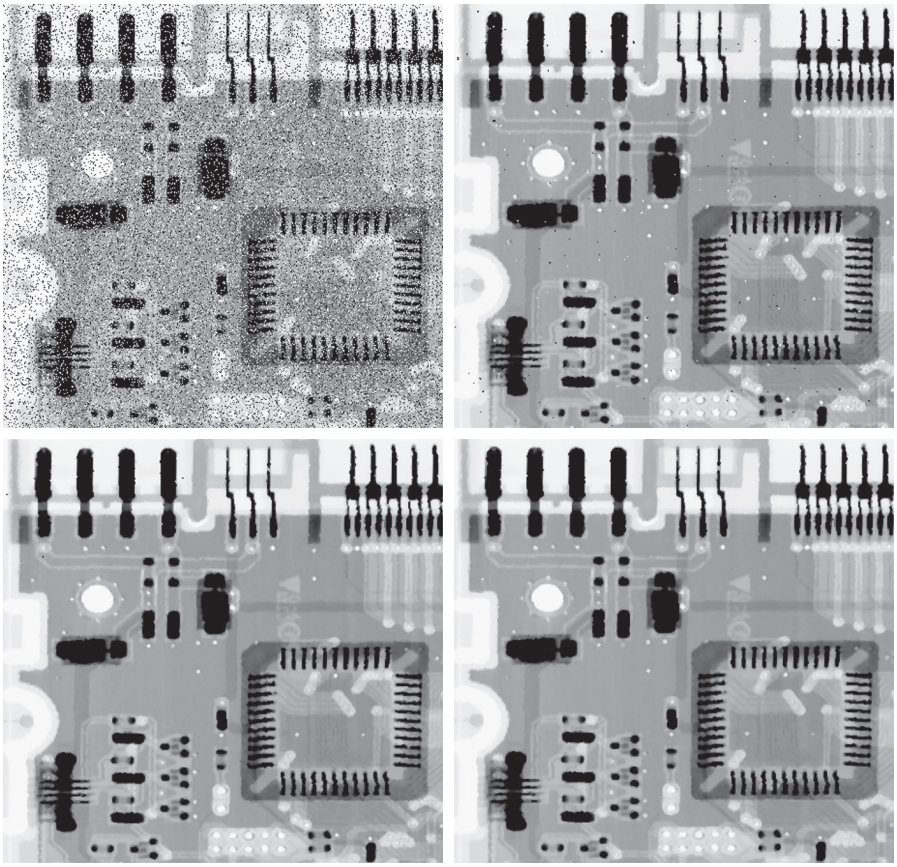
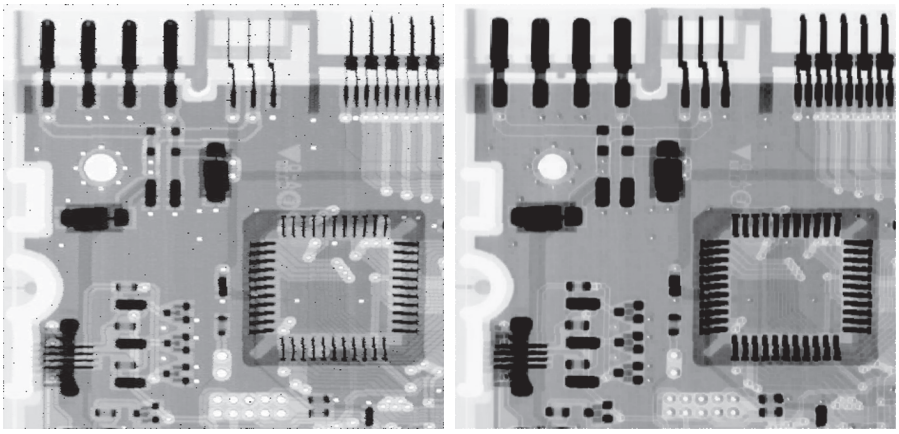


Рис. 5.10. (*а*) Изображение, искаженное биполярным импульсным шумом с вероятностями $P_a = P_b = 0{,}1$. (*б*) Результат обработки с использованием медианного фильтра размерами 3×3. (*в*) Результат обработки (*б*) с использованием того же фильтра. (*г*) Результат обработки (*в*) с использованием того же фильтра

изображение с высоким уровнем шумовых искажений далее накладывается биполярный импульсный шум с вероятностями $P_a = P_b = 0{,}1$, что, как видно на рис. 5.12(*б*), еще сильнее ухудшает его. При таком высоком уровне шума нужно использовать фильтры большого размера. На рис. 5.12(*в*)—(*е*) представлены результаты фильтрации, полученные с помощью среднеарифметического, среднегеометрического, медианного и усеченного среднего (с $d = 6$) фильтров размерами 5×5. Как и следовало ожидать, применение среднеарифметического и среднегеометрического фильтров (особенно последнего) не дало хороших результатов, что объясняется присутствием импульсного шума. Результаты, полученные с применением медианного фильтра и фильтра усеченного среднего, намного лучше, причем фильтр усеченного среднего привел к несколько большему уменьшению шума. Обратите внимание, например, что четыре штырька разъема в верхней левой части изображения, полученного в результате применения фильтра усеченного среднего, выглядят несколько более гладко. Это не удивительно, поскольку при больших значениях d фильтр усеченного среднего



а б

Рис. 5.11. (*а*) Результат фильтрации изображения на рис. 5.8(*а*) с использованием фильтра максимума размерами 3×3. (*б*) Результат фильтрации изображения на рис. 5.8(*б*) с использованием фильтра минимума тех же размеров

по своему действию приближается к медианному фильтру, но все еще сохраняет свои сглаживающие способности. ■

5.3.3. Адаптивные фильтры

Рассмотренные до сих пор фильтры обладали одной общей особенностью. Зафиксировав однажды какой-нибудь из этих фильтров, мы в дальнейшем применяли его к изображению без учета того, как свойства изображения меняются от точки к точке. В этом параграфе мы познакомимся с двумя простыми *адаптивными* фильтрами, поведение которых изменяется в зависимости от статистических свойств изображения внутри области действия фильтра, которая определяется прямоугольной $m \times n$ окрестностью S_{xy}. Как показывает дальнейшее обсуждение, возможности адаптивных фильтров превосходят возможности фильтров, которые были рассмотрены до сих пор. Платой за усовершенствование методов фильтрации является увеличение сложности фильтров. Будем иметь в виду, что мы по-прежнему находимся в ситуации, когда искаженное изображение представляет собой сумму исходного изображения и шума. Никакие другие виды искажений до сих пор не рассматривались.

Адаптивные локальные фильтры подавления шума

Простейшими характеристиками случайной величины являются ее среднее значение и дисперсия. Эти параметры естественно взять за основу при создании адаптивного фильтра, поскольку их величины тесно связаны с внешним видом изображения. Среднее значение дает меру средней яркости той области, по которой оно вычисляется, а дисперсия дает меру среднего отклонения в этой области.

Наш фильтр должен действовать в окрестности S_{xy}. Отклик фильтра в некоторой точке (x,y), которая является центром этой окрестности, должен определятся четырьмя величинами: (*а*) значением $g(x,y)$ изображения с шумом

а *б*
в *г*
д *е*

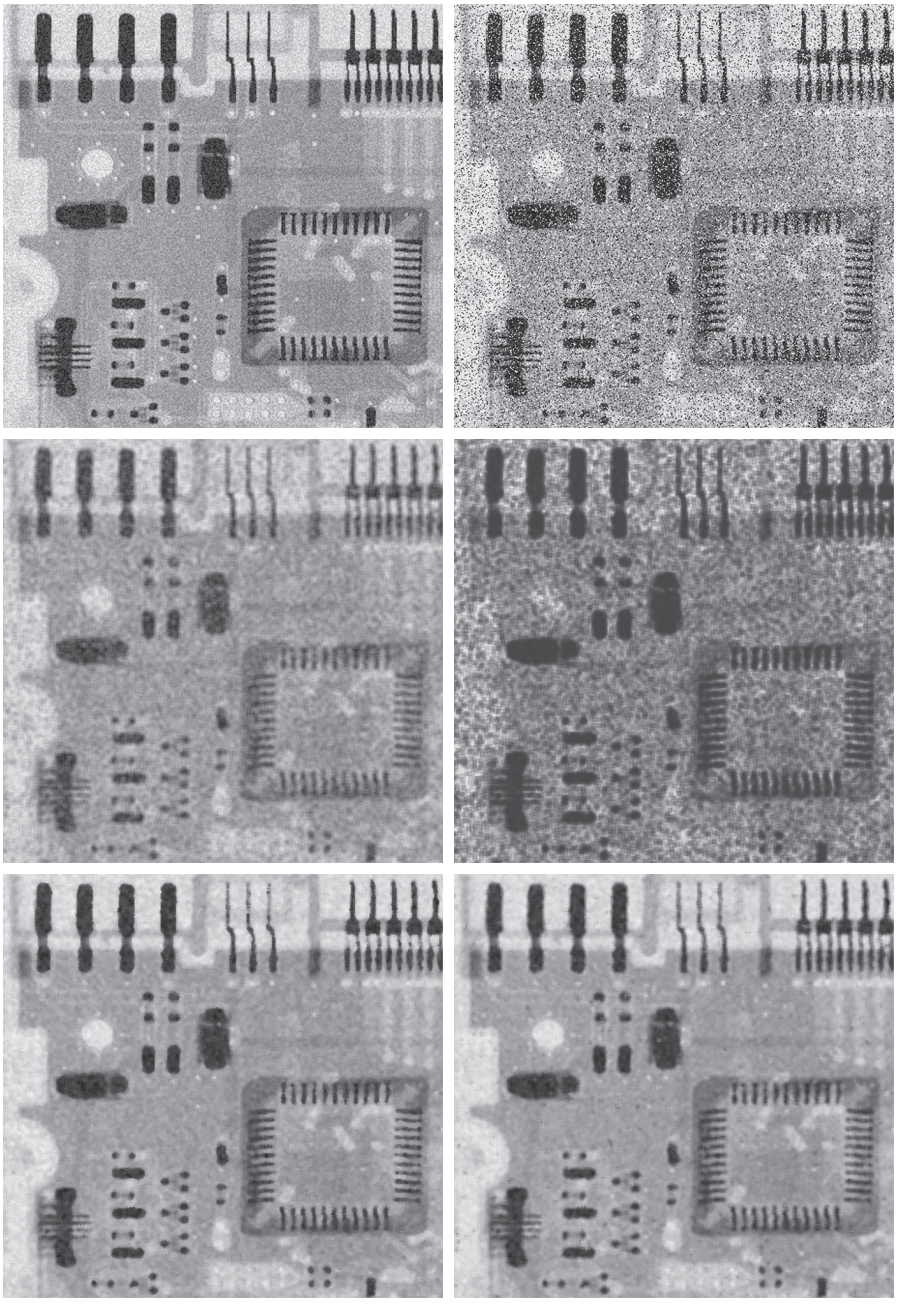


Рис. 5.12. (*а*) Изображение, искаженное аддитивным равномерным шумом. (*б*) Изображение, дополнительно искаженное биполярным импульсным шумом. Результат фильтрации изображения (*б*) применением фильтров размерами 5×5; (*в*) среднеарифметического фильтра; (*г*) среднегеометрического фильтра; (*д*) медианного фильтра, и (*е*) фильтра усеченного среднего с $d = 6$

в точке (x,y); (б) дисперсией σ_η^2 шума, превращающего исходное изображение $f(x,y)$ в искаженное изображение $g(x,y)$; (в) локальным средним m_L по значениям в окрестности S_{xy} и (г) локальной дисперсией σ_L^2 по значениям в окрестности S_{xy}. Мы хотим, чтобы поведение фильтра определялось следующими условиями.

1. Если дисперсия шума σ_η^2 равна нулю, то отклик фильтра должен быть равен значению $g(x,y)$. Это отвечает тривиальному случаю нулевого шума, когда $g(x,y)$ равно $f(x,y)$.
2. Если локальная дисперсия σ_L^2 много больше σ_η^2, то значение отклика фильтра должно быть близко $g(x,y)$. Большое значение локальной дисперсии обычно связано с наличием контуров, которые должны быть сохранены.
3. Если значения обеих дисперсий равны, то отклик фильтра должен быть равен среднему арифметическому значений в окрестности S_{xy}. Условие выполнено в том случае, когда статистические характеристики данной локальной области и изображения в целом совпадают и локальный шум должен быть уменьшен, для чего используется простое усреднение.

Адаптивный фильтр, удовлетворяющий перечисленным условиям, может быть задан следующим выражением:

$$\hat{f}(x,y) = g(x,y) - \frac{\sigma_\eta^2}{\sigma_L^2}\left[g(x,y) - m_L\right]. \qquad (5.3\text{-}12)$$

Единственной величиной, которая должна быть заранее известна или оценена, является полная дисперсия шума σ_η^2. Остальные входящие в формулу величины вычисляются для каждой точки (x,y) по значениям элементов изображения в окрестности S_{xy} с центром в этой точке. В формуле (5.3-12) неявно предполагается, что $\sigma_\eta^2 \le \sigma_L^2$. Поскольку мы рассматриваем модели аддитивного и трансляционно-инвариантного шума, такое предположение является оправданным, так как множество элементов в окрестности S_{xy} является подмножеством элементов изображения. Однако нам редко известна точная информация о значении σ_η^2. Поэтому при реализации формулы (5.3-12) нужно предусмотреть дополнительную проверку условия $\sigma_\eta^2 \le \sigma_L^2$ с тем, чтобы при нарушении этого условия использовать в формуле (5.3-12) значение 1 вместо величины соответствующего отношения. В результате метод фильтрации становится нелинейным. Однако такой подход предотвращает появление бессмысленных результатов (т. е. отрицательных значений яркости при некоторых значениях m_L), обусловленных недостатком информации о дисперсии шума на изображении. Другой подход может состоять в том, чтобы допустить появление отрицательных значений, но затем изменить шкалу яркости. Такой подход приведет к уменьшению динамического диапазона изображения.

Пример 5.4. Восстановление с помощью адаптивных фильтров локального уменьшения шума.

■ На рис. 5.13(*а*) представлено изображение платы, на этот раз искаженное добавлением гауссова шума с нулевым средним и дисперсией 1000. Такой уровень шума приводит к значительным искажениям, что создает, однако, идеальное тестовое изображение для сравнения рабочих характеристик фильтров. На рис. 5.13(*б*) представлен результат обработки изображения с шумом при по-

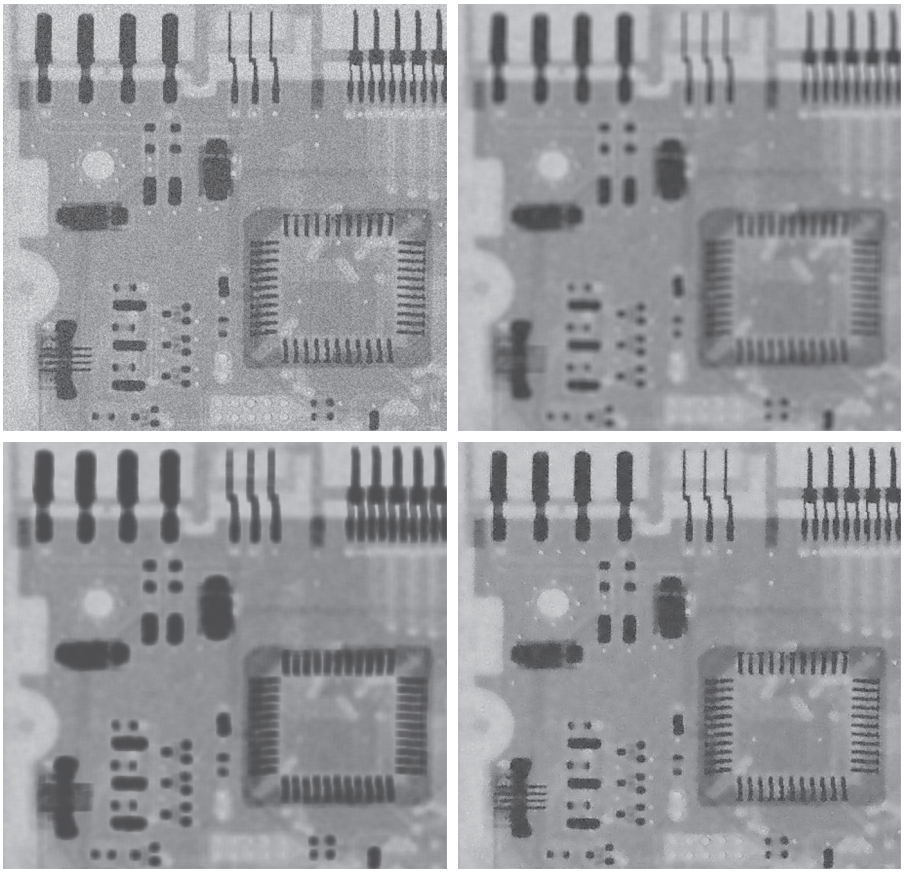


Рис. 5.13. (*а*) Изображение, искаженное аддитивным гауссовым шумом с нулевым средним и дисперсией 1000. (*б*) Результат обработки с использованием среднеарифметического фильтра. (*в*) Результат обработки с использованием среднегеометрического фильтра. (*г*) Результат обработки с использованием адаптивного фильтра уменьшения шума. Размеры всех фильтров равны 7×7

мощи среднеарифметического фильтра размерами 7×7. Шум на изображении удалось сгладить ценой значительного размывания. Это же имеет место и для изображения на рис. 5.13(*в*), где представлен результат обработки при помощи среднегеометрического фильтра тех же размеров 7×7. Различия между последними двумя изображениями аналогичны тем, которые обсуждались при рассмотрении примера 5.2, увеличилась лишь степень размывания.

На рис. 5.13(*г*) представлен результат применения адаптивного фильтра (5.3-12) размерами 7×7 при $\sigma_\eta^2 = 1000$. Этот результат значительно лучше двух предыдущих. Общее уменьшение шума, которое достигается при использовании адаптивного фильтра, сравнимо с уменьшением шума при использовании среднеарифметического и среднегеометрического фильтров. Однако изображение, получаемое после обработки с помощью адаптивного фильтра, является намного более резким. Например, штырьки разъема в верхней части изображе-

ния платы выглядят на рис. 5.13(*г*) намного более резкими. Другие характерные детали, такие как отверстия или восемь ножек элемента схемы в левом нижнем углу изображения, видны на рис. 5.13(*г*) значительно более отчетливо. Эти результаты являются типичным примером того, чего можно достичь, используя адаптивные фильтры. Как было отмечено ранее, платой за улучшение рабочих характеристик фильтров является увеличение их сложности.

При получении представленных выше результатов мы использовали значение σ_η^2, которое в точности соответствует значению дисперсии шума. Если эта величина заранее неизвестна и используемая для нее оценка слишком занижена, то в результате обработки мы получим изображение, которое очень похоже на исходное (до обработки) изображение, поскольку коррекция будет меньше, чем ей следовало бы быть. Использование завышенной оценки приведет к тому, что отношение дисперсий в (5.3-12) будет обрезаться на уровне 1 для большего, чем следовало бы, числа точек. Для таких точек отклик фильтра будет равен значению локального среднего[7]. При другом подходе к реализации алгоритма — когда допускаются отрицательные значения (т. е. обрезание отсутствует), а в конце производится изменение шкалы яркости — результатом использования слишком завышенной оценки станет, как указывалось выше, сокращение динамического диапазона. ■

Адаптивные медианные фильтры

Рассмотренные в разделе 5.3.2 медианные фильтры работают хорошо, если пространственная плотность импульсного шума невелика (эмпирическое правило — P_a и P_b не превышают 0,2). В этом параграфе показано, что адаптивная медианная фильтрация помогает справиться с импульсным шумом, вероятности которого превышают указанные значения. Дополнительное преимущество адаптивного медианного фильтра состоит в том, что такой фильтр «старается сохранить детали» в областях, искаженных не импульсным шумом. Обычный медианный фильтр таким свойством не обладает. Подобно всем рассмотренным до сих пор фильтрам, адаптивный медианный фильтр осуществляет обработку в прямоугольной окрестности S_{xy}. Однако, в отличие от этих фильтров, адаптивный медианный фильтр изменяет (увеличивает) размеры окрестности S_{xy} во время работы в соответствии с приведенными ниже условиями. Будем помнить о том, что отклик фильтра представляет собой единственное число, замещающее значение элемента изображения в той точке (x,y), которая является центром окрестности S_{xy} в текущий момент.

Введем следующие обозначения:

$z_{\min}$ — минимальное значение яркости в S_{xy};
$z_{\max}$ — максимальное значение яркости в S_{xy};
z_{med} — медиана значений яркости в S_{xy};
z_{xy} — значение яркости в точке (x, y);
$S_{\max}$ — максимальный допустимый размер S_{xy}.

---

[7] Т. е. будет совпадать с откликом среднеарифметического фильтра. В других точках отклик фильтра также будет ближе, чем следовало бы, к среднеарифметическому значению, что в конечном счете приведет к излишнему размыванию изображения. — *Прим. перев.*

Алгоритм адаптивной медианной фильтрации состоит из двух ветвей, обозначенных ниже как ветвь А и ветвь Б, и его действие заключается в следующем.

Ветвь А: $A1 = z_{\text{med}} - z_{\text{min}}$;
$A2 = z_{\text{med}} - z_{\text{max}}$;
Если $A1 > 0$ И $A2 < 0$, перейти к ветви Б;
Иначе увеличить размер окрестности;
Если размер окрестности $\le S_{\text{max}}$, повторить ветвь А;
Иначе результат равен z_{med}.

Ветвь Б: $B1 = z_{xy} - z_{\text{min}}$;
$B2 = z_{xy} - z_{\text{max}}$;
Если $B1 > 0$ И $B2 < 0$, результат равен z_{xy};
Иначе результат равен z_{med}.

Для понимания того, как работает этот алгоритм, необходимо помнить, что его применение преследует три основные цели: удалить биполярный импульсный шум, обеспечить сглаживание шумов других типов, а также свести к минимуму такие искажения, как чрезмерное утончение или утолщение границ объектов. Значения z_{min} и z_{max} воспринимаются алгоритмом статистически как значения «импульсных» составляющих шума, даже если они не равны наименьшему и наибольшему возможным значениям яркости на изображении.

С учетом последнего замечания мы видим, что ветвь А алгоритма преследует цель определить, является ли медиана z_{med} импульсом («черным» или «белым») или нет. Если условие $z_{\text{min}} < z_{\text{med}} < z_{\text{max}}$ выполнено, то в силу указанных в предыдущем абзаце причин z_{med} не может быть импульсом. В этом случае мы переходим к ветви Б и проверяем, является ли импульсом значение z_{xy} в той точке, которая отвечает центру окрестности (напомним, что мы строим отклик фильтра в этой точке). Если условия $B1 > 0$ И $B2 < 0$ выполнены, то $z_{\text{min}} < z_{xy} < z_{\text{max}}$ и значение z_{xy} не является импульсным по тем же причинам, что и выше. В этом случае алгоритм дает на выходе неизмененное значение z_{xy}. Сохранение значений в таких точках «промежуточного уровня» яркости минимизирует искажения, вносимые обработкой изображения. Если одно из условий $B1 > 0$ И $B2 < 0$ нарушено, то либо $z_{xy} = z_{\text{min}}$, либо $z_{xy} = z_{\text{max}}$. В обоих случаях значение является экстремальным, и алгоритм дает на выходе значение медианы z_{med}, которое, как следует из результатов работы ветви А, не является значением импульсного шума. Последняя операция соответствует действию обычного медианного фильтра. Отличие заключается в том, что обычный медианный фильтр заменяет значение в каждой точке на значение медианы по соответствующей окрестности. Это приводит к излишним искажениям деталей на изображении.

Продолжим далее объяснение работы алгоритма и предположим, что вычисленная при работе ветви А медиана *является* импульсом (т.е. нарушено условие перехода к ветви Б). В таком случае алгоритм предполагает увеличение размеров окрестности и повторение вычислений по ветви А. Процесс повторяется до тех пор, пока либо не будет найдена медиана, отличная от импульса, либо размеры окрестности не достигнут максимального допустимого размера. В последнем случае алгоритм дает на выходе значение z_{med}. При этом нет гарантий того, что это значение не является импульсным. Чем меньше вероятности шума P_a и/или P_b или чем больше максимальный допустимый размер окрестности S_{max}, тем меньше вероятность такого преждевременного (без пере-

хода к ветви Б) выхода из алгоритма. Это утверждение выглядит очень правдоподобно. Естественно ожидать, что при увеличении плотности импульсов будет необходимо использовать окрестность большего размера для устранения шумовых пиков.

После получения значения обрабатываемого элемента изображения центр окрестности смещается в позицию следующего элемента. Алгоритм инициализируется вновь и применяется к пикселям внутри окрестности S_{xy}, находящейся в новом положении. В задаче 3.18 требовалось разработать алгоритм обновления значения медианы при передвижении центра окрестности от точки к точке и тем самым уменьшить объем необходимых вычислений.

Пример 5.5. Восстановление с помощью адаптивного медианного фильтра.
■ На рис. 5.14(*а*) представлено изображение монтажной платы, искаженное биполярным импульсным шумом с вероятностями $P_a = P_b = 0{,}25$; уровень шума здесь в 2,5 раза выше, чем на рис. 5.10(*а*). Шум настолько велик, что мешает разглядеть бóльшую часть деталей на изображении. Чтобы получить образец для сравнения, мы сначала отфильтровали изображение, используя наименьший медианный фильтр, необходимый для удаления наиболее явных следов импульсного шума. Для этого потребовался медианный фильтр размерами 7×7. Результат фильтрации представлен на рис. 5.14(*б*). Хотя шум удалось эффективно устранить, однако фильтрация привела к значительным искажениям деталей на изображении. Например, некоторые штырьки разъема в верхней части изображения выглядят деформированными или разорванными. Другие детали изображения искажены аналогичным образом.

На рис. 5.14(*в*) представлен результат применения адаптивного медианного фильтра с $S_{\max} = 7$. Степень устранения шума такая же, как в случае медианного фильтра. Однако что касается сохранения резкости и правильного воспроизведения деталей, улучшения весьма значительны. Штырьки разъема менее деформированы, некоторые другие характерные детали, которые после восстановления с помощью медианного фильтра были плохо различимы или искажены, выглядят на рис. 5.14(*в*) более резко и лучше опознаются. В качестве двух примечательных примеров можно указать небольшие белые отверстия, раз-

а б в

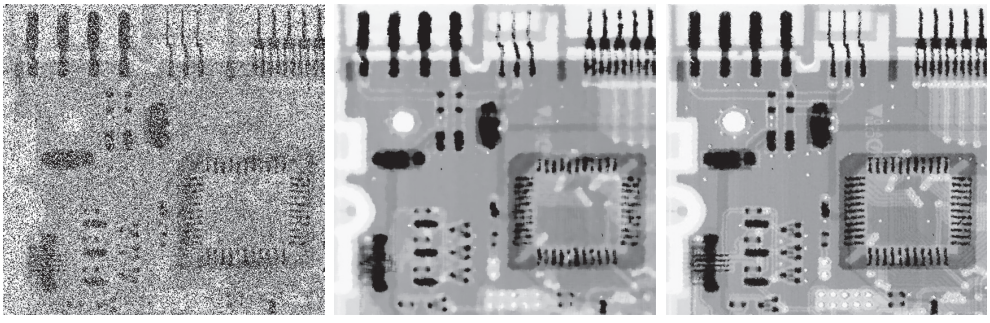


Рис. 5.14. (*а*) Изображение, искаженное биполярным импульсным шумом с вероятностями $P_a = P_b = 0{,}25$. (*б*) Результат обработки с использованием медианного фильтра размерами 7×7. (*в*) Результат адаптивной медианной фильтрации с $S_{\max} = 7$

бросанные по поверхности платы, и элемент схемы с восемью ножками в левом нижнем углу изображения.

Принимая во внимание высокий уровень шума на рис. 5.14(*а*), результат работы адаптивного алгоритма может быть признан вполне хорошим. Выбор максимального допустимого размера окрестности в алгоритме зависит от конкретного приложения, однако разумную оценку этого параметра можно получить, экспериментируя предварительно с обычными медианными фильтрами различных размеров. Эти эксперименты позволят также оценить, каких результатов можно ожидать от применения адаптивного алгоритма. ■

5.4. Подавление периодического шума — частотная фильтрация

Периодический шум можно анализировать и достаточно эффективно фильтровать при помощи частотных методов. Основная идея заключается в том, что в спектре Фурье периодический шум выглядит как концентрированный выброс энергии в позиции, соответствующей частотам периодической помехи. Подход сводится к использованию избирательного фильтра (см. раздел 4.10), способного изолировать шум. В качестве основного способа снижения периодического шума используются три типа избирательных фильтров (режекторный, полосовой и узкополосный, введенные в разделе 4.10), которые рассмотрены в разделах 5.4.1—5.4.3. В разделе 5.4.4 также рассматривается подход к выбору оптимального узкополосного фильтра.

5.4.1. Режекторные фильтры

Передаточные функции трех режекторных фильтров: идеального, Баттерворта и гауссова, введенных в разделе 4.10.1, приведены в табл. 4.6. На рис. 5.15 представлены трехмерные изображения этих фильтров, а следующий пример демонстрирует способ использования режекторного фильтра для устранения эффекта периодического шума.

Пример 5.6. Использование режекторных фильтров для устранения периодического шума.

■ Одним из главных применений режекторных фильтров является удаление периодического шума в тех случаях, когда достаточно точно известна лока-

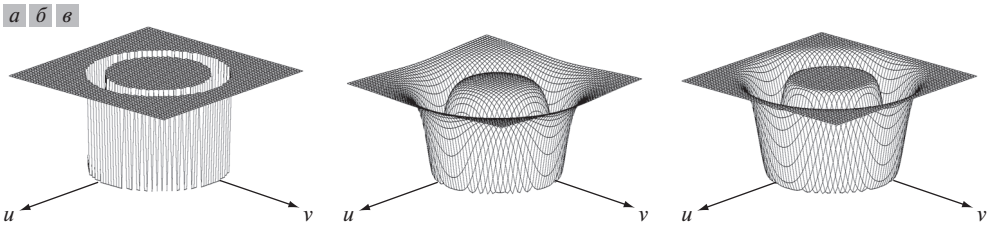
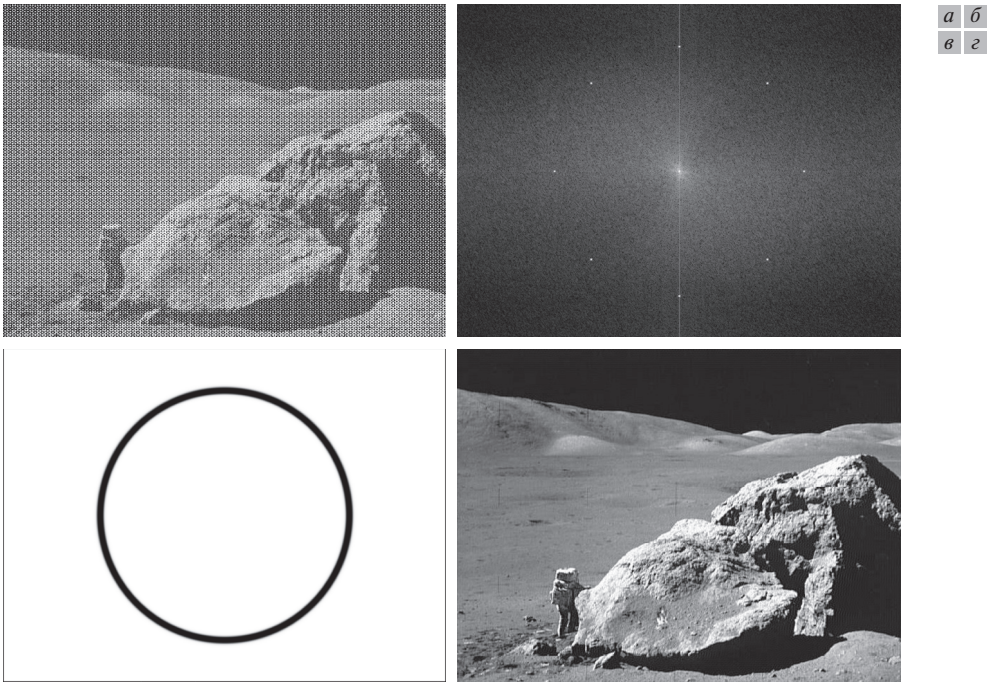


Рис. 5.15. Трехмерные изображения режекторных фильтров: (*а*) идеальный фильтр; (*б*) фильтр Баттерворта порядка 1; (*в*) гауссов фильтр

лизация шумовых составляющих в частотном пространстве. Хорошим примером здесь является изображение, искаженное аддитивным периодическим шумом, который может быть в некотором приближении представлен как суперпозиция двумерных синусоидальных функций. Нетрудно показать, что Фурье-преобразование синуса представляет собой два импульса, являющихся зеркальными отражениями друг друга относительно начала координат преобразования. Их положения показаны в табл. 4.3. Один из импульсов локализован в точке, отвечающей частотам синуса, другой локализован в центрально-симметричной точке. Амплитуды обоих импульсов являются чисто мнимыми (действительная часть Фурье-преобразования синуса равна нулю) и комплексно-сопряженными по отношению друг к другу. Этот вопрос будет дополнительно обсуждаться в разделах 5.4.3 и 5.4.4. В настоящий момент целью является демонстрация метода режекторной фильтрации.

На рис. 5.16(*а*), который повторяет рис. 5.5(*а*), представлено изображение, сильно искаженное синусоидальным шумом различных частот. На рис. 5.16(*б*), где представлен Фурье-спектр этого изображения, хорошо видны частотные компоненты шума в виде пар симметричных ярких точек. В рассматриваемом примере эти компоненты лежат приблизительно на окружности с центром в начале координат частотного пространства, и, таким образом, использование центрально-симметричного режекторного фильтра представляется вполне оправданным. На рис. 5.16(*в*) представлен режекторный фильтр Баттерворта



а б
в г

Рис. 5.16. (*а*) Изображение, искаженное синусоидальным шумом. (*б*) Спектр (*а*). (*в*) Режекторный фильтр Баттерворта (белым показаны точки со значением 1). (*г*) Результат фильтрации. (Изображение предоставлено NASA)

порядка 4, радиус и ширина которого выбраны так, чтобы шумовые импульсы полностью попадали в соответствующую область. Поскольку, вообще говоря, желательно удалять как можно меньшую часть Фурье-преобразования, обычно используемые режекторные фильтры являются крутыми и узкими. Результат фильтрации изображения на рис. 5.16(*а*) с помощью выбранного фильтра представлен на рис. 5.16(*г*). Улучшение изображения вполне очевидно. Использованный простой метод фильтрации позволил эффективно восстановить даже мелкие детали и текстуры на изображении. Стоит отметить также, что прямой подход, основанный на фильтрации в пространственной области с использованием масок небольшого размера, не позволяет достичь подобного результата. ■

5.4.2. Полосовые фильтры

Полосовые фильтры осуществляют операцию, противоположную режекторным фильтрам. В разделе 4.10.1 было показано, как передаточная функция полосового фильтра $H_{\text{ВР}}(u,v)$ может быть получена из передаточной функции соответствующего режекторного фильтра с передаточной функцией $H_{\text{ВР}}(u,v)$ при помощи выражения

$$H_{\text{ВР}}(u,v) = 1 - H_{\text{ВR}}(u,v). \tag{5.4-1}$$

Вывод выражений для полосовых фильтров, которые соответствуют режекторным фильтрам, приведенным в табл. 4.6, оставлен читателю в качестве упражнения (задача 5.12).

Пример 5.7. Полосовая фильтрация для выделения шумовой составляющей изображения.

■ Полосовая фильтрация обычно не используется для улучшения изображений непосредственно, поскольку, как правило, в результате ее применения слишком большая часть деталей оказывается удалена. Однако полосовая фильтрация оказывается весьма полезной для отделения тех компонентов, которые обусловлены частотными составляющими в выбранном диапазоне.

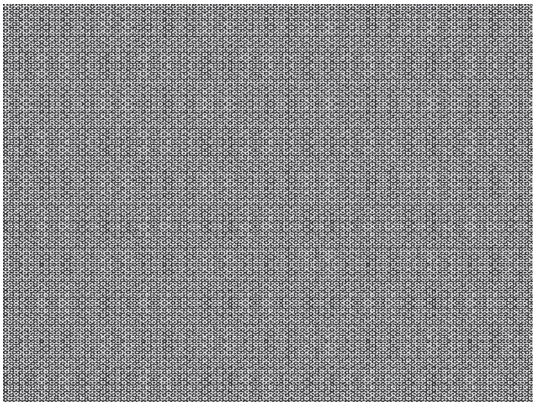


Рис. 5.17. Составляющая помехи изображения на рис. 5.16(*а*), полученная при помощи полосовой фильтрации

Рис. 5.17 иллюстрирует сказанное. Изображение на этом рисунке получено следующим образом: (1) при помощи (5.4-1) был получен полосовой фильтр, соответствующий режекторному фильтру, использованному при обработке изображения на рис. 5.16, и (2) осуществлено обратное преобразование сигнала после полосовой фильтрации. Все детали изображения оказались потеряны, однако оставшаяся информация весьма полезна, поскольку понятно, что восстановленная таким методом шумовая составляющая очень близка к шумовой составляющей, вызвавшей искажение изображения на рис. 5.16(*а*). Другими словами, полосовая фильтрация позволила выделить шумовую составляющую изображения. Это полезно, поскольку упрощает процедуру анализ шума, которая теперь может быть осуществлена в достаточной мере независимо от содержательной части изображения. ■

5.4.3. Узкополосные фильтры

Узкополосные (пропускающие) фильтры и *узкополосные режекторные* фильтры соответственно пропускают или не пропускают частоты в определенных окрестностях своих центральных частот. Уравнения для узкополосных фильтров детально рассмотрены в разделе 4.10.2. На рис. 5.18 представлены трехмерные изображения узкополосных режекторных фильтров (идеального фильтра, фильтра Баттерворта и гауссова фильтра). Из свойств симметрии преобразования Фурье следует, что для получения осмысленных (т.е. вещественных) результатов фильтрации любой фильтр должны быть симметричен относительно начала координат (центра частотного прямоугольника). Поэтому узкополосные

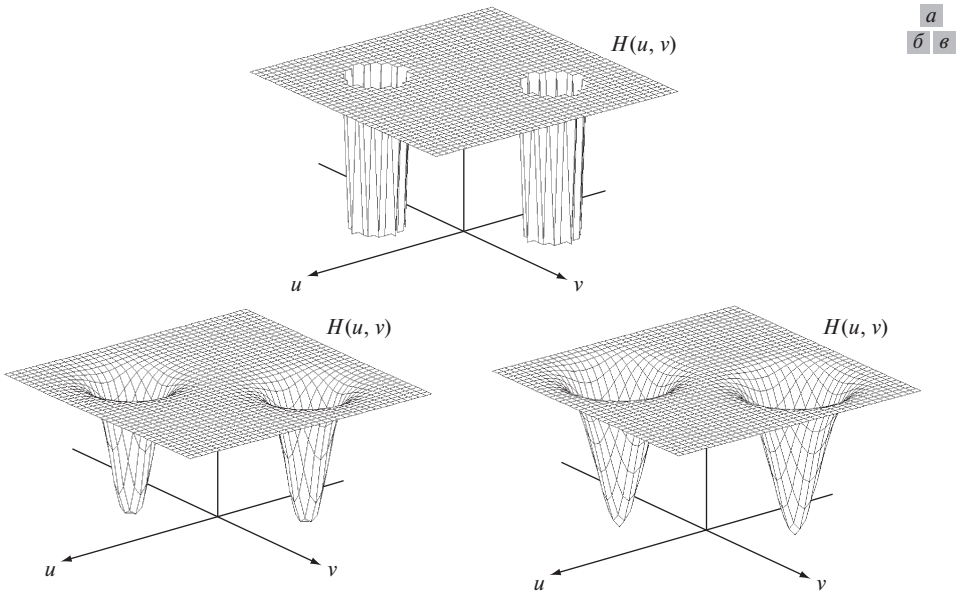


Рис. 5.18. Трехмерные изображения узкополосных режекторных фильтров: (*а*) идеальный фильтр; (*б*) фильтр Баттерворта порядка 2; (*в*) гауссов фильтр

фильтры должны иметь вид симметричных относительно начала координат пар. Единственным исключением является симметричный узкополосный фильтр с центральной частотой в начале координат. Хотя для иллюстрации на рис. 5.18 мы ограничились фильтрами, состоящими только из одной пары, количество таких пар для узкополосных фильтров, используемых на практике, может быть произвольным. Также произвольной (например прямоугольной) может быть и форма окрестностей, в которых узкополосный фильтр подавляет (пропускает) частотные составляющие.

Как показано в разделе 4.10.2, получить узкополосные фильтры, которые пропускают, а не подавляют частоты в окрестностях некоторых своих центральных частот, можно тем же способом, что и полосовые фильтры в предыдущем параграфе. Поскольку действие таких фильтров строго противоположно действию узкополосных режекторных фильтров, то их передаточные функции могут быть заданы выражением

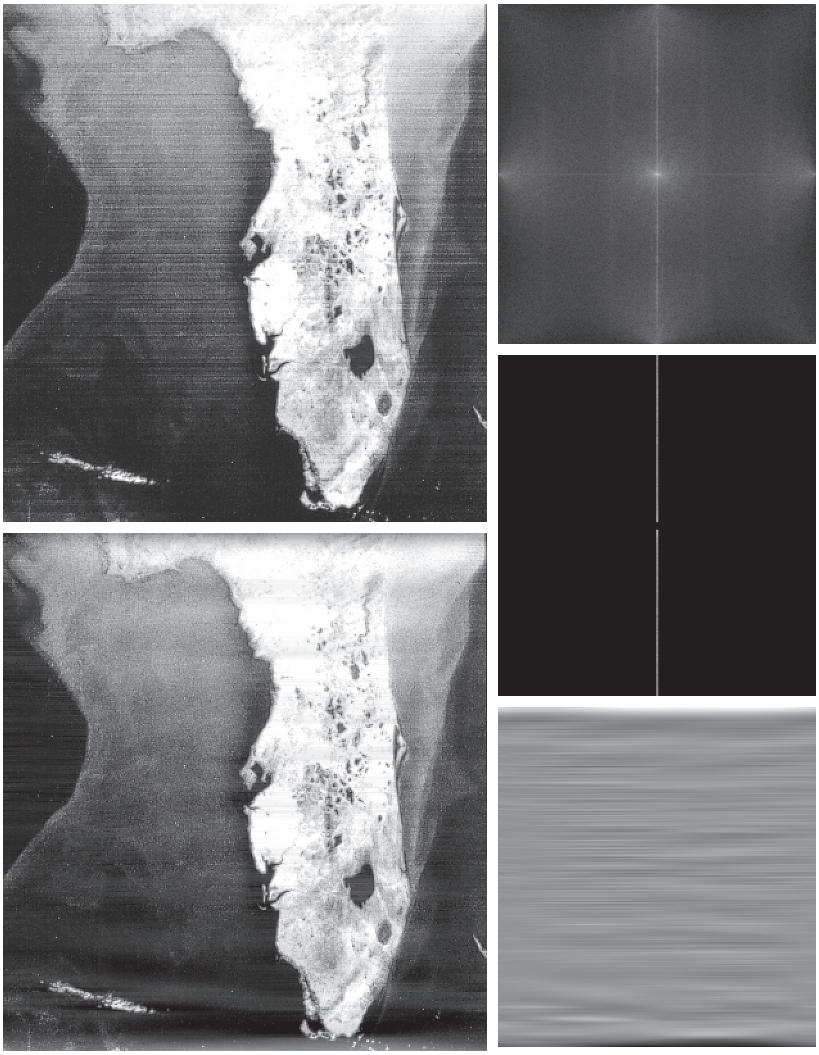
$$H_{\mathrm{NP}}(u,v) = 1 - H_{\mathrm{NP}}(u,v), \tag{5.4-2}$$

где $H_{\mathrm{NP}}(u,v)$ — передаточная функция узкополосного (пропускающего) фильтра, соответствующего узкополосному режекторному фильтру с передаточной функцией $H_{\mathrm{NR}}(u,v)$.

Пример 5.8. Удаление периодического шума с помощью узкополосной фильтрации.

■ На рис. 5.19(*а*) представлено то же изображение, что и на рис. 4.51(*а*). Представленный ниже подход, основанный на использовании узкополосных фильтров, позволяет уменьшить шум на изображении, не приводя при этом к заметному сглаживанию, которое наблюдалось в разделе 4.8.4. За исключением тех случаев, когда по указанным в том разделе причинам необходимо осуществить сглаживание изображения, использование узкополосной фильтрации предпочтительно, если только удается подобрать подходящий фильтр.

Непосредственное рассмотрение картины шума на рис. 5.19(*а*), состоящего из почти горизонтальных линий, позволяет предположить, что частотная составляющая шума локализована вдоль вертикальной оси координат в частотной области. Однако при рассмотрении спектра на рис. 5.19(*б*) становится ясно, что влияние шума недостаточно для появления в частотной области хорошо заметной шумовой составляющей вдоль вертикальной оси. Для того чтобы составить представление о том, какой вклад вносит шум в изображение, мы построим простой идеальный узкополосный фильтр, пропускающий лишь частоты вблизи вертикальной оси координат в частотном пространстве, как показано на рис. 5.19(*в*). Изображение шумовой составляющей в пространственной области (полученное как результат фильтрации с использованием указанного узкополосного фильтра) представлено на рис. 5.19(*г*). Эта картина шума хорошо соответствует картине шума на рис. 5.19(*а*). Построив, таким образом, подходящий узкополосный фильтр, который с разумной степенью точности выделяет шумовую составляющую, мы можем получить соответствующий ему узкополосный режекторный фильтр по формуле (5.4-2). Результат обработки изображения при помощи этого узкополосного режекторного фильтра представлен



а *б* *в* *д* *в*

Рис. 5.19. (*а*) Полученное со спутника изображение шт. Флорида и Мексиканского залива, на котором наблюдаются помехи в виде полос вдоль линий сканирования. (*б*) Спектр изображения (*а*). (*в*) Узкополосный фильтр, накладываемый на (*б*). (*г*) Пространственное изображение помехи. (*д*) Результат узкополосной режекторной фильтрации. (Исходное изображение предоставлено NOAA)

на рис. 5.19(*д*). Это последнее изображение содержит значительно меньше шума (заметных горизонтальных линий), чем исходное изображение. ■

5.4.4. Оптимальная узкополосная фильтрация

Другой пример периодических искажений такого рода представлен на рис. 5.20(*а*), который содержит цифровое изображение поверхности Марса,

а б

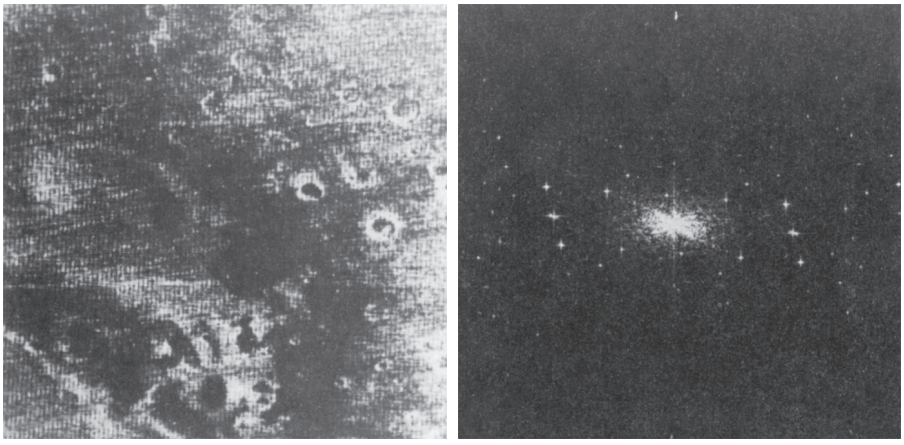


Рис. 5.20. (*а*) Изображение поверхности Марса, полученное с космического аппарата «Маринер-6». (*б*) Фурье-спектр, указывающий на наличие периодических помех. (Исходное изображение предоставлено NASA)

полученное космическим аппаратом «Маринер-6». Структура помех на этом изображении довольно похожа на аналогичную структуру на рис. 5.16(*а*), однако сами помехи значительно менее различимы и, следовательно, труднее поддаются обнаружению в частотной области. На рис. 5.20(*б*) представлен Фурье-спектр рассматриваемого изображения. Появление похожих по форме на звезды частотных компонент связано с интерференцией. Наличие в спектре нескольких центрально-симметрично локализованных пар частотных составляющих свидетельствует о том, что помехи содержат более чем одну периодическую компоненту.

Когда помехи содержат несколько составляющих, рассмотренные в предыдущих параграфах методы не всегда применимы, поскольку их использование может привести к потере слишком большого количества информации на изображении в процессе фильтрации (что особенно нежелательно, когда изображения являются уникальными и/или их получение связано с большими материальными затратами). Кроме того, как правило, частотные составляющие помех не являются узко локализованными вблизи некоторых отдельных точек в частотной области. Напротив, каждой такой составляющей обычно отвечает достаточно широкая область в частотном пространстве, которая содержит соответствующую информацию. Нахождение этих областей обычными методами Фурье-анализа далеко не всегда является простой задачей. Альтернативные методы фильтрации, позволяющие уменьшить эффекты, связанные с такими искажениями, весьма полезны во многих прикладных задачах. Обсуждаемый ниже метод является оптимальным в том смысле, что он минимизирует значения локальной дисперсии восстановленного изображения $\hat{f}(x,y)$.

Метод состоит в том, чтобы сначала получить в виде отдельного изображения основной вклад, привносимый помехой, а затем вычесть из исходного искаженного изображения некоторую непостоянную весовую долю полученного изображения помехи. Хотя мы разовьем наш метод в рамках конкретного приложения, используемый нами подход является достаточно общим и применим

в других задачах восстановления, когда приходится иметь дело с периодическими помехами сложной структуры.

Первый шаг состоит в выделении основных частотных составляющих помехи. Как и ранее, это может быть осуществлено при помощи узкополосного фильтра $H_{NP}(u,v)$, который пропускает частоты в окрестностях каждого связанного с помехой пика. Коль скоро данный фильтр выбран таким образом, чтобы пропускать только связанные с помехой частотные компоненты, то Фурье-преобразование шумовой составляющей (помехи) дается выражением

$$N(u,v) = H_{NP}(u,v)G(u,v), \tag{5.4-3}$$

где, как обычно, $G(u,v)$ обозначает Фурье-преобразование искаженного изображения.

Построение фильтра $H_{NP}(u,v)$ требует принятия важного решения о том, является ли каждый конкретный пик в частотной области шумовым пиком (т. е. пиком, связанным с помехой) или нет. По этой причине узкополосный фильтр подбирается, как правило, интерактивно на основе визуального анализа спектра изображения $G(u,v)$. После того, как конкретный фильтр выбран, соответствующее изображение шума (помех) в пространственной области может быть получено следующим образом:

$$\eta(x,y) = \mathfrak{F}^{-1}\left\{H_{NP}(u,v)G(u,v)\right\}. \tag{5.4-4}$$

Мы исходим из предположения, что искаженное изображение $g(x,y)$ получается из неискаженного изображения $f(x,y)$ прибавлением помехи $\eta(x,y)$. Поэтому, если бы мы точно знали функцию $\eta(x,y)$, то получение функции $f(x,y)$ представляло бы собой простейшую задачу, заключающуюся в вычитании $\eta(x,y)$ из $g(x,y)$. Проблема, разумеется, состоит в том, что фильтрация обычно позволяет получить лишь некоторое приближение к функции, определяющей связанную с помехой составную часть изображения. Эффект, связанный с отличием построенного приближения $\eta(x,y)$ от реально существующей помехи, может быть уменьшен, если при построении приближения для неискаженного изображения $f(x,y)$ мы вычтем из искаженного изображения $g(x,y)$ некоторую *взвешенную* долю функции $\eta(x,y)$:

$$\hat{f}(x,y) = g(x,y) - w(x,y)\eta(x,y), \tag{5.4-5}$$

где, как и ранее, $\hat{f}(x,y)$ обозначает приближение для $f(x,y)$, а $w(x,y)$ — подлежащая определению функция. Функция $w(x,y)$ называется *весовой функцией* или *функцией модуляции*, и задача метода состоит в таком выборе этой функции, чтобы результат оказался в некотором смысле оптимальным. Один из критериев выбора функции $w(x,y)$ заключается в том, чтобы величина локальной дисперсии получаемого приближения $\hat{f}(x,y)$ по заданной окрестности принимала минимальное значение в каждой точке (x,y).

Рассмотрим окрестность некоторой точки (x,y) размерами $(2a + 1)\times(2b + 1)$. Локальная дисперсия функции $\hat{f}(x,y)$ в точке с координатами (x,y) может быть получена следующим образом:

$$\sigma^2(x,y) = \frac{1}{(2a+1)(2b+1)}\sum_{s=-a}^{a}\sum_{t=-b}^{b}\left[\hat{f}(x+s,y+t) - \bar{\hat{f}}(x,y)\right]^2, \tag{5.4-6}$$

где $\bar{\hat{f}}(x,y)$ — среднее значение функции $\hat{f}$ по окрестности, т. е.

$$\bar{\hat{f}}(x,y)=\frac{1}{(2a+1)(2b+1)}\sum_{s=-a}^{a}\sum_{t=-b}^{b}\hat{f}(x+s,y+t). \tag{5.4-7}$$

При обработке точек на границе или около нее можно рассматривать неполные окрестности или считать, что изображение расширяется нулевой рамкой необходимой ширины.

Подставляя (5.4-5) в (5.4-6), получаем

$$\sigma^2(x,y)=\frac{1}{(2a+1)(2b+1)}\sum_{s=-a}^{a}\sum_{t=-b}^{b}\left\{\left[g(x+s,y+t)-w(x+s,y+t)\eta(x+s,y+t)\right]-\right.$$
$$\left.-\left[\bar{g}(x,y)-\overline{w(x,y)\eta(x,y)}\right]\right\}^2. \tag{5.4-8}$$

Предположим, что функция $w(x,y)$ практически постоянна в пределах окрестности, т. е.

$$w(x+s,y+t)=w(x,y) \tag{5.4-9}$$

при $-a\le s\le a$ и $-b\le s\le b$. При этом в окрестности будет иметь место равенство

$$\overline{w(x,y)\eta(x,y)}=w(x,y)\bar{\eta}(x,y). \tag{5.4-10}$$

С учетом двух последних формул (5.4-8) примет вид

$$\sigma^2(x,y)=\frac{1}{(2a+1)(2b+1)}\sum_{s=-a}^{a}\sum_{t=-b}^{b}\left\{\left[g(x+s,y+t)-w(x,y)\eta(x+s,y+t)\right]-\right.$$
$$\left.-\left[\bar{g}(x,y)-w(x,y)\bar{\eta}(x,y)\right]\right\}^2. \tag{5.4-11}$$

Для того чтобы найти функцию $w(x,y)$, на которой реализуется экстремум (минимум) функционала $\sigma^2(x,y)$, заданного формулой (5.4-11), нужно решить уравнение[8]

$$\frac{\partial\sigma^2(x,y)}{\partial w(x,y)}=0 \tag{5.4-12}$$

относительно $w(x,y)$. Искомое решение имеет вид

$$w(x,y)=\frac{\overline{g(x,y)\eta(x,y)}-\bar{g}(x,y)\bar{\eta}(x,y)}{\overline{\eta^2}(x,y)-\bar{\eta}^2(x,y)}. \tag{5.4-13}$$

Для того чтобы получить восстановленное изображение $\hat{f}(x,y)$, нужно вычислить функцию $w(x,y)$ по (5.4-13), а затем использовать (5.4-5). Поскольку мы предполагаем, что функция $w(x,y)$ является постоянной в пределах окрестности, то нет необходимости вычислять значения этой функции для всех точек изображения. Вместо этого можно вычислить по *одному* значению $w(x,y)$ в некоторой точке каждого из непересекающихся его фрагментов (предпочтительно в центральной точке фрагмента), а затем использовать это значение при обработке всех точек изображения, содержащихся в этом фрагменте.

---

[8] Это, по существу, уравнение Эйлера (необходимое условие экстремума) для функционала (5.4- 11), т. е. условие обращения в нуль его вариации. Координаты (x, y) можно рассматривать в этом уравнении как параметры. — *Прим. перев.*

Пример 5.9. Восстановление с помощью оптимальной узкополосной фильтрации.

■ Рис. 5.21—5.23 иллюстрируют процесс применения описанной выше техники восстановления к изображению на рис. 5.20(*а*). Это изображение имеет размеры 512×512 пикселей и параметры, определяющие размеры окрестности в процедуре оптимизации, были выбраны следующим образом: $a = b = 15$. На рис. 5.21 представлен Фурье-спектр искаженного изображения. В этом конкретном случае спектр не подвергался процедуре центрирования, поэтому начало координат $u = v = 0$ находится в левом верхнем углу изображения на рис. 5.21. На рис. 5.22(*а*)

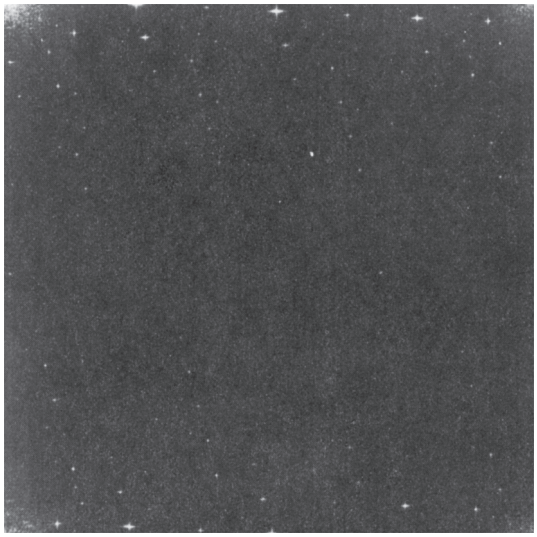


Рис. 5.21. Фурье-спектр (не центрированный) изображения, представленного на рис. 5.20(*а*). (Изображение предоставлено NASA)

а *б*

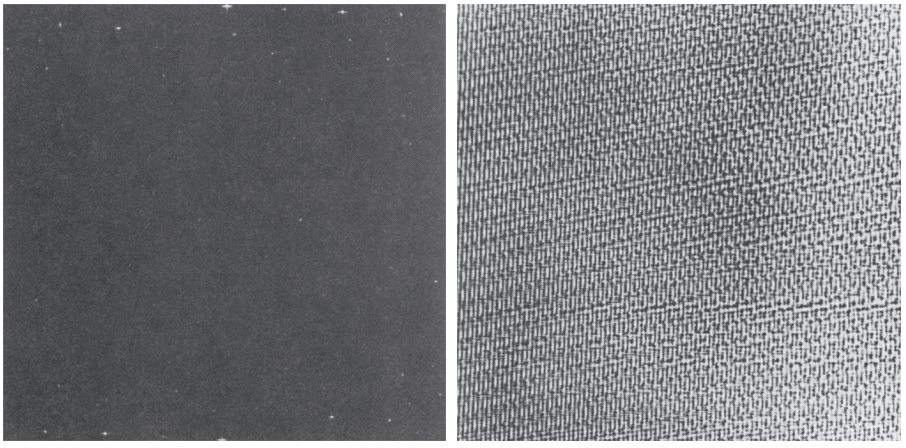


Рис. 5.22. (*а*) Фурье-спектр $N(u,v)$ и (*б*) соответствующее изображение помех (шумовой составляющей) $\eta(x,y)$. (Изображение предоставлено NASA)

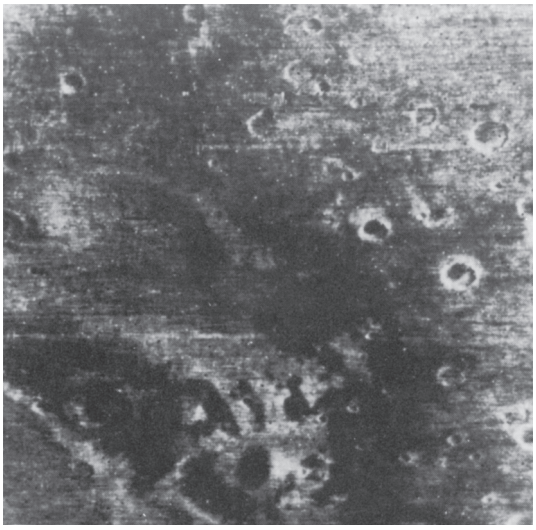


Рис. 5.23. Восстановленное изображение. (Изображение предоставлено NASA)

представлен спектр шума $N(u,v)$, т.е. на изображении присутствуют лишь те пики, которые связаны с шумовой составляющей. На рис. 5.22(*б*) представлено изображение помех $\eta(x,y)$, которое получено вычислением обратного преобразования Фурье от функции $N(u,v)$. Обратите внимание на сходство структуры этого изображения и структуры шумовой составляющей на рис. 5.20(*а*). Наконец, на рис. 5.23 представлен результат восстановления, полученный с использованием выражения (5.4-13). Периодическая помеха практически устранена. ■

5.5. Линейные трансляционно–инвариантные искажения

Искажающее преобразование на рис. 5.1, которому подвергается функция $f(x,y)$ на этапе, предшествующем восстановлению, может быть записано в виде

$$g(x,y) = H\left[f(x,y)\right] + \eta(x,y). \tag{5.5-1}$$

Временно предположим, что $\eta(x,y) = 0$, так что $g(x,y) = H[f(x,y)]$. Как было определено в разделе 2.6.2, оператор H является *линейным*, если

$$H\left[af_1(x,y) + bf_2(x,y)\right] = aH\left[f_1(x,y)\right] + bH\left[f_2(x,y)\right], \tag{5.5-2}$$

где a и b — любые скаляры, а $f_1(x,y)$ и $f_2(x,y)$ — две любые функции (два любых изображения). В случае $a = b = 1$, равенство (5.5-2) принимает вид

$$H\left[f_1(x,y) + f_2(x,y)\right] = H\left[f_1(x,y)\right] + H\left[f_2(x,y)\right]. \tag{5.5-3}$$

Выражаемое равенством (5.5-3) свойство называется свойством *аддитивности*. Это свойство просто означает, что если H — линейный оператор, то результат его действия на сумму двух функций равен сумме результатов его действия на каждую из этих функций.

Для ознакомления с теорией линейных систем обратитесь к интернет-сайту книги.

В случае $f_2(x,y) = 0$ равенство (5.5-2) принимает вид

$$H\left[af_1(x,y)\right] = aH\left[f_1(x,y)\right]. \tag{5.5-4}$$

Выражаемое равенством (5.5-4) свойство называется свойством *однородности*. Это свойство означает, что результат действия оператора на произведение константы на функцию равен произведению этой константы на результат действия оператора на функцию. Таким образом, линейный оператор обладает как свойством аддитивности, так и свойством однородности.

Оператор, действующий по правилу $g(x,y) = H[f(x,y)]$, называется *трансляционно-инвариантным* (или *пространственно-инвариантным*), если для любой функции $f(x,y)$ и для любых чисел α и β выполняется равенство

$$H\left[f(x-\alpha, y-\beta)\right] = g(x-\alpha, y-\beta). \tag{5.5-5}$$

В соответствии с данным определением действие оператора в точке зависит лишь от *значения* аргумента и не зависит от *местоположения* этой точки в пространстве.

Небольшое видоизменение формы записи[9] определения (4.5-3) дискретной импульсной функции дает возможность выразить функцию $f(x,y)$ в терминах непрерывной[10] импульсной функции (δ-функции):

$$f(x,y) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(\alpha,\beta)\delta(x-\alpha, y-\beta)\,d\alpha\,d\beta. \tag{5.5-6}$$

Снова временно предположим, что $\eta(x,y) = 0$. Тогда подстановка (5.5-6) в (5.5-1) дает

$$g(x,y) = H\left[f(x,y)\right] = H\left[\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(\alpha,\beta)\delta(x-\alpha, y-\beta)\,d\alpha\,d\beta\right]. \tag{5.5-7}$$

Пусть H — линейный оператор. Поскольку свойство аддитивности распространяется на интегралы, то

$$g(x,y) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} H\left[f(\alpha,\beta)\delta(x-\alpha, y-\beta)\right]d\alpha\,d\beta. \tag{5.5-8}$$

Используя свойство однородности и учитывая, что $f(\alpha,\beta)$ не зависит от x и y, получаем

$$g(x,y) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(\alpha,\beta)\,H\left[\delta(x-\alpha, y-\beta)\right]d\alpha\,d\beta. \tag{5.5-9}$$

---

[9] Для того чтобы упростить изложение материала, автору удобно перейти к рассмотрению непрерывного случая, что не изменяет существа дела. Такой переход осуществляется заменой дискретных переменных на непрерывные, конечные суммы заменяются на соответствующие интегралы, а дискретная δ-функция (символ Кронекера) — на непрерывную δ-функцию. Обратные замены позволяют вернуться к дискретному случаю. — *Прим. перев.*

[10] Касательно непрерывных и дискретных переменных см. примечание 23 в данной главе.

Функция под знаком интеграла в правой части последнего равенства

$$h(x,\alpha,y,\beta)=H\left[\delta(x-\alpha,y-\beta)\right] \tag{5.5-10}$$

называется *импульсным откликом* (*импульсной характеристикой*) или *ядром* оператора *H*. Таким образом, функция $h(x,\alpha,y,\beta)$ представляет собой результат действия (отклик) оператора *H* на δ-функцию, локализованную в точке с координатами (x,y). В оптике, когда импульс соответствует светящейся точке, функцию $h(x,\alpha,y,\beta)$ обычно называют *функцией рассеяния точки* (ФРТ). Происхождение термина связано с тем обстоятельством, что любая реальная оптическая система до некоторой степени размывает (рассеивает) светящуюся точку, причем величина рассеивания определяется качеством оптической системы.

Подставив (5.5-10) в (5.5-9), получим выражение

$$g(x,y)=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(\alpha,\beta)h(x,\alpha,y,\beta)\,d\alpha\,d\beta\,, \tag{5.5-11}$$

которое называется *интегралом Фредгольма первого рода*. В последнем выражении заключен фундаментальный результат, лежащий в основе теории линейных систем. Этот результат устанавливает, что если известен отклик системы на импульсную функцию, то отклик системы на *любую* функцию $f(\alpha,\beta)$ может быть вычислен на основе (5.5-11). Другими словами, любая линейная система *H* полностью характеризуется своим импульсным откликом (ядром соответствующего оператора).

Если оператор *H* является трансляционно-инвариантным, то из (5.5-5)следует, что

$$H\left[\delta(x-\alpha,y-\beta)\right]=h(x-\alpha,y-\beta)\,. \tag{5.5-12}$$

Выражение (5.5-11) в этом случае принимает вид

$$g(x,y)=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(\alpha,\beta)h(x-\alpha,y-\beta)\,d\alpha\,d\beta\,. \tag{5.5-13}$$

Это выражение является *интегралом свертки*, введенным для одной переменной уравнением (4.2-20) и расширенным для двумерного случая в задаче 4.11. Выражение (5.5-13) показывает, что, зная ядро линейного оператора, можно вычислить результат *g* его действия на *любую* функцию *f*. Этот результат просто представляет собой свертку ядра с соответствующей функцией.

При наличии аддитивного шума выражение, определяющее линейную модель искажений (см. (5.5-11)), принимает вид

$$g(x,y)=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(\alpha,\beta)h(x,\alpha,y,\beta)\,d\alpha\,d\beta+\eta(x,y)\,. \tag{5.5-14}$$

Если оператор *H* трансляционно-инвариантный, то (5.5-14) записывается в виде

$$g(x,y)=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(\alpha,\beta)h(x-\alpha,y-\beta)\,d\alpha\,d\beta+\eta(x,y)\,. \tag{5.5-15}$$

Значения описывающего шум слагаемого $\eta(x,y)$ являются случайными величинами, которые предполагаются не зависящими от точки пространства. Используя привычные обозначения для свертки, можно переписать (5.5-15) как

$$g(x,y) = h(x,y) \star f(x,y) + \eta(x,y) \qquad (5.5\text{-}16)$$

или, используя теорему о свертке (см. раздел 4.6.6), перейти в частотную область, что дает

$$G(u,v) = H(u,v)F(u,v) + N(u,v). \qquad (5.5\text{-}17)$$

Последние два выражения суть выражения (5.1-1) и (5.1-2). Напомним, что умножение понимается как поэлементное. Так, элемент произведения $H(u,v)F(u,v)$ с номером ij есть произведение ij-того элемента $H(u,v)$ на ij-тый элемент $F(u,v)$.

Итак, проведенное рассмотрение показывает, что воздействие линейной трансляционно-инвариантной искажающей системы с аддитивным шумом может быть смоделировано в пространственной области как свертка искажающей функции (ядра искажающего оператора) с изображением и последующее прибавление аддитивного шума. На основе теоремы о свертке то же воздействие может быть выражено в частотной области как произведение Фурье-преобразований изображения и искажающей функции с последующим прибавлением Фурье-преобразования шума. Для перехода в частотную область мы используем рассмотренный в разделе 4.11 алгоритм БПФ. Будем иметь в виду также, что применение дискретного преобразования Фурье требует дополнения функций нулями, как это объяснялось в разделе 4.6.6.

Линейные трансляционно-инвариантные модели могут быть использованы для приближенного описания многих типов искажений. Преимущество такого подхода заключается в том, что огромное количество используемых в линейной теории методов и средств становятся применимы для решения задач восстановления изображений. Хотя нелинейные и трансляционно-неинвариантные методы являются более общими (и обычно более точными), но их использование часто приводит к непреодолимым или очень трудно решаемым численными методами проблемам. Рассмотрения настоящей главы сосредоточены на линейных трансляционно-инвариантных методах. Поскольку искажение представляет собой результат свертки, то для восстановления необходимо найти такой фильтр, применение которого приводило бы к обратному процессу. Поэтому для обозначения линейного процесса восстановления часто используется термин *реконструкция* (*деконволюция*[11]) *изображений*. Аналогично фильтры, используемые для восстановления, часто называются *реконструирующими фильтрами*.

5.6. Оценка искажающей функции

Существуют три основных способа оценки искажающей функции (ядра искажающего оператора) для последующего ее использования при восстановле-

[11] Процесс, обратный свертке («convolution»), по-английски называется «deconvolution» — «деконволюция». В дальнейшем мы предпочитаем вместо этого термина использовать термин «реконструкция». — *Прим. перев.*

нии изображений: (1) визуальный анализ, (2) эксперимент и (3) математическое моделирование. Вследствие того, что истинная искажающая функция нечасто бывает известна полностью, процесс восстановления изображения с использованием приближения искажающей функции, полученного некоторым образом, иногда называют реконструкцией «вслепую».

5.6.1. Оценка на основе визуального анализа изображения

Предположим, что имеется искаженное изображение, но информация об искажающей функции H отсутствует. Основываясь на предположении, что искажающий изображение процесс являлся линейным и пространственно-инвариантным, один из способов оценить эту функцию состоит в выделении информации непосредственно из изображения. Например, если изображение является размытым, мы можем рассмотреть его небольшой прямоугольный фрагмент, содержащий примеры структуры, такие как часть некоторого объекта и фон. Для того чтобы уменьшить влияние шума, следует выбрать ту область изображения, которая содержит полезный сигнал большой амплитуды (т. е. фрагмент высокого контраста). Следующий этап состоит в такой обработке выбранного фрагмента изображения, чтобы насколько возможно максимально убрать размытость. Например, для этого можно использовать повышение резкости с помощью соответствующих фильтров или даже обработать небольшую область вручную.

Обозначим рассматриваемую часть изображения как $g_s(x, y)$ и обрабатываемый фрагмент (который в действительности представляет собой наше приближение для части неискаженного изображения в рассматриваемой области) как $\hat{f}_s(x, y)$. Далее, предполагая, что влияние шума пренебрежимо мало в силу нашего выбора области с большим полезным сигналом, на основании (5.5-17) имеем

$$H_s(u,v) = \frac{G_s(u,v)}{F_s(u,v)}. \tag{5.6-1}$$

Исходя из свойств функции $H_s(u,v)$ и опираясь на предположение о трансляционной инвариантности искажений, мы теперь можем вывести полную искажающую функцию $H(u,v)$. Предположим, например, что радиальный профиль функции $H_s(u,v)$ приблизительно совпадает с формой гауссовой кривой. Это может быть использовано для построения функции $H(u,v)$ той же самой формы, но большего размера. Затем $H(u,v)$ используется одним из методов восстановления, которые будут рассмотрены в последующих разделах. Ясно, что это является достаточно трудоемким процессом, применяющимся лишь в исключительных обстоятельствах, например таких, как восстановление старых фотографий исторического значения.

5.6.2. Оценка на основе эксперимента

Если оборудование, аналогичное тому, которое использовалось при получении изображения, доступно, то в принципе возможно получить точную оценку искажающей функции. Сначала необходимо так подобрать параметры системы, чтобы искажения на получаемых с ее помощью изображениях, похожих по сце-

нарию на подлежащее восстановлению изображение, как можно лучше соответствовали искажениям на этом изображении. Далее идея состоит в том, чтобы сформировать импульсный отклик (ядро искажающего оператора), для чего нужно получить изображение импульса (маленькой яркой точки), используя систему с подобранными значениями параметров. Как было отмечено в разделе 5.5, линейная трансляционно-инвариантная система определяется полностью своим импульсным откликом.

Импульс симулируется яркой световой точкой. Чтобы уменьшить влияние шума, яркость должна быть как можно больше. Затем, учитывая, что Фурье-преобразование импульса есть константа, из (5.5-17) получаем

$$H(u,v)=\frac{G(u,v)}{A}, \tag{5.6-2}$$

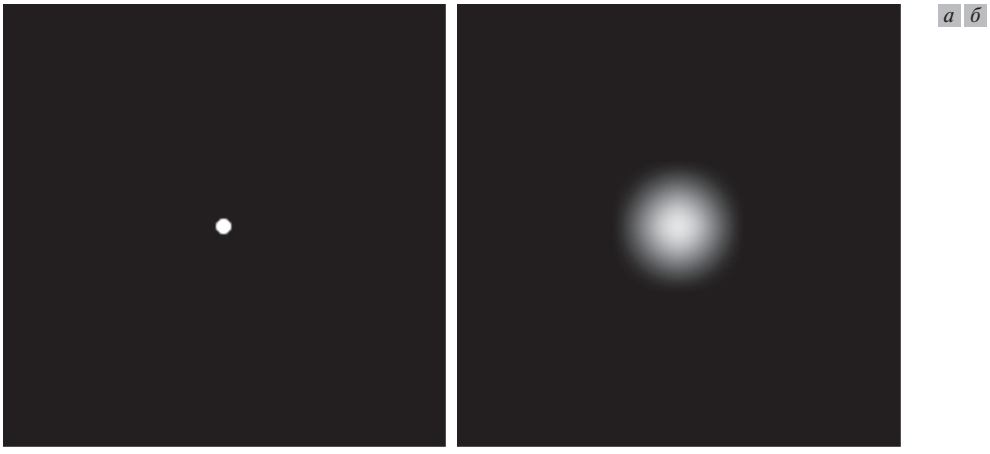
где, как и раньше, $G(u,v)$ — Фурье-преобразование полученного изображения, A — константа, описывающая величину яркости импульса. На рис. 5.24 приведен соответствующий пример.

5.6.3. Оценка на основе моделирования

Моделирование искажений используется уже в течение многих лет, так как оно позволяет проникнуть в суть задачи восстановления изображений. В некоторых случаях модель позволяет даже учесть внешние условия, которые вызывают искажения. Например, в основе предложенной в работе [Hufnagel, Stanley, 1964] модели искажений лежит учет таких физических свойств атмосферы, как турбулентность. Эта модель имеет следующий знакомый вид:

$$H(u,v)=e^{-k(u^2+v^2)^{5/6}}, \tag{5.6-3}$$

где константа k описывает турбулентные свойства атмосферы. С точностью до коэффициента 5/6 в показателе экспоненты это выражение совпадает по фор-



а б

Рис. 5.24. Оценка искажающей функции с помощью импульса. (*а*) Световой импульс (показан с увеличением). (*б*) Изображение (искаженного) импульса

ме с выражением для гауссова низкочастотного фильтра, который рассматривался в разделе 4.8.3. В действительности гауссовы ФНЧ иногда используются для моделирования умеренной однородной расфокусировки. На рис. 5.25 представлены примеры, полученные в результате имитации расфокусировки изображения с использованием выражения (5.6-3), в котором параметр k принимал значения $k = 0{,}0025$ (в данном случае сильная турбулентность), $k = 0{,}001$ (умеренная турбулентность) и $k = 0{,}00025$ (слабая турбулентность). Размеры изображений — 480×480 пикселей[12].

Другим важным аспектом моделирования является построение математической модели непосредственно из основных принципов. В качестве иллюстрации мы детально рассмотрим случай, когда размывание (смазывание) возникает

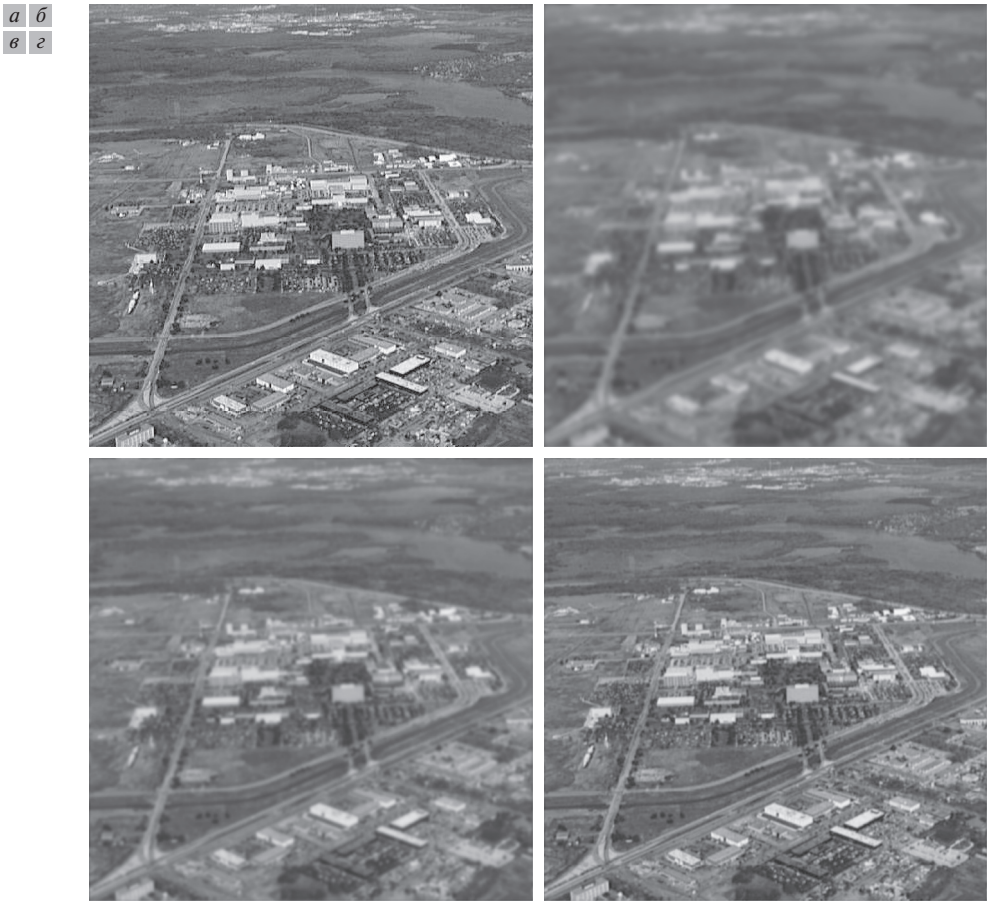


Рис. 5.25. Результаты моделирования турбулентности атмосферы. (*а*) Турбулентность пренебрежимо мала. (*б*) Сильная турбулентность, $k = 0{,}0025$. (*в*) Умеренная турбулентность, $k = 0{,}001$. (*г*) Слабая турбулентность, $k = 0{,}00025$. (Исходное изображение предоставлено NASA)

[12] Кроме того, при генерации изображений был добавлен небольшой по величине аддитивный шум (см. пример 5.15), который практически незаметен, но играет важную роль в рассматриваемых ниже процедурах восстановления. — *Прим. перев.*

в результате равномерного поступательного движения изображения сцены относительно регистрирующей системы в процессе фотосъемки. Предположим, что изображение $f(x,y)$ участвует в плоском движении и что функции $x_0(t)$ и $y_0(t)$ определяют закон движения[13] в направлениях x и y соответственно. Полная экспозиция в любой точке записывающего носителя (скажем, пленки или матрицы сенсоров) определяется как интеграл по времени (т. е. по времени, в течение которого открыт затвор регистрирующей системы) от величины мгновенной экспозиции.

Предположим, что затвор системы открывается и закрывается мгновенно и что за исключением эффектов, связанных с движением, процесс регистрации изображения является идеальным. Тогда если T — время экспозиции, то

$$g(x,y)=\int_0^T f\left(x-x_0(t),y-y_0(t)\right)dt, \tag{5.6-4}$$

где $g(x,y)$ — смазанное изображение.

В соответствии с (4.5-7) Фурье-преобразование (5.6-4) имеет вид

$$\begin{aligned} G(u,v) &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} g(x,y)e^{-i2\pi(ux+vy)}dxdy = \\ &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\left[\int_0^T f\left(x-x_0(t),y-y_0(t)\right)dt\right]e^{-i2\pi(ux+vy)}dxdy. \end{aligned} \tag{5.6-5}$$

Изменение порядка интегрирования позволяет записать (5.6-5) в виде

$$G(u,v)=\int_0^T\left[\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f\left(x-x_0(t),y-y_0(t)\right)e^{-i2\pi(ux+vy)}dxdy\right]dt. \tag{5.6-6}$$

Член внутри квадратных скобок представляет собой Фурье-преобразование сдвинутой функции $f(x-x_0(t), y-y_0(t))$. Используя (4.6-4), имеем

$$G(u,v)=\int_0^T F(u,v)e^{-i2\pi(ux_0(t)+vy_0(t))}dt=F(u,v)\int_0^T e^{-i2\pi(ux_0(t)+vy_0(t))}dt, \tag{5.6-7}$$

причем второе равенство имеет место, поскольку функция $F(u,v)$ не зависит от переменной t.

Положив по определению

$$H(u,v)=\int_0^T e^{-i2\pi(ux_0(t)+vy_0(t))}dt, \tag{5.6-8}$$

мы можем переписать (5.6-7) в стандартном виде

$$G(u,v)=H(u,v)F(u,v). \tag{5.6-9}$$

Если функции $x_0(t)$ и $y_0(t)$, определяющие закон движения изображения, известны, то передаточная функция $H(u,v)$ может быть получена прямо из (5.6-8). Предположим, например, что рассматриваемое изображение участвует в равномерном поступательном движении только в x-направлении со скоростью, кото-

[13] Заметим, что эти функции описывают закон движения изображения сцены, которое возникает в поле кадра регистрирующей системы, а не само движение сцены в пространстве. — *Прим. перев.*

а б

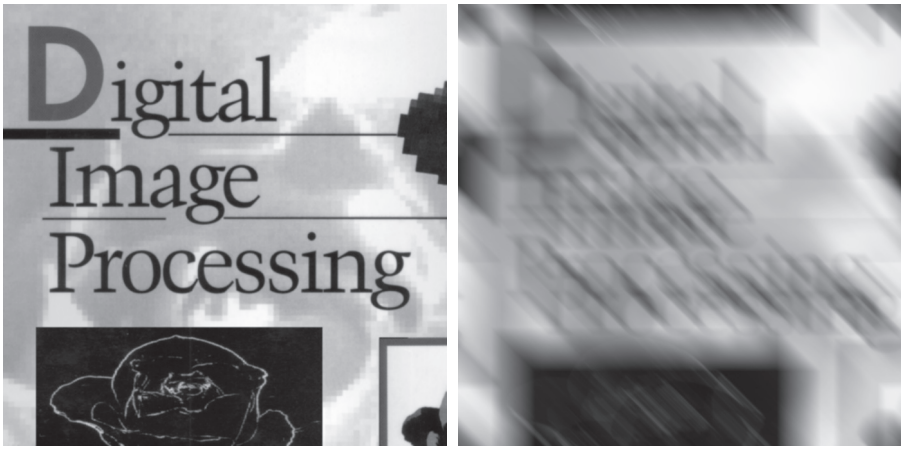


Рис. 5.26. (*а*) Исходное изображение. (*б*) Результат смазывания изображения с использованием искажающей функции вида (5.6-11) со значением параметров $a = b = 0{,}1$ и $T = 1$

рая определяется выражением $x_0(t) = at/T$. За время экспозиции T изображение смещается на общее расстояние a. Полагая в (5.6-8) $y_0(t) = 0$, имеем

$$H(u,v) = \int_0^T e^{-i2\pi u x_0(t)} dt = \int_0^T e^{-i2\pi uat/T} dt = \frac{T}{\pi ua} \sin(\pi ua) e^{-i\pi ua}. \qquad (5.6\text{-}10)$$

Легко заметить, что H обращается в нуль в точках $u = n/a$, где n — целое. Если бы мы также рассмотрели равномерное движение в направлении y вида $y_0(t) = bt/T$, то получили бы для искажающей функции следующее выражение:

$$H(u,v) = \frac{T}{\pi(ua + vb)} \sin\big(\pi(ua + vb)\big) e^{-i\pi(ua+vb)}. \qquad (5.6\text{-}11)$$

Как объяснено в конце табл. 4.3, для получения дискретного фильтра уравнение (5.6-11) дискретизуется по u и v.

Пример 5.10. Размывание (смазывание) изображения в результате движения. ■ На рис. 5.26(*б*) представлено смазанное изображение, полученное в результате вычисления Фурье-преобразования изображения на рис. 5.26(*а*), умножения этого преобразования на искажающую функцию вида (5.6-11) и вычисления обратного преобразования. Размеры изображений составляют 688×688 пикселей; параметры в (5.6-11) выбирались следующим образом: $a = b = 0{,}1$ и $T = 1$. Ряд непростых вопросов, возникающих в процессе восстановления исходного изображения по его искаженному аналогу, особенно при наличии шума, обсуждаются в разделах 5.8 и 5.9. ■

5.7. Инверсная фильтрация

В этом разделе мы сделаем первый шаг в решении задачи восстановления изображений, искаженных оператором H, ядро (искажающая функция) которого

задано или определено с помощью методов, рассмотренных в предыдущем разделе. Простейшим способом восстановления является инверсная фильтрация, которая предполагает получение оценки $\hat{F}(u,v)$ Фурье-преобразования исходного изображения делением Фурье-преобразования искаженного изображения на частотное представление искажающей функции:

$$\hat{F}(u,v)=\frac{G(u,v)}{H(u,v)}. \tag{5.7-1}$$

Деление здесь понимается как операция над соответственными элементами массивов, как это было определено в разделе 2.6.1 и в связи с уравнением (5.5-17). Подставив правую часть выражения (5.1-2) для $G(u,v)$ в (5.7-1), получим

$$\hat{F}(u,v)=F(u,v)+\frac{N(u,v)}{H(u,v)}. \tag{5.7-2}$$

Последнее выражение представляет интерес. Из него видно, что даже зная искажающую функцию, невозможно точно восстановить неискаженное изображение (обратное Фурье-преобразование функции $F(u,v)$), поскольку функция $N(u,v)$ неизвестна. Имеется и еще одна проблема. Если функция $H(u,v)$ принимает нулевые или близкие к нулевым значения, то вклад второго слагаемого в правой части (5.7-2) может стать доминирующим. Как вскоре будет видно, эта ситуация часто реализуется на практике.

Один из способов обойти указанную проблему состоит в том, чтобы ограничить частоты фильтра значениями вблизи начала координат[14]. Из обсуждения уравнения (4.2-22) известно, что значение $H(0,0)$ обычно является наибольшим значением $H(u,v)$ в частотной области. Поэтому ограничиваясь рассмотрением частот вблизи начала координат, мы уменьшаем вероятность встретить нулевое значение. Следующий пример служит иллюстрацией рассмотренного метода.

Пример 5.11. Инверсная фильтрация.

■ Представленное на рис. 5.25(*б*) изображение, искаженное турбулентностью атмосферы, было восстановлено при помощи инверсной фильтрации на основе (5.7-1) с использованием функции, являющейся обратной к функции, вызвавшей расфокусировку этого изображения. В данном случае искажающая функция была равна

$$H(u,v)=e^{-k\left((u-M/2)^2+(v-N/2)^2\right)^{5/6}}$$

со значением параметра $k = 0{,}0025$. Константы $M/2$ и $N/2$ определяют сдвиг, который центрирует функцию в частотной области так, чтобы она соответствовала центрированному Фурье-преобразованию изображения, как уже многократно обсуждалось в предыдущей главе. В данном случае $M = N = 480$. Искажающая функция не обращается в нуль, поэтому проблема деления на нуль не возникает. Однако несмотря на это, значения искажающей функции становятся при больших частотах настолько малыми, что результат полной инверсной фильтрации

[14] Т. е. считать, что второе слагаемое в (5.7 2) обращается в нуль вне некоторой области вблизи начала координат. Можно, например, умножить это слагаемое на передаточную функцию некоторого идеального низкочастотного фильтра. — *Прим. перев.*

(представленный на рис. 5.27(*а*)) совершенно бесполезен. Причины столь скверного результата были выяснены при обсуждении равенства (5.7-2).

На рис. 5.27(*б*)—(*г*) представлены результаты восстановления, полученные при обрезании значений отношения $G(u,v)/H(u,v)$ вне кругов с радиусами 40, 70 и 85 отсчетов соответственно. Обрезание осуществлялось путем умножения этого отношения на передаточную функцию низкочастотного фильтра Баттерворта порядка 10, что обеспечило быстрое (но гладкое) убывание в переходной зоне требуемого радиуса. Радиусы порядка 70 обеспечивают получение результатов наилучшего качества (см. рис. 5.27(*в*)). При меньших значениях радиуса восстановленное изображение остается расфокусированным, что и показывает изображение на рис. 5.27(*б*), полученное при значении радиуса 40. Увеличение значения радиуса свыше 70 приводит к значительному ухудшению получаемого изображения, как это видно на изображении на рис. 5.27(*г*), полученном при значении радиуса 85. Шум явно доминирует на изображении, и его содержательная

а б
в г

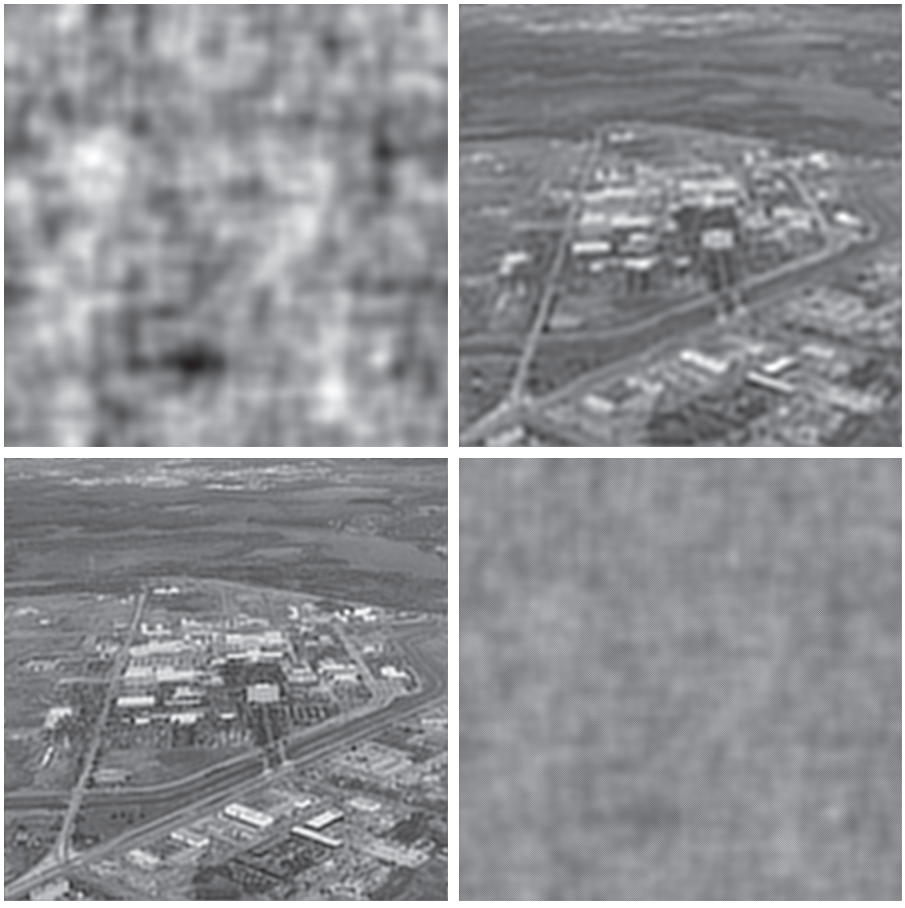


Рис. 5.27. Восстановление изображения рис. 5.25(*б*) по формуле (5.7-1). (*а*) Результат применения полного фильтра; результаты применения фильтра, обрезанного: (*б*) вне круга радиуса 40; (*в*) вне круга радиуса 70; (*г*) вне круга радиуса 85

часть едва видна из-за шумовой «завесы». При дальнейшем увеличении значения радиуса мы получаем изображения, все более и более похожие на рис. 5.27(*а*). ■

Результаты приведенного примера свидетельствуют о слабых возможностях метода инверсной фильтрации вообще. Основной темой следующих трех разделов является вопрос о том, как можно улучшить этот метод.

5.8. Фильтрация методом минимизации среднего квадрата отклонения (винеровская фильтрация)

Рассмотренный в предыдущем разделе метод инверсной фильтрации не обеспечивает корректной работы по отношению к шуму. В настоящем разделе мы рассмотрим метод, соединяющий в себе учет свойств искажающей функции и статистических свойств шума в процессе восстановления. Метод основан на рассмотрении изображений и шума как случайных переменных, и задача ставится следующим образом: найти такую оценку $\hat{f}$ для неискаженного изображения f, чтобы средний квадрат отклонения этих величин друг от друга (ошибка) было минимальным. Данная мера отклонения e задается формулой

> Заметим, что неискаженные изображения также считаются случайными переменными, в соответствии с тем, как это обсуждалось в конце раздела 2.6.8.

$$e^2 = E\left\{(f - \hat{f})^2\right\}, \tag{5.8-1}$$

где $E\{\cdot\}$ обозначает математическое ожидание своего аргумента. Предполагается, что выполнены следующие условия: (1) шум и неискаженное изображение не коррелированы между собой; (2) либо шум, либо неискаженное изображение имеют нулевое среднее значение; (3) оценка линейно зависит от искаженного изображения. При выполнении этих условий минимум среднего квадрата отклонения (5.8-1) достигается на функции, которая задается в частотной области выражением

$$\begin{aligned}
\hat{F}(u,v) &= \left(\frac{H^*(u,v)S_f(u,v)}{S_f(u,v)|H(u,v)|^2 + S_\eta(u,v)}\right)G(u,v) = \\
&= \left(\frac{H^*(u,v)}{|H(u,v)|^2 + S_\eta(u,v)/S_f(u,v)}\right)G(u,v) = \\
&= \left(\frac{1}{H(u,v)}\frac{|H(u,v)|^2}{|H(u,v)|^2 + S_\eta(u,v)/S_f(u,v)}\right)G(u,v),
\end{aligned} \tag{5.8-2}$$

причем последнее равенство имеет место в силу того, что произведение комплексного числа на комплексно-сопряженное равно квадрату модуля. Приведенный результат был получен Н. Винером [N. Wiener, 1942], и метод известен как оптимальная фильтрация по Винеру. Фильтр, представленный выражением внутри скобок, часто называют *фильтром минимального среднеквадратиче-*

ского отклонения или *винеровским фильтром*. В конце главы приведены ссылки на источники, в которых можно найти подробный вывод выражения для винеровского фильтра. Отметим, что, как видно из первой строки (5.8-2), проблема нулей в спектре искажающей функции при использовании винеровского фильтра не возникает, за исключением тех точек (u,v), при которых знаменатель обращается в нуль.

В формуле (5.8-2) использованы следующие обозначения:

$H(u,v)$ — искажающая функция (ее частотное представление);

$H^*(u,v)$ — комплексное сопряжение $H(u,v)$;

$|H(u,v)|^2 = H^*(u,v)H(u,v)$;

$S_\eta(u,v) = |N(u,v)|^2$ — энергетический спектр шума (см. уравнение (4.6-18))[15];

$S_f(u,v) = |F(u,v)|^2$ — энергетический спектр неискаженного изображения.

Как и ранее, $G(u,v)$ — Фурье-преобразование искаженного изображения. Восстановленное изображение в пространственной области получается применением обратного преобразования Фурье к оценке $\hat{F}(u,v)$. Отметим, что если шум равен нулю, то его энергетический спектр обращается в нуль, и винеровская фильтрация в этом случае сводится к инверсной фильтрации.

Многие из полезных мер основываются на энергетических спектрах шума и искаженного изображения. Одним из наиболее важных является *отношение сигнал—шум*, приближенно выражаемое через значения в частотной области как

$$\mathrm{SNR} = \frac{\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}|F(u,v)|^2}{\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}|N(u,v)|^2}. \qquad (5.8\text{-}3)$$

Это соотношение дает меру уровня информационного отношения мощности сигнала (т. е. исходного, неискаженного изображения) к уровню мощности шума. Изображения с низким шумом будут иметь большое значение SNR, а те же изображения с высоким уровнем шума — малое значение SNR. Само по себе данное соотношение имеет ограниченное значение, но оно является важной метрикой, используемой для описания характеристик алгоритмов восстановления.

Средний квадрат отклонения, приведенный в статистической форме в виде уравнения (5.8-1), в терминах суммирования значений исходного и восстановленного изображений может быть приближенно выражен следующей формулой:

$$\mathrm{MSE} = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[f(x,y) - f(x,y)]^2. \qquad (5.8\text{-}4)$$

---

[15] Выражение $|N(u,v)|^2$ также называют *автокорреляцией* шума. Эта терминология вытекает из теоремы о корреляции (первая строка пункта 7 в табл. 4.3). Если обе функции тождественны, то корреляция будет *автокорреляцией*, а правая часть становится $N^*(u,v)N(u,v)$, т. е. равной $|N(u,v)|^2$. То же самое относится и к выражению $|F(u,v)|^2$, которое является автокорреляцией изображения. Более подробно корреляция будет рассмотрена в главе 12.

В действительности, если рассматривать восстановленное изображение как сигнал, а разницу между ним и исходным — как шум, можно определить отношение сигнал—шум в пространственной области как

$$\mathrm{SNR} = \frac{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\hat{f}(x,y)^2}{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[f(x,y)-\hat{f}(x,y)]^2}. \qquad (5.8\text{-}5)$$

Чем ближе f и $\hat{f}$, тем больше будет данное отношение. Иногда используются квадратные корни этих мер, и в таком случае их соответственно называют *среднеквадратическое* (*стандартное*) *отклонение* и *квадратный корень отношения сигнал—шум*. Как уже неоднократно упоминалось, следует помнить, что численные значения метрик совсем не обязательно отражают воспринимаемое качество изображения.

Когда мы имеем дело с белым шумом, спектр которого $|N(u,v)|^2$ является постоянной функцией, происходят соответствующие упрощения. Однако спектр неискаженного изображения редко бывает известен. В тех случаях, когда спектры шума и неискаженного изображения неизвестны и не могут быть оценены, часто используется подход, состоящий в аппроксимации выражения (5.8-2) выражением

$$\hat{F}(u,v) = \left(\frac{1}{H(u,v)}\frac{|H(u,v)|^2}{|H(u,v)|^2 + K}\right)G(u,v), \qquad (5.8\text{-}6)$$

где K — определенная константа, прибавляемая ко всем значениям $|H(u,v)|^2$. В приводимых ниже примерах фильтрации используется именно это последнее выражение.

Пример 5.12. Сравнение инверсной фильтрации и винеровской фильтрации.

■ Рис. 5.28 демонстрирует преимущества винеровской фильтрации по сравнению с инверсной фильтрацией. На рис. 5.28(*а*) воспроизведен представлен-

а б в

Рис. 5.28. Сравнение инверсной фильтрации и винеровской фильтрации. (*а*) Результат восстановления с использованием полного инверсного фильтра. (*б*) Результат восстановления с использованием обрезанного инверсного фильтра. (*в*) Результат винеровской фильтрации

ный на рис. 5.27(*а*) результат восстановления с использованием полного инверсного фильтра. Аналогично на рис. 5.28(*б*) воспроизведен представленный на рис. 5.27(*в*) результат восстановления с использованием инверсного фильтра, обрезанного на высоких частотах. Эти результаты повторены здесь для удобства сравнения. На рис. 5.28(*в*) представлен результат, полученный при помощи винеровской фильтрации на основе (5.8-3) с искажающей функцией из примера 5.11. Значение K было подобрано так, чтобы обеспечить наилучшее качество восстановления. В этом примере очевидно преимущество винеровской фильтрации по сравнению с инверсной фильтрацией. Сравнивая рис. 5.25(*а*) и 5.28(*в*), мы видим, что винеровская фильтрация позволяет получить результат, очень близкий по виду к исходному изображению. ■

Пример 5.13. Дальнейшие сравнения винеровской фильтрации.

■ В верхнем ряду на рис. 5.29 представлены слева направо следующие изображения: (1) смазанное изображение рис. 5.26(*б*), дополнительно сильно искаженное аддитивным гауссовым шумом с нулевым средним и дисперсией 650; (2) результат его восстановления с помощью инверсной фильтрации; (3) результат восстановления с помощью винеровской фильтрации. Мы использовали винеровский фильтр из (5.8-3) с искажающей функцией примера 5.10 и со значением K, подобранным для получения возможно лучшего результата. Отметим, что шум на изображении, которое получено методом инверсной фильтрации, очень велик и имеет ярко выраженную диагональную структуру в направлении смазывания. Результат фильтрации по Винеру никоим образом нельзя признать идеальным, но он дает некоторое представление о содержании изображения. Текст на изображении читается, хотя и не без труда.

В среднем ряду на рис. 5.29 представлена та же последовательность изображений, но отвечающая шуму с дисперсией, уменьшенной на один порядок. Это уменьшение не дало заметного эффекта в случае инверсной фильтрации, но результат винеровской фильтрации заметно улучшился. Читать текст теперь значительно легче. Для изображений нижнего ряда на рис. 5.29 дисперсия шума была уменьшена по величине более чем на пять порядков по сравнению с изображениями верхнего ряда. На самом деле изображение на рис. 5.29(*ж*) уже не содержит заметного на глаз шума. В этом случае результат инверсной фильтрации представляет интерес. Шум все еще хорошо заметен, но текст можно видеть через шумовую «завесу». Это хорошая иллюстрация к тому, что было сказано при обсуждении формулы (5.7-2). Другими словами, из рассмотрения изображения на рис. 5.29(*з*) ясно, что инверсный фильтр вполне в состоянии существенно уменьшить степень размывания. Шум, однако, по-прежнему превалирует на изображении. Если бы мы могли «заглянуть за шумовую завесу» на рис. 5.29(*б*) и (*д*), то мы также обнаружили бы незначительную степень размывания. Результат винеровской фильтрации на рис. 5.29(*и*) превосходен и весьма близок к исходному изображению на рис. 5.26(*а*). Результаты подобного рода показательны в плане того, чего можно достичь с помощью винеровской фильтрации в том случае, когда возможно построение хорошей оценки для искажающей функции. ■

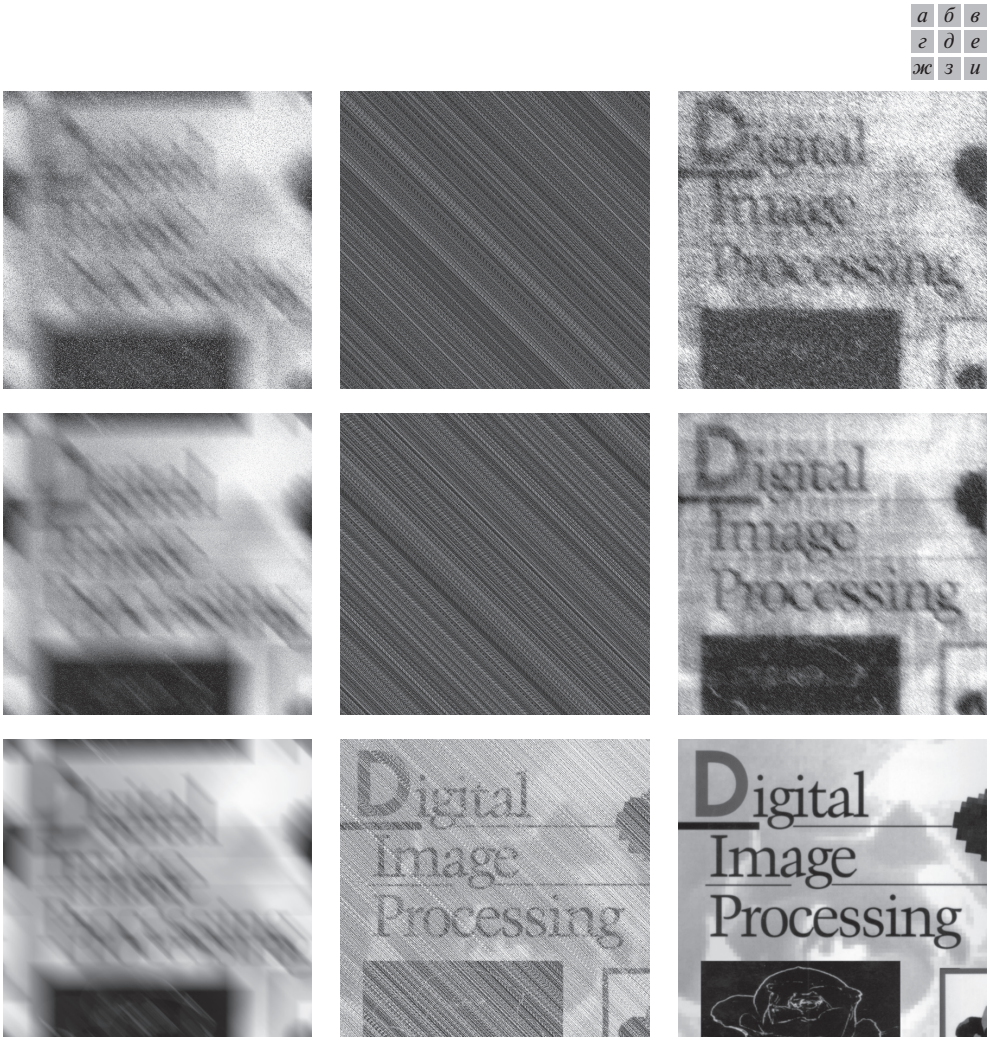


Рис. 5.29. (*а*) 8-битовое изображение, смазанное в результате движения и дополнительно искаженное аддитивным шумом. (*б*) Результат инверсной фильтрации. (*в*) Результат винеровской фильтрации. (*г*)—(*е*) Те же изображения, но дисперсия шума на порядок меньше по величине. (*ж*)—(*и*) Те же изображения, но дисперсия шума уменьшена на пять порядков по величине по отношению к (*а*). Обратите внимание на рисунок (*з*), где несмазанное изображение проступает через «завесу» помехи

5.9. Фильтрация методом минимизации сглаживающего функционала со связью

Проблема, заключающаяся в необходимости иметь некоторую информацию относительно искажающей функции, является общей для всех рассматриваемых в этой главе методов восстановления. Применение винеровской фильтрации связано с дополнительной трудностью, состоящей в том, что энергетические

спектры неискаженного изображения и шума также должны быть известны. В предыдущем разделе было показано, что использование приближения (5.8-6) позволяет получать отличные результаты. Однако использование константы в качестве оценки для отношения энергетических спектров не всегда приводит к удовлетворительному решению задачи.

Применение метода, рассматриваемого в этом разделе, требует только знания среднего значения и дисперсии шума. Это является важным преимуществом метода, поскольку, как показано в разделе 5.2.4, обычно можно оценить указанные величины на основе заданного искаженного изображения. Другое отличие состоит в том, что винеровская фильтрация основана на минимизации в смысле некоторого статистического критерия и, следовательно, является оптимальной в некотором среднестатистическом смысле. Метод, рассматриваемый в настоящем разделе, обладает тем замечательным свойством, что позволяет получить оптимальный результат для *каждого* конкретного изображения, к которому он применяется. Конечно, важно понимать, что тот критерий, по отношению к которому результат является оптимальным с теоретической точки зрения, не связан с механизмом зрительного восприятия. Поэтому выбор в пользу того или иного метода почти всегда будет определяться (по крайней мере частично) на основе визуальной оценки получаемых результатов.

Используя определение свертки (4.6-23), как уже объяснялось в разделе 2.6.6, можно переписать уравнение (5.5-16) в матрично-векторном виде следующим образом:

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \boldsymbol{\eta}\,. \tag{5.9-1}$$

Для ознакомления с операциями над векторами и матрицами обратитесь к интернет-сайту книги.

Пусть, например, изображение $g(x,y)$ имеет размеры $M \times N$. Тогда мы можем сформировать вектор $\mathbf{g}$ таким образом, чтобы первые N его элементов были равны значениям в первой строке изображения $g(x,y)$, следующие N элементов были равны значениям во второй строке и т. д. Полученный вектор будет иметь размер $MN \times 1$. Аналогично формируются векторы $\mathbf{f}$ и $\boldsymbol{\eta}$, которые в результате имеют те же размеры. Далее матрица $\mathbf{H}$ имеет размеры $MN \times MN$. Ее элементы задаются значениями h в свертке (4.6-23).

Естественно предположить, что задача восстановления может быть таким способом сведена к задаче линейной алгебры. К сожалению, дело обстоит не так просто. Предположим для примера, что мы работаем с изображениями средних размеров; пусть для определенности $M = N = 512$. Тогда векторы в формуле (5.9-1) будут иметь размеры 262144, а матрица $\mathbf{H}$ будет иметь размеры 262144×262144. Работа с векторами и матрицами подобных размеров представляет собой далеко не простую задачу. Дело дополнительно осложняется тем, что матрица $\mathbf{H}$ очень чувствительна к шумам[16] (что, учитывая опыт, приобретенный нами в преды-

[16] Матрица $\mathbf{H}$ является плохо определенной (т. е. ее определитель близок или даже равен нулю), вследствие чего соответствующая линейная задача является в высшей степени некорректной. Ее решение в обычном смысле может вообще не существовать, и даже если оно существует, то является очень неустойчивым по отношению к шуму η. Решение подобных задач на практике является делом исключительной трудности. — *Прим. перев.*

дущих двух параграфах при рассмотрении влияния шума на результаты восстановления, не должно показаться удивительным). Несмотря на это, формулировка задачи восстановления в матричном виде облегчает построение методов восстановления.

Корни метода восстановления, составляющего предмет настоящего раздела, лежат в области матричного анализа. Мы не будем полностью обосновывать этот метод, но приведем в конце главы ссылки на работы, в которых содержится детальный его вывод. Главной проблемой является чувствительность задачи по отношению к шуму. Один из способов преодоления этой трудности состоит в регуляризации задачи, которая достигается заменой исходной задачи на задачу нахождения экстремума (минимума) некоторого сглаживающего функционала[17] (*фильтрация по Тихонову* или *тихоновская регуляризация*). В качестве такого функционала $C[f]$ можно использовать квадрат нормы уже знакомого нам лапласиана

> В разделе обучающих материалов на сайте книги в Интернете имеется целая глава, посвященная методам восстановления изображений на основе матричной алгебры.

$$C[f]=\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[\nabla^2 f(x,y)\right]^2 \tag{5.9-2}$$

с дополнительным ограничением (связью) вида

$$\left\|\mathbf{g}-\mathbf{H}\hat{\mathbf{f}}\right\|^2=\|\boldsymbol{\eta}\|^2, \tag{5.9-3}$$

где $\|\mathbf{w}\|^2 \triangleq \mathbf{w}^T\mathbf{w}$ — евклидова норма вектора, а $\hat{\mathbf{f}}$ — искомая оценка неискаженного изображения. Оператор Лапласа ∇^2 определен выражением (3.6-3). Напомним, что если $\mathbf{w}$ — n-компонентный вектор, то

$$\mathbf{w}^T\mathbf{w}=\sum_{k=1}^{n} w_k^2,$$

где w_k — k-я координата вектора.

Решение оптимизационной задачи (5.9-2) с условием (5.9-3) в частотной области дается выражением

$$\hat{F}(u,v)=\left(\frac{H^*(u,v)}{|H(u,v)|^2+\gamma|P(u,v)|^2}\right)G(u,v), \tag{5.9-4}$$

где параметр γ (параметр регуляризации) должен быть выбран таким образом, чтобы выполнялось условие (5.9-3), а функция $P(u,v)$ есть Фурье-преобразование функции

---

[17] Эта идея лежит в основе общего подхода к решению некорректных задач, который был предложен и развит школой А. Н. Тихонова. Такой подход называется *тихоновским методом регуляризации*, и рассматриваемый ниже метод является его весьма частным случаем. В связи со сказанным мы далее будем называть этот метод фильтрации, основанный на минимизации сглаживающего функционала со связью, *фильтрацией по Тихонову* или *тихоновской регуляризацией*. — *Прим. перев.*

$$p(x,y)=\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}, \tag{5.9-5}$$

т. е. той функции, с помощью которой в разделе 3.6.2 был определен оператор Лапласа[18]. Как уже отмечалось выше, важно помнить о том, что функция $p(x,y)$, равно как и все остальные функции в пространственной области, должны быть правильно дополнены нулями перед вычислением их Фурье-преобразования для использования в (5.9-4). Вопрос о дополнении нулями подробно рассматривался в разделе 4.6.6. Обратим внимание, что при обращении параметра регуляризации γ в нуль выражение (5.9-4) сводится к инверсной фильтрации.

Пример 5.14. Сравнение винеровской фильтрации и фильтрации по Тихонову.

■ На рис. 5.30 представлены результаты восстановления изображений на рис. 5.29(*а*), (*г*) и (*ж*) с помощью фильтрации по Тихонову. Значения параметра γ подбирались вручную так, чтобы обеспечить наилучшее визуальное качество восстановления. Тот же метод выбора параметров был использован для получения результатов фильтрации по Винеру, которые были представлены на рис. 5.29(*в*), (*е*), (*и*). Сравнивая между собой фильтрации по Тихонову и по Винеру, можно отметить, что первая дала несколько лучшие результаты в случаях большого и среднего по величине шума, при том что обе дали практически одинаковые результаты в случае низкого шума. То, что фильтрация по Тихонову оказалась лучше фильтрации по Винеру в случае подбора значений параметров вручную, не является неожиданностью. Дело в том, что параметр регуляризации γ в (5.9-4) является числом, в то время как параметр K в (5.8-6) является по сути значением постоянной функции, используемой для приближения отношения двух неизвестных функций в частотной области, которое редко является постоянным. Поэтому естественно ожидать, что результат, основанный на подборе параметра γ «вручную», даст более точную оценку неискаженного изображения. ■

а *б* *в*

Рис. 5.30. Результаты фильтрации по Тихонову. Сравните (*а*), (*б*) и (*в*) с результатами винеровской фильтрации на рис. 5.29(*в*), (*е*) и (*и*) соответственно

[18] Можно, в принципе, использовать и функцию $H(u,v) = -(u^2 + v^2)$, которая в соответствии с разделом 4.9.4 определяет «частотный» оператор Лапласа. — *Прим. перев.*

Как показывает приведенный пример, значения параметра регуляризации γ можно перебирать в интерактивном режиме до тех пор, пока приемлемый результат не будет получен. Однако если мы хотим получить оптимальное решение, то значение параметра должно быть выбрано так, чтобы выполнялось условие связи (5.9-3). Ниже приведена итеративная процедура такого выбора.

Определим вектор невязки **r** следующим образом:

$$\mathbf{r} = \mathbf{g} - \mathbf{H}\hat{\mathbf{f}}. \tag{5.9-6}$$

Поскольку решение $\hat{F}(u,v)$ (и соответствующий вектор $\hat{\mathbf{f}}$), определяемое по (5.9-4), есть функция параметра γ, вектор невязки **r** также зависит от этого параметра. Можно показать (см. [Hunt, 1973]), что функционал невязки

$$\phi(\gamma) = \mathbf{r}^T\mathbf{r} = \|\mathbf{r}\|^2 \tag{5.9-7}$$

является монотонной функцией параметра γ. Мы хотели бы так выбрать параметр γ, чтобы выполнялось условие

$$\|\boldsymbol{\eta}\|^2 - a \le \|\mathbf{r}\|^2 \le \|\boldsymbol{\eta}\|^2 + a, \tag{5.9-8}$$

где коэффициент a задает приемлемую точность выполнения условия связи. В случае $a = 0$ имеет место $\|\mathbf{r}\|^2 = \|\boldsymbol{\eta}\|^2$, и ввиду (5.9-6) условие (5.9-3) выполняется точно.

Поскольку функционал невязки $\phi(\gamma)$ является монотонной функцией параметра регуляризации, нахождение искомого значения γ не представляет трудности. Один из алгоритмов состоит в следующем.

1. Задать начальное значение γ.
2. Вычислить $\|\mathbf{r}\|^2$.
3. Если условие (5.9-8) выполняется, то цель достигнута. В противном случае увеличить значение γ, если $\|\mathbf{r}\|^2 < \|\boldsymbol{\eta}\|^2 - a$, или уменьшить его, если $\|\mathbf{r}\|^2 > \|\boldsymbol{\eta}\|^2 + a$, и повторить шаг 2, используя новую оценку $\hat{F}(u,v)$, пересчитанную по (5.9-4) с новым значением γ.

Для увеличения скорости сходимости можно использовать более продвинутые методы, например метод касательных Ньютона.

Для того чтобы использовать данный алгоритм, необходимы величины $\|\mathbf{r}\|^2$ и $\|\boldsymbol{\eta}\|^2$. Для вычисления $\|\mathbf{r}\|^2$ заметим, что из (5.9-6) следует равенство

$$R(u,v) = G(u,v) - H(u,v)F(u,v), \tag{5.9-9}$$

и функция $r(x,y)$ может быть получена вычислением обратного Фурье-преобразования от функции $R(u,v)$. Далее

$$\|\mathbf{r}\|^2 = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} r^2(x,y). \tag{5.9-10}$$

Вычисление $\|\boldsymbol{\eta}\|^2$ приводит к интересному результату. Прежде всего рассмотрим дисперсию шума полного изображения, которую мы оцениваем методом выборочного среднего, рассмотренным в разделе 3.3.4:

$$\sigma_\eta^2 = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left(\eta(x,y) - m_\eta\right)^2, \tag{5.9-11}$$

где

$$m_\eta = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\eta(x,y) \tag{5.9-12}$$

есть выборочное среднее значение шума. Сравнивая два последних выражения с выражением для $\|\mathbf{\eta}\|^2$, которое имеет тот же вид, что и (5.9-10), мы видим, что

$$\|\mathbf{\eta}\|^2 = MN\left(\sigma_\eta^2 + m_\eta^2\right). \tag{5.9-13}$$

Это очень важный результат, который показывает, что для реализации оптимального алгоритма восстановления нам достаточно знать лишь среднее значение и дисперсию шума. Эти величины нетрудно оценить (см. раздел 5.2.4) в предположении, что значения шума и изображения не коррелированы. Это предположение является основным для всех рассматриваемых в настоящей главе методов.

Пример 5.15. Итеративная оптимальная фильтрация по Тихонову.

■ На рис. 5.31(*а*) представлен результат восстановления изображения на рис. 5.25(*б*), полученный с помощью оптимального алгоритма фильтрации на основе только что рассмотренной итеративной процедуры. Начальное значение γ было равно 10^{-5}, величина изменения значений γ в итеративной процедуре была равна 10^{-6}, а значение коэффициента a равнялось 0,25. Параметры шума в алгоритме восстановления выбирались такими же, как и при генерации изображения на рис. 5.25(*б*) (шум имел нулевое среднее и дисперсию 10^{-5}). Результат восстановления столь же хорош, что и результат на рис. 5.28(*в*), для получения которого была использована винеровская фильтрация, а параметр K подбирался вручную так, чтобы обеспечить наилучший в визуальном смысле результат. Изображение на рис. 5.31(*б*) демонстрирует, что может случиться при использовании неправильных оценок для параметров шума. В данном случае считалось,

а *б*

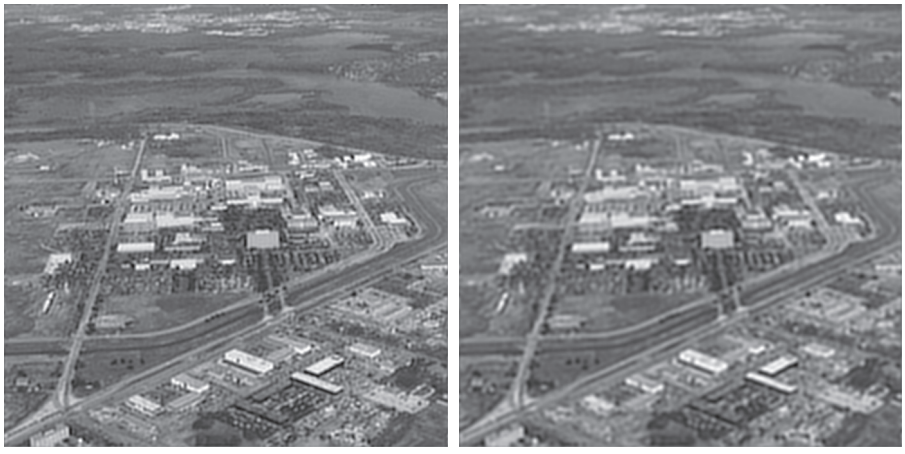


Рис. 5.31. (*а*) Результат восстановления изображения на рис. 5.25(*б*) с помощью итеративной оптимальной фильтрации по Тихонову, использующей правильные значения параметров шума. (*б*) Аналогичный результат, полученный при неправильных значениях параметров шума

что дисперсия шума равна 10^{-2}, а среднее значение по-прежнему равно нулю. Результат восстановления расфокусирован заметно сильнее. ■

Как было сказано в начале этого раздела, важно понимать, что оптимальное в смысле минимизации сглаживающего функционала со связью восстановление не обязательно является «лучшим» в смысле визуального качества. В зависимости от природы и величины искажений и шума разные факторы влияют на вид окончательного результата, получаемого с помощью итеративного алгоритма построения оптимальной оценки. Вообще фильтры для восстановления, которые строятся в автоматическом режиме, дают результаты хуже, чем фильтры, параметры которых настраиваются вручную под конкретное изображение. Это относится, в частности, к рассмотренному фильтру, который полностью задается одним числовым параметром.

5.10. Среднегеометрический фильтр

Рассмотренный в разделе 5.8 винеровский фильтр можно несколько обобщить. Это обобщение приводит к так называемому *среднегеометрическому фильтру*:

$$\hat{F}(u,v)=\left(\frac{H^*(u,v)}{|H(u,v)|^2}\right)^{\alpha}\left(\frac{H^*(u,v)}{|H(u,v)|^2+\beta\left(\frac{S_\eta(u,v)}{S_f(u,v)}\right)}\right)^{1-\alpha}G(u,v), \qquad (5.10\text{-}1)$$

где α и β — положительные вещественные константы. Среднегеометрический фильтр представляет собой произведение двух выражений в скобках, которые соответственно возводятся в степени α и $1-\alpha$.

При $\alpha = 1$ этот фильтр сводится к инверсному фильтру. При $\alpha = 0$ фильтр превращается в так называемый *параметрический винеровский фильтр*, который сводится к обычному винеровскому фильтру при $\beta = 1$. При $\alpha = 1/2$ фильтр представляет собой произведение двух величин, возведенных в одинаковую степень 1/2, т. е. является средним геометрическим этих величин, откуда и происходит название фильтра. При $\alpha = 1/2$ и $\beta = 1$ этот фильтр известен как *фильтр эквализации спектра*. При $\beta = 1$ по мере увеличения α больше 1/2 работа фильтра все больше напоминает работу инверсного фильтра. Аналогично при убывании α меньше 1/2 фильтр ведет себя как винеровский фильтр. Выражение (5.10-1) весьма полезно, поскольку с его помощью в рамках одного выражения в действительности удается описать целое семейство фильтров.

5.11. Реконструкция изображения по проекциям

В предыдущих разделах настоящей главы изучались методы восстановления изображения по его искаженному варианту. В данном разделе будет рассмотрена проблема реконструкции изображения по множеству его проекций; основ-

ное внимание будет уделено рентгеновской компьютерной томографии (КТ). Это самый первый и до сих пор наиболее широко используемый тип КТ, являющийся одним из основных приложений цифровой обработки изображений в медицине.

> Как отмечалось в главе 1, наравне с термином «компьютерная томография» (КТ) используется термин «компьютерная аксиальная томография» (КАТ).

5.11.1. Введение

Задача реконструкции в принципе проста и может быть качественно объяснена непосредственным, интуитивно понятным способом. Для начала рассмотрим рис. 5.32(*а*), состоящий из единственного объекта на однородном фоне. Чтобы внести в последующее объяснение физический смысл, предположим, что это изображение представляет собой сечение некой трехмерной области тела человека. Также допустим, что фон на изображении соответствует мягким тканям, а круглый объект — опухоли, тоже однородной, но с более высоким коэффициентом поглощения.

Далее предположим, что мы посылаем тонкий плоский пучок рентгеновских лучей слева направо (вдоль плоскости изображения), как показано на рис. 5.32(*а*), и будем считать, что объект поглощает больше энергии лучей, чем фон, как это обычно и бывает. Поставив с противоположной стороны области линейку детекторов рентгеновских лучей, получим показанный сигнал (профиль поглощения), амплитуда которого (яркость) пропорциональна поглощению[19]. Можно рассматривать каждую точку полученного сигнала как сумму значений коэффициента поглощения вдоль того луча из пучка, который пространственно соответствует именно этой точке (такую сумму часто называют *суммой* (*интегралом*) *вдоль луча*[20]). Положение дел таково, что полученный одномерный сигнал поглощения — это вся информация, которую мы в данный момент имеем об объекте.

По одной проекции у нас нет способа определить, имеем ли мы дело только с одним объектом или со многими, расположенными вдоль направления лучей; но мы начнем процесс реконструкции с построения *изображения*, базирующегося именно на такой информации. Подход состоит в проецировании одномерного сигнала назад вдоль направления распространения пучка лучей, как показано на рис. 5.32(*б*). Процесс обратного проецирования одномерного сигнала по двумерной области иногда называют обратным *размыванием*[21] проекции по области. В терминах цифровой обработки изображений это означает копирование одного и того же одномерного сигнала во все сечения изображения, перпендикулярные направлению распространения пучка. Так, изображение рис. 5.32(*б*) было построено копированием одномерного сигнала во все

---

[19] Трактовка физики источников и детекторов рентгеновских лучей остается за границами нашего рассмотрения, которое сконцентрировано на аспектах обработки изображений в КТ. В качестве отличного введения в физику формирования рентгеновских изображений см. [Prince, Links, 2006].

[20] В оригинале *raysum*. — *Прим. перев.*

[21] В оригинале *smearing*. — *Прим. перев.*

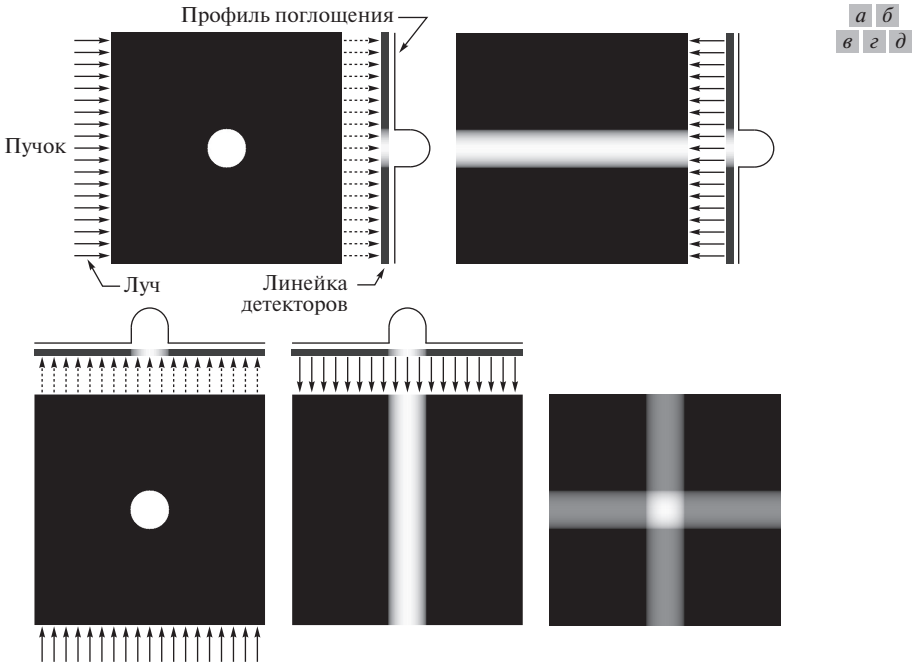


Рис. 5.32. (*а*) Плоская область с простым объектом, входной параллельный пучок лучей и линейка детекторов. (*б*) Результат обратного проецирования полученных линейкой детекторов данных (т. е. одномерный профиль поглощения). (*в*) Лучи и детекторы, повернутые на 90°. (*г*) Обратная проекция. (*д*) Область, где перекрываются обратные проекции, имеет яркость вдвое выше яркости одной обратной проекции

столбцы изображения. По очевидным причинам такой процесс называется *обратным проецированием*.

Далее, предположим, мы повернули пару излучатель-детектор на 90°, как показано на рис. 5.32(*в*). Повторяя описанный выше процесс, получим изображение обратной проекции по вертикальному направлению (рис. 5.32(*г*)). Для продолжения реконструкции *прибавим* данный результат к предыдущей обратной проекции, как это показано на рис. 5.32(*д*). Теперь мы можем сказать, что интересующий объект расположен на пересечении светлых полос в квадрате, амплитуда которого вдвое превышает амплитуду каждой из обратных проекций. Небольшое размышление приводит к выводу, что добавив аналогичным способом дополнительное число проекций, мы узнаем больше о форме объекта. Фактически именно это и показано на рис. 5.33. С увеличением числа проекций мощность участков обратных проекций, где отсутствует пересечение, уменьшается по сравнению с мощностью области пересечения. Общий эффект состоит в том, что светлые области будут доминировать на результате, а обратные проекции без пересечений или с малым их числом упадут в яркости до фона, как только изображение будет масштабировано по яркости для визуализации.

Изображение рис. 5.33(*е*), сформированное по 32 проекциям, иллюстрирует эту концепцию. Однако заметим, что хотя полученная реконструкция является сравнительно неплохим приближением формы исходного объекта, изображе-

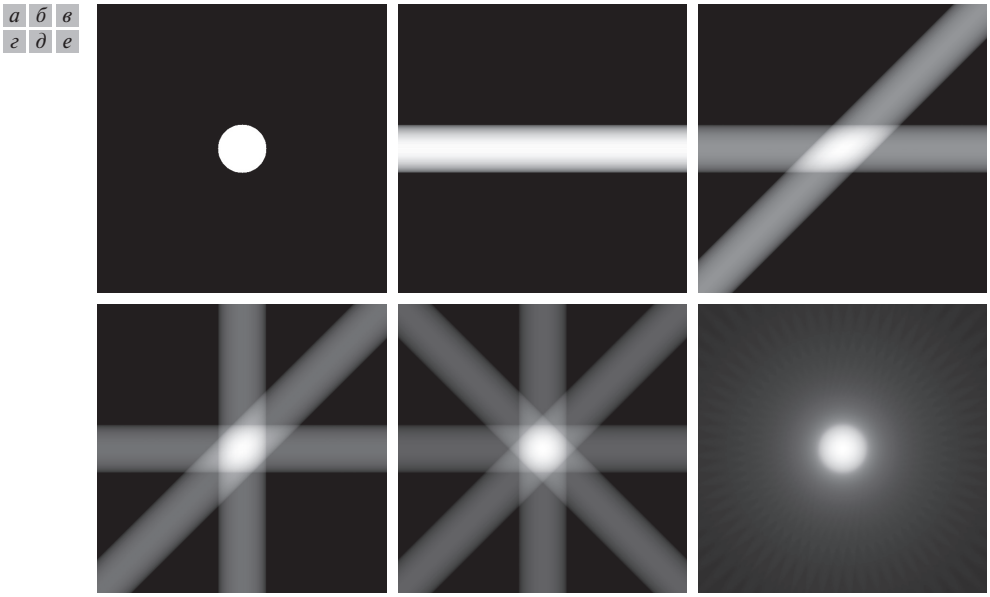


Рис. 5.33. (*а*) То же изображение, что и рис. 5.32(*а*). (*б*)—(*д*) Реконструкция с использованием 1, 2, 3 и 4 обратных проекций, поворачиваемых на 45°. (*е*) Реконструкция по 32 обратным проекциям с углами поворота 5,625°. (Обратите внимание на размывание)

ние размыто эффектом «ореола», возникновение которого можно проследить на последовательных стадиях рис. 5.33. Так, ореол на рис. 5.33(*д*) выглядит как «звезда», яркость которой ниже яркости объекта, однако выше, чем у фона. Увеличение числа проекций делает форму ореола круглой, как на рис. 5.33(*е*). Размывание при КТ-реконструкции является важным вопросом, решение которого будет рассматриваться в разделе 5.11.5. Заканчивая обсуждение рис. 5.32 и 5.33, отметим, что проекции, в которых направления распространения пучков составляют угол 180°, являются зеркально симметричными, поэтому для получения всех необходимых для реконструкции проекций достаточно сформировать лишь проекции в диапазоне [0°, 180°].

Пример 5.16. Обратные проекции простой плоской области, содержащей два объекта.

■ На рис. 5.34 иллюстрируется реконструкция на основе обратных проекций несколько более сложной области, содержащей два объекта с различными свойствами поглощения. Рис. 5.34(*б*) представляет результат одной из обратных проекций. На нем можно отметить три различающихся области: тонкую горизонтальную серую полосу, соответствующую участку проекции, занятому одним лишь малым объектом, над ней яркую полосу (большее поглощение), соответствующую участку, который загораживается одновременно двумя объектами, и широкую верхнюю полосу — проекцию оставшейся части эллиптического объекта. Рис. 5.34(*в*) и (*г*) демонстрируют реконструкцию с помощью двух проекций (под углами 90°) и четырех проекций (под 45°) соответственно. Интерпретация этих изображений аналогична обсуждению рис. 5.33(*в*)—(*д*). Использова-

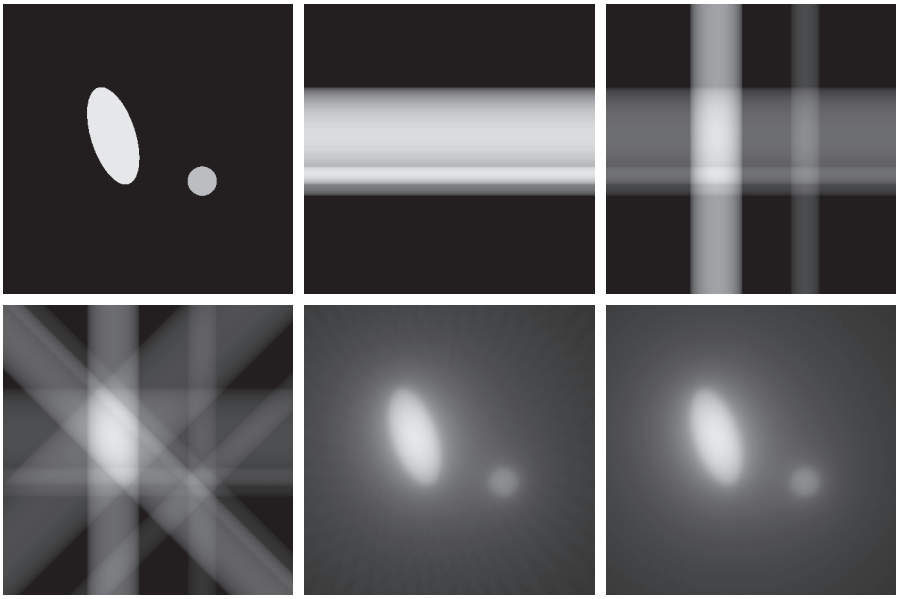


Рис. 5.34. (*а*) Область с двумя объектами. (*б*)—(*г*) Реконструкция с использованием 1, 2 и 4 обратных проекций, с шагом по углу 45°. (*д*) Реконструкция по 32 обратным проекциям с угловым шагом 5,625°. (*е*) Реконструкция по 64 обратным проекциям с угловым шагом 2,8125°

ние 32 и 64 обратных проекций на рис. 5.34(*д*) и (*е*) демонстрирует более точное восстановление. Оба этих результата визуально весьма близки, и оба демонстрируют одну и ту же проблему, связанную с размыванием, которая уже отмечалась раньше; решение ее будет рассматриваться в разделе 5.11.5. ■

5.11.2. Принципы компьютерной томографии (КТ)

Цель рентгеновской компьютерной томографии — получить трехмерное представление внутренней структуры объекта при помощи просвечивания его рентгеновскими лучами под многими различными направлениями. Представим обычную рентгенографию грудной клетки, получаемую путем помещения пациента перед чувствительной к рентгеновским лучам пластиной и «просвечивания» его пучком рентгеновских лучей конической формы. Рентгеновская пластина создает изображение, яркость которого в каждой точке пропорциональна[22] энергии рентгеновских лучей, падающих на эту точку после прохождения сквозь объект. Данное изображение является двумерным эквивалентом проекций, рассмотренных в предыдущем разделе. Можно выполнить обратное проецирование всего полученного двумерного изображения и сформировать трехмерное объемное изображение-проекцию. Повторяя этот процесс по многим углам и складывая обратные проекции, в результате получим трехмерное

[22] В качестве рентгеновской пластины (в оригинале — *X-ray sensitive plate*), удовлетворяющей таким свойствам, может выступать, например, фосфоресцирующий экран. — *Прим. перев.*

изображение структуры полости грудной клетки. Задача КТ — получить ту же информацию (или какие-то ограниченные ее части) путем формирования *двумерных сечений* по всему объему тела. Трехмерное представление может быть затем получено наложением слоев друг на друга. КТ-реализация много экономичнее, поскольку для получения одного слоя с высоким пространственным разрешением требуется значительно меньшее число детекторов, чем для получения полной двумерной проекции того же разрешения. Объемы вычислений и дозы рентгеновских лучей также значительно снижаются, делая КТ, основанную на использовании одномерных проекций, более практичным подходом.

Как и с преобразованием Фурье, которое обсуждалось в предыдущей главе, основные математические концепции, требуемые для КТ, были известны за годы до того, как цифровые вычислительные машины сделали их практически реализуемыми. Теоретическое обоснование КТ восходит к Иоганну Радону (Johann Radon), математику из Вены, который предложил в 1917 г. метод проецирования двумерного объекта вдоль параллельных лучей в качестве части своей работы по линейным интегралам. Метод теперь известен как *преобразование Радона*, которое будет рассмотрено в следующем разделе. Через сорок пять лет Аллан Кормак (Allan M. Cormack), физик из Университета Тафтса (Tufts University), частично «переоткрыл» эти концепции и использовал их для КТ. Кормак опубликовал свои исходные результаты в 1963 и 1964 гг. и показал, как они могут быть использованы для реконструкции изображений поперечных сечений тела по рентгеновским снимкам, сделанным под различными углами. Он дал математические формулы, необходимые для реконструкции, и построил прототип КТ, чтобы показать практичность своих идей. Работая независимо, инженер-электрик Гофри Хаунсфилд (Godfrey N. Hounsfield) со своими коллегами из EMI в Лондоне сформулировал похожее решение и построил первый медицинский аппарат КТ. Кормак и Хаунсфилд в 1979 г. разделили Нобелевскую премию по медицине за свой вклад в медицинскую томографию.

Компьютерные томографы (*КТ-сканеры*) *первого поколения* (*G1*) используют узкий пучок рентгеновских лучей и единственный детектор, как показано на рис. 5.35(*а*). Для каждого угла поворота пара излучатель/детектор сдвигаются пошагово вдоль показанного линейного направления. Проекция (как те, что показаны на рис. 5.32) формируется измерением значения на выходе детектора при каждом пространственном шаге. По окончании линейного пространственного перемещения система излучатель—детектор поворачивается, и процедура повторяется для получения новой проекции под другим углом. Процедура повторяется для всех требуемых углов в диапазоне [0°, 180°], формируя тем самым полный набор проекций, из которого с помощью операции обратного проецирования формируется одно изображение, как объяснено в предыдущем разделе. Крестообразная метка на голове изображенного пациента (объекта) на рис. 5.35 означает наличие движения в направлении, перпендикулярном по отношению к плоскости получения изображения системой излучатель—детектор. Набор изображений поперечных сечений (слоев) обеспечивается пошаговым смещением объекта (после полного сканирования каждого сечения) относительно плоскости излучатель—детектор. Накладывая эти сечения вычислительным образом друг на друга, получим трехмерный объем участка тела. Для медицинских целей томографы типа G1 больше не производятся, но поскольку они дают

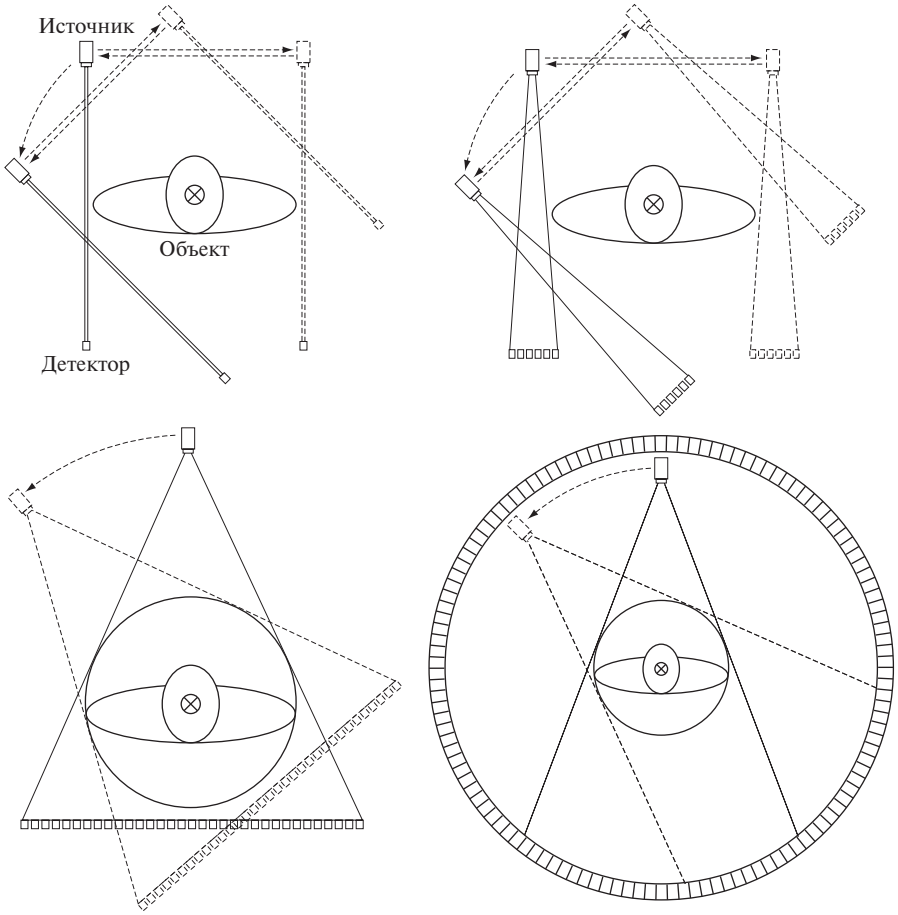


Рис. 5.35. Четыре поколения компьютерных томографов. Пунктирные линии со стрелками означают пошаговое линейное движение. Пунктирные кривые со стрелками означают пошаговое круговое движение. Крестовые метки на голове объектов означают линейное движение в направлении, перпендикулярном плоскости данного рисунка. Двунаправленные пунктирные линии со стрелками на (*а*) и (*б*) означают, что блок «излучатель—детектор» сдвигается, а затем возвращается в первоначальное положение

изображение пучка параллельных лучей (как видно на рис. 5.32), геометрия их съемки преимущественно используется для ознакомления с основами получения изображения в КТ. Как обсуждается в следующем разделе, эта геометрия служит отправной точкой для вывода уравнений, необходимых для осуществления реконструкции изображения по проекциям.

Компьютерные томографы второго поколения (*G2*) (рис. 5.35(*б*)) работают по тому же принципу, что и томографы G1, но их пучок имеет веерный вид. Это позволяет использовать сразу несколько детекторов, что позволяет уменьшить число смещений пары излучатель/детектор. *Компьютерные томографы третьего поколения* (*G3*) имеют значительно более совершенную геометрию по сравнению с томографами

предыдущих поколений. Как показано на рис. 5.35(*в*), G3 томографы используют достаточно длинную линейку детекторов (порядка 1000 отдельных детекторов), позволяющую перекрыть все поле зрения одним широким пучком. Поэтому съемка под каждым из углов дает сразу всю проекцию, устраняя необходимость сдвигать пару излучатель—детектор, как того требует геометрия томографов типов G1 и G2. *Компьютерные томографы четвертого поколения* (*G4*) стали еще одним шагом вперед. Расположив детекторы (порядка 5000 отдельных детекторов) кольцеобразно, для получения всего набора проекций достаточно вращать лишь излучатель. Основным преимуществом томографов G3 и G4 является скорость съемки, а основными недостатками — цена и повышенное рассеяние рентгеновских лучей, из-за чего для достижения характеристик сигнал—шум, аналогичных томографам G1 и G2 типов, требуются повышенные дозы излучения.

Появляются новые принципы построения томографов. Например, *компьютерные томографы пятого поколения* (*G5*), также называемые *томографами с электронным лучом* (*electron beam computed tomography* (*EBCT*) *scanners*), устраняют движения всех механических элементов за счет использования электронных пучков, управляемых электромагнитным полем. Соударение электронных пучков с окружающими пациента вольфрамовыми анодами приводит к возникновению рентгеновских лучей, из которых затем формируется проходящий сквозь пациента веерный пучок, регистрируемый кольцом детекторов, как и в томографах четвертого поколения.

Обычно, при получении КТ-изображения пациент должен находиться в неподвижном положении в течение времени, необходимого для получения изображения одного сечения. Затем сканер останавливается, и пациент перемещается в направлении, перпендикулярном плоскости сканирования при помощи механизированного стола. Затем формируется следующее изображение, и процедура повторяется для всего набора шагов, требуемых для перекрытия заданного участка тела пациента. Хотя одно изображение может быть получено менее чем за секунду, имеются процедуры (например сканирование брюшной полости или грудной клетки), которые требуют от пациента на протяжении всего времени сканирования задерживать дыхание. Для завершения операции получения, скажем, 30 изображений, может потребоваться несколько минут. Растет применение подхода, называемого *спиральная КТ* (*helical CT*); томографы данного типа иногда называют *компьютерными томографами шестого поколения* (*G6*). При таком подходе томограф типа G3 и G4 снабжается контактными кольцами, которые избавляют от необходимости использовать электрические и сигнальные кабели между излучателем, приемниками и блоком обработки. При этом пара излучатель/детектор вращается постоянно без остановки, а пациент сдвигается с постоянной скоростью вдоль оси, перпендикулярной плоскости сканирования. Результатом является «спиральный» массив данных, который затем обрабатывается для получения изображений отдельных сечений.

Появляются *компьютерные томографы седьмого поколения* (*G7*), также называемые *многослойными компьютерными томографами* (*multislice CT scanners*), в которых используются «толстые» (конусные) веерные пучки в сочетании с параллельными блоками детекторов для одновременного сбора объемных данных КТ. Т. е. при каждой экспозиции рентгеновского излучения формируется не одно изображение, а целый блок изображений соседних поперечных сечений. Кроме

значительного повышения детальности, данный подход имеет преимущество в том, что более экономно использует рентгеновские трубки, тем самым снижая затраты и потенциально уменьшая необходимую дозу облучения.

Начиная со следующего раздела будет развит математический аппарат, необходимый для формулировки алгоритмов проецирования и реконструкции изображений. Основное внимание будет сконцентрировано на принципе обработки изображений, который лежит в основе всех только что рассмотренных подходов КТ. Информация, касающаяся механических характеристик и параметров излучателей/детекторов КТ-систем, излагается в литературе, цитируемой в конце главы.

5.11.3. Проекции и преобразование Радона

В нижеследующем материале детально рассматривается математика, необходимая для реконструкции изображений в контексте рентгеновской компьютерной томографии, но те же базовые принципы применимы и к другим методам КТ, таким как ОФЭКТ — однофотонная эмиссионная компьютерная томография (single photon emission tomography, SPECT), позитронно-эмиссионная томография (ПЭТ), магнитно-резонансная томография (МРТ), а также к некоторым методам формирования ультразвуковых изображений.

> На протяжении данного раздела мы следуем соглашениям, принятым в КТ, и помещаем начало координат плоскости *xy* в центр вместо нашего обычного варианта в левом верхнем углу (см. раздел 2.4.2). Однако заметим, что обе координатные системы являются правосторонними, и единственная разница в том, что в нашей системе отсутствуют отрицательные полуоси. Можно вычислить разницу, возникающую при таком простом сдвиге начала координат, так что оба представления являются взаимозаменяемыми.

Прямая линия в декартовых координатах может быть представлена либо в форме *уравнения с угловым коэффициентом* $y = ax + b$, либо в форме *представления нормалью*, как показано на рис. 5.36:

$$x\cos\theta + y\sin\theta = \rho. \tag{5.11-1}$$

Проекция пучка параллельных лучей может быть представлена множеством таких линий, как показано на рис. 5.37. Произвольная точка в сигнале проекции дается суммой по лучу вдоль линии $x\cos\theta_k + y\sin\theta_k = \rho_j$. Поскольку в данном случае работа идет с непрерывными величинами[23], сумма по лучу является линейным интегралом, определяемым формулой

[23] В главе 4 мы уделяли большое внимание обозначению непрерывных координат на изображении через (t, z), а дискретных — через (x, y). В то время различия были важны, поскольку разрабатывались основные концепции перехода от непрерывных переменных к дискретным. В настоящем обсуждении нам придется многократно переходить от непрерывных координат к дискретным и обратно, и соблюдение того соглашения приведет здесь только к лишней неразберихе. По этой причине, а также следуя сложившейся традиции в публикуемой литературе по этому вопросу (например, см. [Prince, Links, 2006]), вопрос о принадлежности значений координат (x, y) непрерывному или дискретному множеству решается по контексту. Если они непрерывны, в выражениях будут интегралы, иначе будет суммирование.

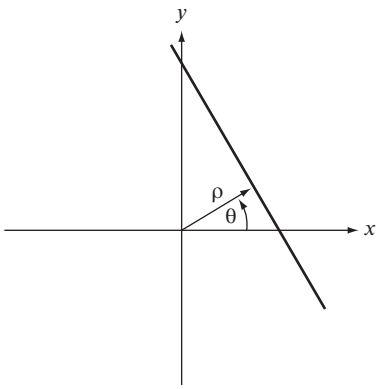


Рис. 5.36. Представление прямой линии нормалью

$$g(\rho_j,\theta_k)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\delta(x\cos\theta_k+y\sin\theta_k-\rho_j)dxdy, \qquad (5.11\text{-}2)$$

где используются свойства импульса, рассмотренные в разделе 4.5.1. Другими словами, правая часть (5.11-2) обращается в нуль всюду, кроме точек, в которых аргумент δ-функции принимает нулевое значение, т. е. интеграл берется только вдоль линии $x\cos\theta_k + y\sin\theta_k = \rho_j$. Если рассматривать все значения ρ и θ, предыдущее выражение примет общий вид

$$g(\rho,\theta)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\delta(x\cos\theta+y\sin\theta-\rho)dxdy. \qquad (5.11\text{-}3)$$

Это равенство, которое задает проекцию (линейный интеграл) $f(x,y)$ вдоль произвольной прямой на плоскости xy, является *преобразованием Радона*, упомянутым в предыдущем разделе. Иногда в уравнении (5.11-3) для обозначения преобразования Радона вместо $g(\rho,\theta)$ используют обозначение $\Re\{f(x,y)\}$ или $\Re\{f\}$,

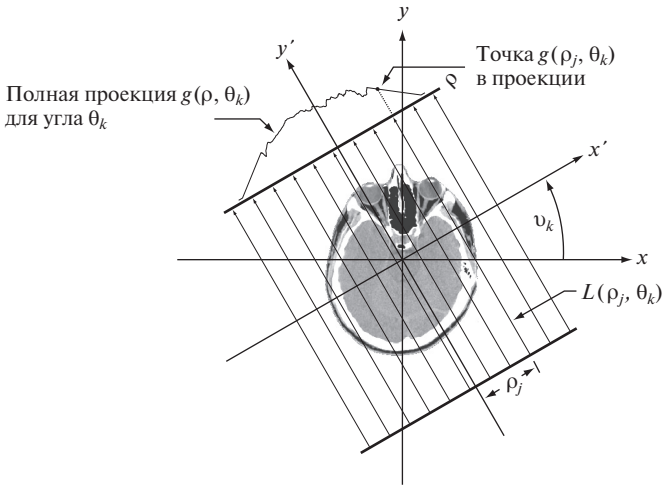


Рис. 5.37. Геометрия пучка параллельных лучей

однако обозначения, использованные в (5.11-3), более привычны. Как станет очевидно из нижеследующего рассмотрения, преобразование Радона является краеугольным камнем реконструкции по проекциям, при этом его главным приложением в области обработки изображений является компьютерная томография.

В дискретном случае выражение (5.11-3) будет выглядеть

$$g(\rho,\theta)=\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)\delta(x\cos\theta+y\sin\theta-\rho), \qquad (5.11\text{-}4)$$

где x, y, ρ и θ теперь являются дискретными переменными. Как видно, (5.11-4) есть сумма пикселей $f(x,y)$ вдоль прямой, заданной значениями θ и ρ. Если зафиксировать θ и позволить варьироваться ρ в пределах, захватывающих все изображение, то (5.11-4) в результате даст одну проекцию. Изменив θ и повторив предшествующую процедуру, получим другую проекцию и так далее. Точно таким образом были получены проекции в разделе 5.11.1.

Пример 5.17. Использование преобразования Радона для получения проекции круглой области.

■ Прежде всего проиллюстрируем, как при помощи преобразования Радона получить аналитическое выражение для проекции круглого объекта на рис. 5.38(*а*):

$$f(x,y)=\begin{cases}A & x^2+y^2\le r^2\\ 0 & \text{в остальных случаях,}\end{cases}$$

где A — константа, а r — радиус объекта. Полагаем, что центр круга расположен в начале координат плоскости xy. Поскольку объект имеет круговую симметрию, его проекции одинаковы для всех углов, так что достаточно получить проекцию для $\theta = 0$. Уравнение (5.11-3) тогда примет вид

$$g(\rho,\theta)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}f(x,y)\delta(x-\rho)dxdy=\int_{-\infty}^{\infty}f(\rho,y)dy,$$

где второе равенство следует из (4.2-10). Как отмечалось ранее, это линейный интеграл (в данном случае вдоль линии $L(\rho,0)$). Также отметим, что $g(\rho,\theta) = 0$ при $|\rho| > r$. При $|\rho| \le r$ интеграл берется от $y=-\sqrt{r^2-\rho^2}$ до $y=\sqrt{r^2-\rho^2}$. Таким образом,

$$g(\rho,\theta)=\int_{-\sqrt{r^2-\rho^2}}^{\sqrt{r^2-\rho^2}}f(\rho,y)\,dy=\int_{-\sqrt{r^2-\rho^2}}^{\sqrt{r^2-\rho^2}}A\,dy.$$

Выполняя интегрирование, получим:

$$g(\rho,\theta)=g(\rho)=\begin{cases}2A\sqrt{r^2-\rho^2} & |\rho|\le r,\\ 0 & \text{в остальных случаях,}\end{cases}$$

где использовался отмеченный выше факт, что $g(\rho,\theta) = 0$ при $|\rho| > r$. На рис. 5.38(*б*) показан результат, который согласуется с изображениями проекций, представленными на рис. 5.32 и 5.33. Отметим, что $g(\rho,\theta) = g(\rho)$; т. е. g не зависит от θ, по-

скольку объект имеет круговую симметрию относительно начала координат. ■

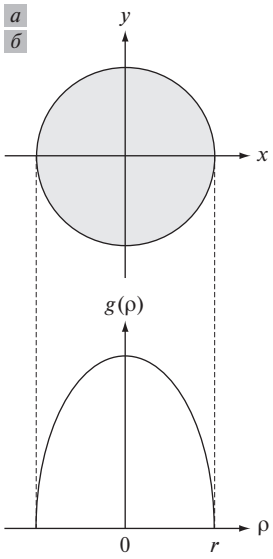


Рис. 5.38. (*а*) Круглый объект и (*б*) график его преобразования Радона, полученный аналитически. Здесь мы имеем возможность изобразить преобразование в виде графика, поскольку оно зависит лишь от одной переменной. Если g зависит и от ρ и от θ, преобразование Радона будет выглядеть как изображение, оси которого ρ и θ, а яркость пикселя пропорциональна полученному значению g в этой точке

Преобразование Радона, представленное в виде изображения в прямоугольных координатах ρ и θ, называют *синограммой* (sinogram). Такое представление аналогично представлению Фурье-спектра в виде изображения; однако в отличие от преобразования Фурье преобразование Радона $g(\rho,\theta)$ всегда является действительной функцией. Подобно преобразованию Фурье, синограмма содержит данные, необходимые для восстановления $f(x, y)$. Как и в случае с изображениями Фурье-спектра, синограмма может быть легко истолкована в случае простых областей, но при усложнении области становится все более трудной для «чтения». Рис. 5.39(*б*) является синограммой прямоугольника слева. Вертикальная и горизонтальная оси соответствуют θ и ρ соответственно. Таким образом, нижняя строка является проекцией на горизонтальную ось (т. е. $\theta = 0°$), а средняя — на вертикальную ось ($\theta = 90°$). Тот факт, что ненулевая часть нижней строки меньше, чем ненулевая часть средней строки, говорит о том, что объект вытянут в вертикальном направлении. То, что относительно своего центра синограмма симметрична и в вертикальном, и в горизонтальном направлениях, говорит о том, что объект также симметричен и параллелен осям x и y. Наконец, то, что синограмма гладкая, говорит о том, что объект однороден по яркости. Эти общие высказывания — все, что можно сказать об объекте по данной синограмме.

Для построения массива, содержащего строки одинакового размера, минимальный размер синограммы по направлению ρ соответствует максимальному размеру проекции объекта. Так, минимальный размер синограммы квадрата размерами $M \times M$, полученной с шагом 1°, будет $180 \times Q$, где Q — минимальное целое большее, чем $\sqrt{2}M$.

На рис. 5.39(*в*) представлено изображение *фантома Шеппа — Логана* — широко используемого искусственного объекта, созданного для имитации поглощения основными областями мозга, включая небольшие опухоли. Синограмма этого изображения значительно более сложна для интерпретации, как это видно на рис. 5.39(*г*). Некоторые свойства симметрии все еще можно подразумевать, но это все, что можно сказать. Визуальный анализ синограммы имеет весьма ограниченные возможности использования, но иногда он полезен при разработке алгоритмов.

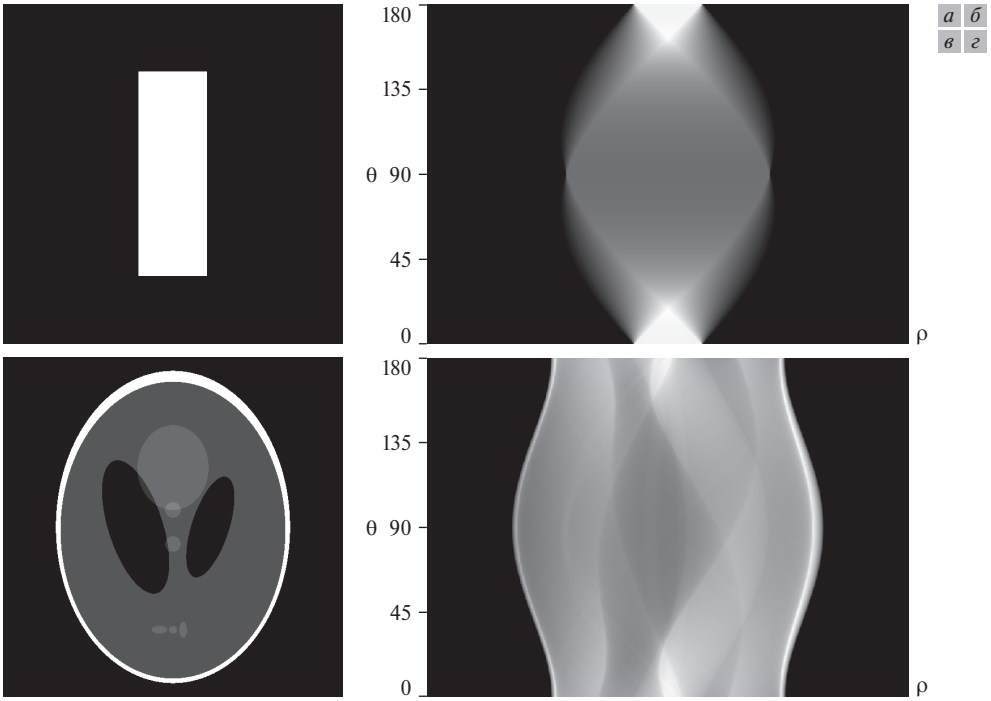


Рис. 5.39. Два изображения и их синограммы (преобразования Радона). Каждый ряд синограммы представляет собой проекцию под углом относительно вертикальной оси. Изображение (*в*) называется *фантом Шеппа—Логана*. В оригинальном виде контраст фантома чрезвычайно мал; здесь он усилен для облегчения восприятия

Главной целью КТ является получение трехмерного представления объемного объекта по его проекциям. Как было наглядно показано в разделе 5.11.1, получение изображения слоя достигается формированием обратных проекций по каждому из проецируемых направлений и их последующим суммированием. Наложение всех полученных изображений слоев дает трехмерную копию сканируемого объекта. Чтобы при помощи преобразования Радона получить формальные выражения для изображения, полученного обратным проецированием, рассмотрим сначала *единственную* точку $g(\rho_j,\theta_k)$ полной проекции $g(\rho,\theta_k)$ при фиксированном значении угла поворота θ_k (см. рис. 5.37). Формирование части изображения путем обратного проецирования этой *единственной* точки есть не что иное, как копирование линии $L(\rho_j,\theta_k)$ на изображение так, что значение каждой точки, принадлежащей данной линии, будет равно $g(\rho_j,\theta_k)$. Повторяя этот процесс для всех значений ρ_j рассматриваемой проекции (но оставляя θ равным θ_k), получим следующее выражение:

$$f_{\theta_k}(x,y) = g(\rho,\theta_k) = g(x\cos\theta_k + y\sin\theta_k,\theta_k),$$

которое дает значения изображения обратной проекции, полученной при фиксированном угле θ_k, как это показано на рис. 5.32(*б*). Поскольку равенство справедливо для любого угла θ_k, в общем виде можно записать, что изображение,

образованное *одной* обратной проекцией, полученной под углом θ, дается выражением

$$f_\theta(x,y) = g(x\cos\theta + y\sin\theta, \theta). \quad (5.11\text{-}5)$$

Окончательное изображение получается интегрированием по всем изображениям обратных проекций:

$$f(x,y) = \int_0^\pi f_\theta(x,y)d\theta. \quad (5.11\text{-}6)$$

В дискретном случае интеграл становится суммой изображений обратных проекций:

$$f(x,y) = \sum_{\theta=0}^{\pi} f_\theta(x,y), \quad (5.11\text{-}7)$$

где x, y и θ являются теперь дискретными величинами. Напомним, что согласно обсуждению в разделе 5.11.1 проекции под углами 0° и 180° являются зеркальными по отношению друг к другу, поэтому суммирование выполняется до значения угла, непосредственно предшествующего углу 180°. Так, если шаг равен 0,5°, то суммирование должно вестись до 179,5°. Изображение, сформированное описанным образом при помощи обратных проекций, называется *ламинограммой*. Очевидно, что ламинограмма является только приближением того изображения, по проекциям которого она была получена; это наглядно демонстрирует следующий пример.

Пример 5.18. Получение изображений обратных проекций из синограмм.

■ При формировании окончательных изображений обратных проекций на рис. 5.32—5.34, являющихся результатом суммирования проекций, полученных при помощи (5.11-4), использовалась формула (5.11-7). Аналогичным образом данные соотношения использовались для формирования изображений рис. 5.40(*а*) и (*б*), которые являются изображениями обратных проекций, соответствующих синограммам рис. 5.39(*б*) и (*г*) соответственно. Как и на предыдущих изображениях, здесь имеется высокий уровень размывания, поэтому очевидно, что прямое использование соотношений (5.11-4) и (5.11-7) не приводит к желаемому результату. Экспериментальные КТ-системы ранее основывались на этих соотношениях; но как будет показано в разделе 5.11.5, возможны значи-

а *б*

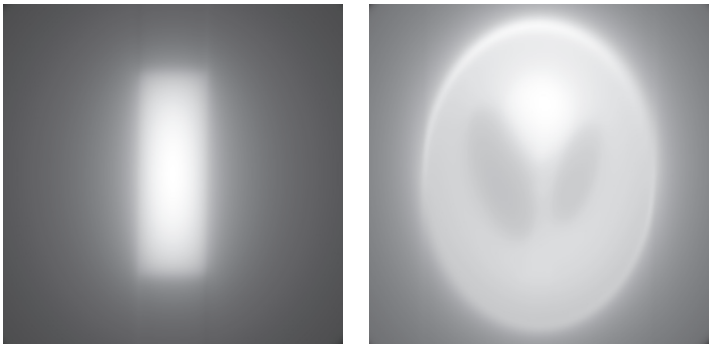


Рис. 5.40. Обратные проекции синограмм на рис. 5.39

тельные улучшения в реконструкции при помощи переформулировки подхода, основанного на обратных проекциях. ■

5.11.4. Теорема о центральном сечении[24]

В этом разделе будут приведены фундаментальные результаты, связывающие одномерное Фурье-преобразование проекции и двумерное Фурье-преобразование изображения той области, по которой была сделана проекция. Это соотношение является основой методов реконструкции, способных справиться с только что рассмотренной проблемой размывания.

Одномерное преобразование Фурье проекции $g(\rho,\theta)$ по переменной ρ при заданном угле θ дается формулой

$$G(\omega,\theta)=\int_{-\infty}^{\infty} g(\rho,\theta)e^{-i2\pi\omega\rho}d\rho, \tag{5.11-8}$$

где, как и в (4.2-16), ω — частотная переменная. Подставляя (5.11-3) вместо $g(\rho,\theta)$, получим выражение

$$\begin{aligned} G(\omega,\theta)&=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\delta(x\cos\theta+y\sin\theta-\rho)e^{-i2\pi\omega\rho}dxdyd\rho=\\ &=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\left[\int_{-\infty}^{\infty}\delta(x\cos\theta+y\sin\theta-\rho)e^{-i2\pi\omega\rho}d\rho\right]dxdy=\\ &=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-i2\pi\omega(x\cos\theta+y\sin\theta)}dxdy, \end{aligned} \tag{5.11-9}$$

где последний шаг следует из свойств импульса, рассмотренных в главе 4. Вводя новые переменные $u=\omega\cos\theta$ и $v=\omega\sin\theta$ и делая замену переменных в последнем интеграле формулы (5.11—9), получим:

$$G(\omega,\theta)=\left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-i2\pi(ux+vy)}dxdy\right]_{u=\omega\cos\theta;\,v=\omega\sin\theta}. \tag{5.11-10}$$

Как видно, полученное выражение есть преобразование Фурье $f(x,y)$ (см. формулу (4.5-7)), вычисленное при указанных значениях переменных u и v. Таким образом,

$$G(\omega,\theta)=\left[F(u,v)\right]_{u=\omega\cos\theta;\,v=\omega\sin\theta}=F(\omega\cos\theta,\omega\sin\theta), \tag{5.11-11}$$

где, как обычно, $F(u,v)$ означает двумерное преобразование Фурье функции $f(x,y)$.

Равенство (5.11-11) известно как *теорема о центральном сечении* (или *теорема о проекциях*). Она утверждает, что Фурье-преобразование проекции есть *сечение* (сделанное под тем же углом) двумерного преобразования Фурье изображения той области, проекция которой была получена. Объяснение терминологии может быть сделано при помощи рис. 5.41. Как видно из рисунка,

---

[24] В оригинале *Fourier-slice theorem*. В русскоязычной литературе она также называется *теоремой о проекциях*. — *Прим. перев.*

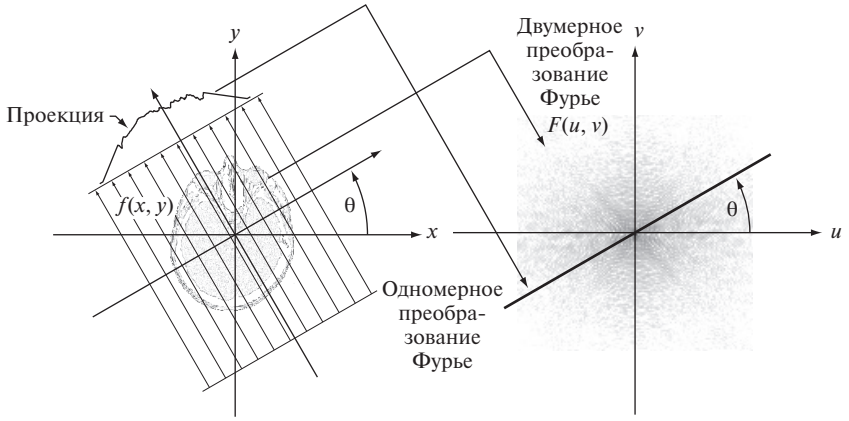


Рис. 5.41. Иллюстрация теоремы о центральном сечении. Одномерное Фурье-преобразование проекции является сечением двумерного преобразования Фурье области, проекция которой была получена. Обратим внимание на совпадение углов θ

преобразование Фурье (одномерное) произвольной проекции получается выбором значений $F(u,v)$, лежащих на прямой, проходящей под тем же углом, под которым была сделана проекция. В принципе, можно получить $f(x,y)$ просто взятием обратного преобразования Фурье от $F(u,v)$[25]. Однако это требует определенных вычислительных затрат на выполнение обратного двумерного преобразования. Подход, рассматриваемый в следующем разделе, значительно более эффективен.

5.11.5. Реконструкция по проекциям в параллельных пучках методом фильтрации и обратного проецирования

Как было показано в разделе 5.11.1 и в примере 5.18, непосредственное использование обратных проекций приводит к неприемлемо размытому результату. К счастью, существует простое решение этой проблемы, основанное на фильтрации проекций перед формированием обратных проекций. Согласно (4.5-8), двумерное обратное преобразование Фурье $F(u,v)$ записывается в виде

$$f(x,y)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}F(u,v)e^{i2\pi(ux+vy)}du\,dv\,. \qquad (5.11\text{-}12)$$

Если, как в (5.11-10) и (5.11-11), сделать замену $u=\omega\cos\theta$ и $v=\omega\sin\theta$, то дифференциалы принимают вид $du\,dv=\omega\,d\omega\,d\theta$ и (5.11-12) можно переписать в полярных координатах следующим образом:

> Соотношение $du\,dv=\omega\,d\omega\,d\theta$ следует из основ интегрального исчисления, когда для замены переменных в качестве базиса используется якобиан.

[25] Следует помнить, что при взятии обратного преобразования Фурье по-прежнему будет присутствовать размывание, поскольку данный результат эквивалентен тому, который был получен с использованием подхода, рассмотренного в предыдущем разделе.

$$f(x,y)=\int_0^{2\pi}\int_0^{\infty}F(\omega\cos\theta,\omega\sin\theta)e^{i2\pi\omega(x\cos\theta+y\sin\theta)}\omega d\omega\, d\theta. \qquad (5.11\text{-}13)$$

Тогда, используя теорему о центральном сечении, получаем

$$f(x,y)=\int_0^{2\pi}\int_0^{\infty}G(\omega,\theta)e^{i2\pi\omega(x\cos\theta+y\sin\theta)}\omega d\omega\, d\theta. \qquad (5.11\text{-}14)$$

Разделив данный интеграл на два выражения — одно для θ в диапазоне от 0° до 180°, а второе — от 180° до 360°, и используя тот факт, что $G(\omega, \theta+180°) = G(-\omega,\theta)$ (см. задачу 5.32), можно записать (5.11-14) в виде

$$f(x,y)=\int_0^{\pi}\int_{-\infty}^{\infty}|\omega|G(\omega,\theta)e^{i2\pi\omega(x\cos\theta+y\sin\theta)}d\omega\, d\theta. \qquad (5.11\text{-}15)$$

При интегрировании по ω выражение ($x\cos\theta + y\sin\theta$) можно считать константой, которая, согласно (5.11-1), равна ρ. Тогда (5.11-15) можно переписать в виде

$$f(x,y)=\int_0^{\pi}\left[\int_{-\infty}^{\infty}|\omega|G(\omega,\theta)e^{i2\pi\omega\rho}d\omega\right]_{\rho=x\cos\theta+y\sin\theta}d\theta. \qquad (5.11\text{-}16)$$

Выражение в квадратных скобках имеет форму одномерного *обратного* преобразования Фурье (см. (4.2-17)) с добавочным членом |ω|, который, основываясь на обсуждении в разделе 4.7, можно считать *одномерной* фильтр-функцией. Заметим, что |ω| является *пилообразным*[26] фильтром[27] (см. рис. 5.42(*а*)). Эта функция не интегрируема, поскольку ее амплитуда возрастает до +∞ в обоих направлениях, так что обратное Фурье-преобразование не определено. Теоретически это можно преодолеть использованием *обобщенных дельта-функций*. Практический подход заключается в использовании ограничивающей функции окна, так что характеристика пилообразного фильтра становится равной нулю вне заданного частотного интервала. Т. е. функция окна *ограничивает полосу* пилообразного фильтра.

Простейший способ ограничить полосу частот функции состоит в использовании в частотной области прямоугольного спектрального окна. Однако, как было видно на рис. 4.4, прямоугольная функция приводит к нежелательным явлениям «звона», поэтому вместо нее используют окно гладкой формы. На рис. 5.42(*а*) показан график пилообразного фильтра после ограничения его полосы прямоугольным окном, а на рисунке (*б*) — соответствующий результат в пространственной области, полученный при помощи обратного Фурье-преобразования. Как и следовало ожидать, после такого ограничения фильтр дает заметный звон в пространственной области. Из главы 4 известно, что фильтрация в частотной области (произведение спектров) эквивалентна свертке в пространственной области, поэтому фильтрация с функцией, содержащей звон, приведет к результату, в котором также будет присутствовать звон. Помочь в такой ситуации может использование в качестве спектрального окна гладкой

---

[26] В оригинале *ramp filter*. — *Прим. перев.*

[27] Пилообразный фильтр иногда называют *Рам — Лак-фильтром* по именам первых предложивших его авторов [Ramachandran, Lakshminarayanan, 1971].

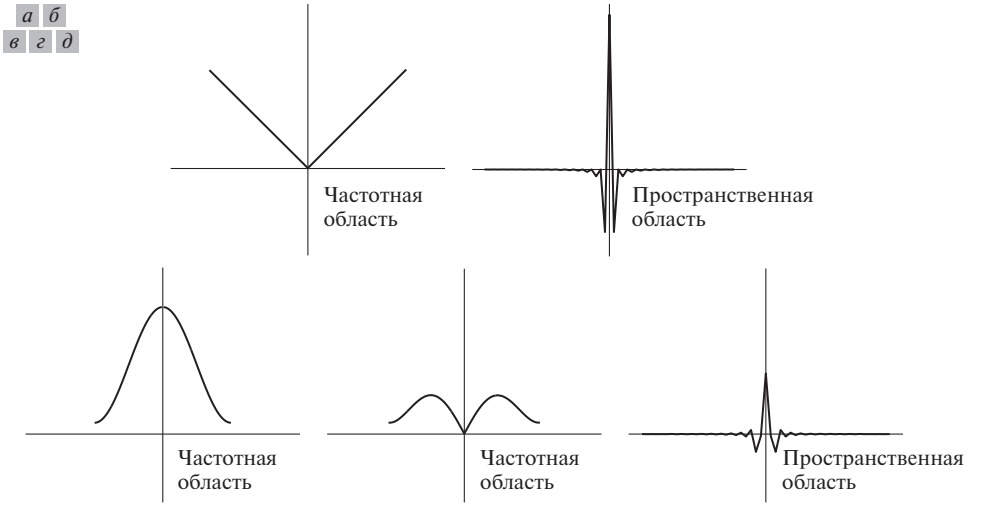


Рис. 5.42. (*а*) График фильтра |ω| в частотной области после ограничения полосы функцией окна. (*б*) Представление фильтра в пространственной области. (*в*) Спектральное окно Хэмминга. (*г*) Ограниченный фильтр как произведение (*а*) на (*в*). (*д*) Представление полученного произведения в пространственной области (заметим уменьшение звона)

функции. *M*-точечная дискретная функция окна, часто используемая при реализации одномерных ДПФ, дается формулой

$$h(\omega)=\begin{cases} c+(c-1)\cos\dfrac{2\pi\omega}{M-1} & 0\le\omega\le M-1, \\ 0 & \text{в остальных случаях.} \end{cases} \qquad (5.11\text{-}17)$$

При $c = 0{,}54$ эта функция называется *окном Хэмминга* (в честь Ричарда Хэмминга), а при $c = 0{,}5$ называется *окном Ханна* (в честь Ю. фон Ханна)[28]. Основная разница между окнами Хэмминга и Ханна состоит в том, что в последнем случае крайние точки оказываются равными нулю. Применительно к обработке изображений разница в результатах между ними обычно незначительна.

На рис. 5.42(*в*) изображен график окна Хэмминга, а на (*г*) — результат произведения этого окна и пилообразного фильтра с ограниченной полосой на рис. (*а*). На рисунке (*д*) показано представление этого произведения в пространственной области, полученное вычислением обратного ДПФ. Сравнение графиков (*д*) и (*б*) показывает, что звон в варианте (*д*) значительно уменьшен (соотношение высоты пика к глубине впадины на (*б*) и (*д*) составляет 2,5 и 3,4 соответственно). С другой стороны, ширина центрального лепестка (*д*) несколько шире, чем (*б*), поэтому следует ожидать, что результаты обратного проецирования, выполненного с использованием окна Хэмминга, будут иметь меньше звона, но несколько большее размывание. Как показано в примере 5.19, именно это имеет место.

[28] Иногда окно Ханна называют окном Ханнинга по аналогии с окном Хэмминга, что неправильно. Подобная неточность терминологии является частым источником ошибок.

Напомним (см. (5.11-8)), что $G(\omega,\theta)$ есть одномерное преобразование Фурье от $g(\rho,\theta)$, которое является *единственной* проекцией, полученной при фиксированном угле θ. Выражение (5.11-16) означает, что для вычисления *полного* изображения обратной проекции $f(x, y)$ необходимо выполнить следующие действия:

1. Вычислить одномерное преобразование Фурье каждой проекции.
2. Умножить каждое Фурье-преобразование на фильтр-функцию $|\omega|$, которая, как объяснялось выше, должна быть предварительно умножена на подходящее окно (например Хэмминга).
3. Выполнить обратное преобразование Фурье каждой полученной в результате фильтрованной функции.
4. Просуммировать все одномерные преобразования из шага 3.

Поскольку используется фильтр-функция, такой метод реконструкции изображения естественно называть *методом фильтрации и обратного проецирования*. На практике данные являются дискретными, так что все вычисления в частотной области выполняются при использовании одномерного алгоритма быстрого преобразования Фурье (БПФ), а фильтрация выполняется использованием тех же основных процедур, объясненных в главе 4 для двумерных функций. В качестве альтернативного варианта можно выполнять фильтрацию и в пространственной области, используя свертку, как объясняется позже в данном разделе.

Предшествующее обсуждение относилось к аспектам ограничения фильтрованной обратной проекции. Как и в случае любой цифровой системы, необходимо уделить внимание вопросу частоты дискретизации. Из главы 4 известно, что выбор частоты дискретизации оказывает существенное влияние на результаты обработки изображений. При обсуждении данного вопроса дискретизации имеются два аспекта. Первым является число используемых лучей, которое задает число отсчетов в каждой проекции. Второй — число углов, под которыми берутся проекции, определяющее число восстанавливаемых изображений, суммирование которых дает в результате искомое изображение. Недостаточное разрешение приводит к наложению спектров, что, как было показано в главе 4, может проявиться в виде артефактов, таких как полосы. Более подробно вопросы дискретизации КТ будут рассмотрены в разделе 5.11.6.

Пример 5.19. Реконструкция изображения методом фильтрации и обратного проецирования.

■ Цель данного примера — показать реконструкцию с использованием обратных проекций, фильтрованных (*а*) пилообразным фильтром и (*б*) пилообразным фильтром, модифицированным окном Хэмминга. Эти фильтрованные обратные проекции сравниваются с результатами «сырой» обратной проекции на рис. 5.40. Для того чтобы увидеть разницу, достигаемую только благодаря фильтрации, остальные параметры были выбраны теми же, что и при формировании рис. 5.40. Проекции генерировались с шагом 0,5°; расстояние между лучами — один пиксель; размеры изображений — 600×600 пикселей, так что длина диагонали составляла около 849 пикселей. Следовательно, чтобы обеспечить полное покрытие области при ее повороте на 45° и 135°, достаточно 849 лучей.

На рис. 5.43(*а*) показана реконструкция прямоугольника при использовании пилообразного фильтра. Самой примечательной особенностью данного

результата является отсутствие какого-либо заметного размывания. Однако, как уже говорилось выше, присутствует звон, видимый как слабые линии, особенно вблизи углов прямоугольника. Эти линии лучше заметны на увеличенном участке рис. 5.43(*в*). Ограничение фильтра окном Хэмминга существенно помогает с решением проблемы звона, однако ценой небольшого размывания, как это можно видеть на рис. 5.43(*б*) и (*г*). По сравнению с рис. 5.40(*а*) улучшения очевидны (даже с использованием только пилообразного фильтра). Изображение фантома не имеет резких перепадов яркости и выступающих контуров, как у прямоугольника, поэтому звон незаметен даже при использовании только одного пилообразного фильтра, как это можно видеть на рис. 5.44(*а*). Добавление модификации его окном Хэмминга приводит к слегка размытому изображению, как это видно на рис. 5.44(*б*). Оба эти результата являются значительными улучшениями по сравнению с рис. 5.40(*б*), иллюстрирующими преимущества подхода с использованием фильтрованных обратных проекций.

В большинстве приложений КТ (особенно в медицине) такие артефакты, как звон, являются серьезным недостатком, поэтому прилагаются значительные усилия к их уменьшению. Среди других решений помочь в уменьшении этих эффектов может подстройка алгоритмов фильтрации, а также увеличение числа детекторов, как это объяснялось в разделе 5.11.2. ■

а *б*
в *г*

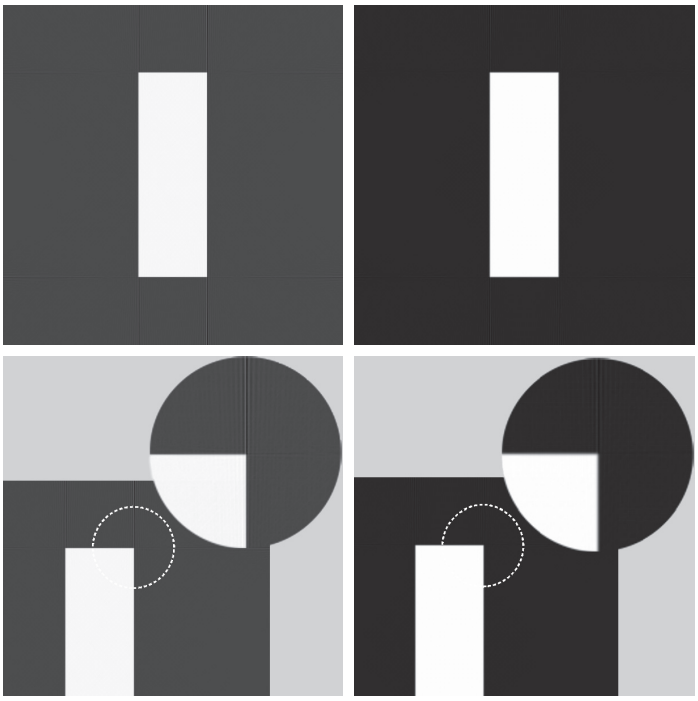
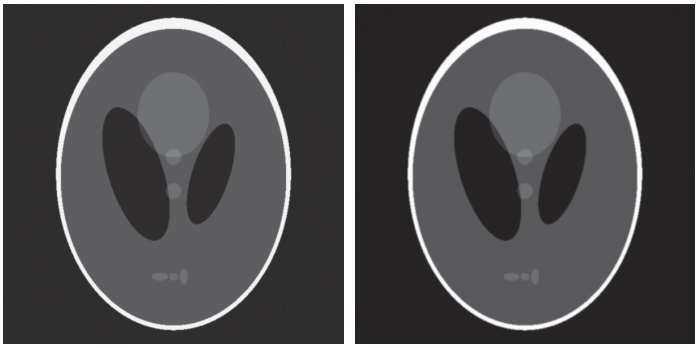


Рис. 5.43. Реконструкция изображения прямоугольника с использованием обратных проекций, фильтрованных: (*а*) пилообразным фильтром; (*б*) пилообразным фильтром, модифицированным окном Хэмминга. Во втором ряду показаны увеличенные участки изображений первого ряда. Сравните с рис. 5.40(*а*)



а б

Рис. 5.44. Реконструкция изображения фантома головы с использованием: (*а*) пилообразного фильтра; (*б*) пилообразного фильтра, модифицированного окном Хэмминга. Сравните с рис. 5.40(*б*)

Вышеприведенное рассмотрение базировалось на формировании фильтрованных обратных проекций в частотной области при использовании ДПФ. Однако из теоремы о свертке в главе 4 известно, что эквивалентный результат можно получить и при использовании пространственной свертки. В частности заметим, что в формуле (5.11-16) член, находящийся в квадратных скобках, представляет собой обратное преобразование Фурье произведения частотных представлений двух функций, которое, в соответствии с теоремой о свертке, эквивалентно свертке пространственных представлений (т. е. обратных Фурье-преобразований) этих двух функций. Другими словами, обозначив через $s(\rho)$ обратное Фурье-преобразование[29] от $|\omega|$, можно переписать (5.11-16) в виде

$$\begin{aligned} f(x,y) &= \int_0^{\pi} \left[\int_{-\infty}^{\infty} |\omega| G(\omega,\theta) e^{i2\pi\omega\rho} d\omega \right]_{\rho = x\cos\theta + y\sin\theta} d\theta = \\ &= \int_0^{\pi} \left[s(\rho) \bigstar g(\rho,\theta) \right]_{\rho = x\cos\theta + y\sin\theta} d\theta = \\ &= \int_0^{\pi} \left[\int_{-\infty}^{\infty} g(\rho,\theta) s(x\cos\theta + y\sin\theta - \rho) d\rho \right] d\theta, \end{aligned} \tag{5.11-18}$$

где, как и в главе 4, «★» обозначает свертку. Вторая строка вытекает из первой по причине, объясненной в предыдущем параграфе. Третья строка следует из фактического определения свертки, данного выражением (4.2-20).

Две нижние строки в (5.11-18) говорят об одном и том же: отдельная обратная проекция, сделанная под углом θ, может быть получена при помощи свертки соответствующей проекции $g(\rho,\theta)$ и обратного Фурье-преобразования пилообразного фильтра $s(\rho)$. Как и ранее, полное изображение обратной проекции получается интегрированием (суммированием) всех изображений отдельных

[29] Если используется модификация функции, например, окном Хэмминга, то обратное Фурье-преобразование выполняется над модифицированной пилообразной функцией. Также здесь опять игнорируется отмечавшийся выше вопрос существования непрерывного обратного преобразования Фурье, поскольку все операции выполняются над дискретными данными конечной длины.

обратных проекций. С точностью до ошибок округления в вычислениях результаты, полученные при использовании свертки или ДПФ, будут идентичными. При практической реализации КТ обычно возвращаются к использованию свертки из-за ее большей вычислительной эффективности; так что большинство современных систем КТ использует именно такой подход. Преобразование Фурье играет центральную роль в теоретических формулировках и разработке алгоритмов (например, обработка КТ-изображений в MATLAB базируется на БПФ). Также следует заметить, что нет необходимости запоминать все изображения отдельных обратных проекций при реконструкции. Вместо этого формируется изображение, являющееся текущей суммой, обновляемое прибавлением очередного изображения каждой из отдельных обратных проекций. По окончании процедуры изображение с текущими суммами будет равно общей сумме всех обратных проекций.

В заключение мы обращаем внимание на тот факт, что пилообразный фильтр (даже если он модифицирован ограничивающей функцией) обнуляет нулевой коэффициент (постоянную составляющую) в частотной области, а значит, каждое из изображений обратных проекций будет иметь нулевое среднее значение яркости (см. рис. 4.20). Как следствие, эти изображения будут иметь как положительные, так и отрицательные значения пикселей. После суммирования всех обратных проекций и формирования финального изображения часть пикселей по-прежнему будет иметь отрицательные значения.

Существует несколько путей решения данной проблемы. Если неизвестно, чему должно равняться среднее значение яркости, простейший подход заключается в допущении отрицательных величин и масштабировании значений яркости, при использовании процедуры, описанной формулами (2.6-10) и (2.6-11). Именно такой подход и применялся в данном разделе. Если имеются сведения о том, чему должно быть равно «типичное» значение средней яркости, это значение может быть добавлено в фильтр в частотной области, смещая тем самым характеристику пилообразного фильтра и предотвращая обнуление нулевого коэффициента (см. рис. 4.31(*в*)). При работе в пространственной области со сверткой сама операция ограничения длины пространственного фильтра (обратного Фурье-преобразования пилообразного фильтра) предохраняет от появления нулевого среднего значения, тем самым вообще устраняя проблему обнуления.

5.11.6. Реконструкция на основе фильтрованных обратных проекций с веерным пучком

До настоящего момента обсуждалась КТ в параллельных лучах. Благодаря своей простоте и наглядности такая геометрия формирования изображения традиционно используется при введении в проблему компьютерной томографии. Однако современные системы КТ используют геометрию веерного пучка (см. рис. 5.35), что и является темой обсуждения в оставшейся части данного раздела.

На рис. 5.45 показана основная геометрия веерного пучка, где детекторы располагаются на дуге окружности, а угловые шаги источника предполагаются одинаковыми. Пусть $p(\alpha,\beta)$ обозначает проекцию веерного пучка, где α — угловое положение конкретного детектора по отношению к *центральному лучу*,

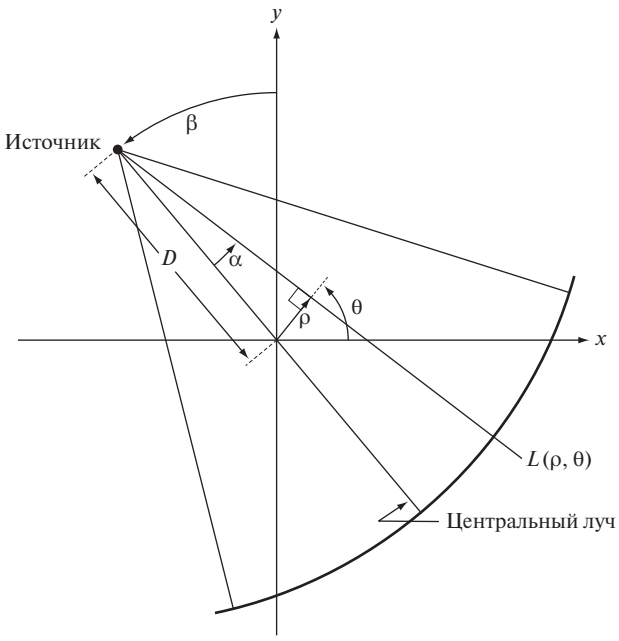


Рис. 5.45. Основная геометрия веерного пучка. Линия, проходящая от источника через начало координат (которое предполагается центром вращения источника), называется *центральным лучом*

а β — угол поворота источника по отношению к оси *y*, как показано на рисунке. На рис. 5.45 показано, что отдельный луч веерного пучка может быть представлен в форме с нормалью $L(\rho,\theta)$, что совпадает с представлением луча в геометрии параллельного пучка, рассмотренной в предыдущем разделе. Это позволяет использовать результаты параллельного пучка в качестве отправной точки для вывода соответствующих уравнений для случая геометрии веерного пучка. Приступим к выводу формул метода реконструкции на базе алгоритма свертки и обратной проекции для случая проекций веерного пучка. Перейдем к этому при помощи вывода фильтрованной обратной проекции с веерным пучком, основанной на свертке[30].

Как показано на рис. 5.45, параметры прямой $L(\rho,\theta)$ связаны с параметрами веерного пучка соотношениями

$$\theta = \beta + \alpha \tag{5.11-19}$$

и

$$\rho = D\sin\alpha, \tag{5.11-20}$$

где *D* — расстояние от центра источника до начала координат плоскости *xy*.

[30] Теорема о центральном сечении была выведена для геометрии параллельного пучка и напрямую неприменима к веерному пучку. Однако выражения (5.11 19) и (5.11 20) дают основу для сведения геометрии веерного пучка к геометрии параллельного пучка, позволяя тем самым использовать разработанный в предыдущем разделе метод фильтрации и обратного проецирования в ситуации, когда теорема о центральном сечении применима. Этот вопрос будет рассмотрен более подробно в конце данного раздела.

Формула свертки и обратной проекции для геометрии параллельного пучка задана выражением (5.11-18). Без потери общности можно предполагать, что мы рассматриваем объекты, расположенные только внутри круговой области радиуса T вокруг начала координат плоскости. Тогда $g(\rho,\theta) = 0$ для $|\rho| > T$ и формула (5.11-18) принимает вид

$$f(x,y) = \frac{1}{2}\int_0^{2\pi}\int_{-T}^{T} g(\rho,\theta)s(x\cos\theta + y\sin\theta - \rho)d\rho d\theta, \qquad (5.11\text{-}21)$$

где использовано то отмеченное в разделе 5.11.1 обстоятельство, что проекции, отвечающие углам, различающимся на 180°, являются зеркальными по отношению друг к другу. По этой причине верхний предел внешнего интеграла в (5.11-21) расширен до полного круга, как это требуется для перехода к геометрии веерного пучка, где детекторы расположены по кругу.

Требуется перейти к интегрированию по переменным α и β. Чтобы сделать это, начнем с перехода к полярным координатам (r,φ). Положим $x = r\cos\varphi$ и $y = r\sin\varphi$, откуда следует

$$x\cos\theta + y\sin\theta = r\cos\varphi\cos\theta + r\sin\varphi\sin\theta = r\cos(\theta - \varphi). \qquad (5.11\text{-}22)$$

Используя этот результат, можно выразить (5.11-21) в виде

$$f(x,y) = \frac{1}{2}\int_0^{2\pi}\int_{-T}^{T} g(\rho,\theta)s[r\cos(\theta-\varphi) - \rho]d\rho d\theta.$$

Данное выражение есть не что иное, как формула реконструкции в параллельном пучке, записанная в полярных координатах. Однако интегрирование все еще ведется по переменным ρ и θ. Для перехода к интегрированию по переменным α и β требуется преобразовать координаты в соответствии с формулами (5.11-19) и (5.11-20):

$$f(r,\varphi) = \frac{1}{2}\int_{-\alpha}^{2\pi-\alpha}\int_{\arcsin(-T/D)}^{\arcsin(T/D)} g(D\sin\alpha, \beta+\alpha)s[r\cos(\beta+\alpha-\varphi) - D\sin\alpha]D\cos\alpha d\alpha d\beta, \qquad (5.11\text{-}23)$$

где использовано равенство $d\rho d\theta = D\cos(\alpha)d\alpha d\beta$ (см. объяснение формулы (5.11-13)).

Это выражение можно еще упростить. Прежде всего отметим, что пределы интегрирования по β от $-\alpha$ до $2\pi - \alpha$ охватывает полный диапазон в 360°. Поскольку все функции, зависящие от β, являются периодическими с периодом 2π, пределы интегрирования внешнего интеграла могут быть заменены на 0 и 2π. Далее рассмотрим выражение $\alpha_m = \arcsin(T/D)$. Поскольку за пределами области радиуса T (при $|\rho| > T$) функция g обращается в нуль, то $g = 0$ при $|\alpha| > \alpha_m$ (см. рис. 5.46). Поэтому можно изменить пределы внутреннего интеграла на $-\alpha_m$ и α_m соответственно. Наконец, рассмотрим прямую $L(\rho,\theta)$ на рис. 5.45. Сумма по лучу из веерного пучка должна быть равна сумме по тому же лучу из параллельного пучка (сумма по лучу есть сумма значений всех пикселей вдоль этой прямой, так что результат должен быть одним и тем же для конкретного луча, независимо от выбранной системы координат). Это выполняется для любого луча при взаимном соответствии значений (α,β) и (ρ,θ). Обозначая через $p(\alpha,\beta)$ проекцию в веерном пучке, можно записать, что $p(\alpha,\beta) = g(\rho,\theta)$ и, согласс-

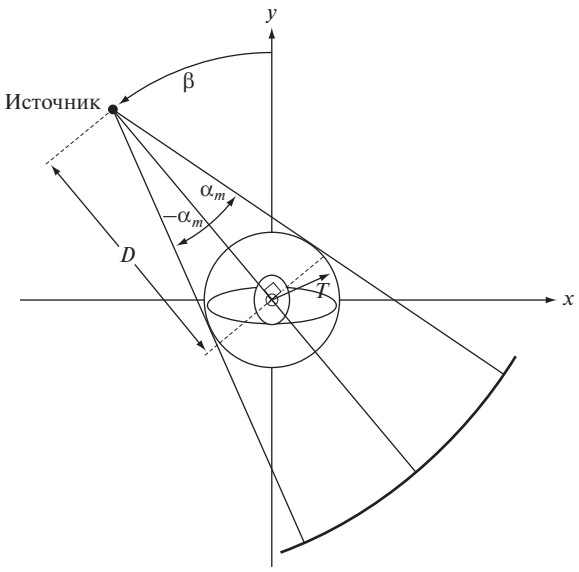


Рис. 5.46. Максимальное значение α, требуемое, чтобы захватить область интереса

но (5.11-19) и (5.11-20), что $p(\alpha,\beta) = g(D\sin\alpha, \alpha+\beta)$. Подставляя эти наблюдения в (5.11-23), получим выражение

$$f(r,\varphi)=\frac{1}{2}\int_0^{2\pi}\int_{-\alpha_m}^{\alpha_m} p(\alpha,\beta)s\left[r\cos(\beta+\alpha-\varphi)-D\sin\alpha\right]D\cos\alpha\,d\alpha\,d\beta. \qquad (5.11\text{-}24)$$

Это фундаментальная формула реконструкции в веерном пучке, основанная на фильтрации и обратном проецировании.

Формула (5.11-24) может быть преобразована и дальше, чтобы привести ее в более удобную форму свертки. В соответствии с чертежом на рис. 5.47 можно показать (задача 5.33), что

$$r\cos(\beta+\alpha-\varphi)-D\sin\alpha=R\sin(\alpha'-\alpha), \qquad (5.11\text{-}25)$$

где R — расстояние от источника до произвольной точки веерного луча, а α' — угол между этим лучом и центральным лучом. Заметим, что R и α' определяются значениями r, φ и β. Подставляя (5.11-25) в (5.11-24), получим

$$f(r,\varphi)=\frac{1}{2}\int_0^{2\pi}\int_{-\alpha_m}^{\alpha_m} p(\alpha,\beta)s[R\sin(\alpha'-\alpha)]D\cos\alpha\,d\alpha\,d\beta. \qquad (5.11\text{-}26)$$

Можно показать (задача 5.34), что

$$s(R\sin\alpha)=\left(\frac{\alpha}{R\sin\alpha}\right)^2 s(\alpha). \qquad (5.11\text{-}27)$$

Воспользовавшись этим выражением, можно переписать (5.11-26) в виде

$$f(r,\varphi)=\int_0^{2\pi}\frac{1}{R^2}\left[\int_{-\alpha_m}^{\alpha_m} q(\alpha,\beta)h(\alpha'-\alpha)\,d\alpha\right]d\beta, \qquad (5.11\text{-}28)$$

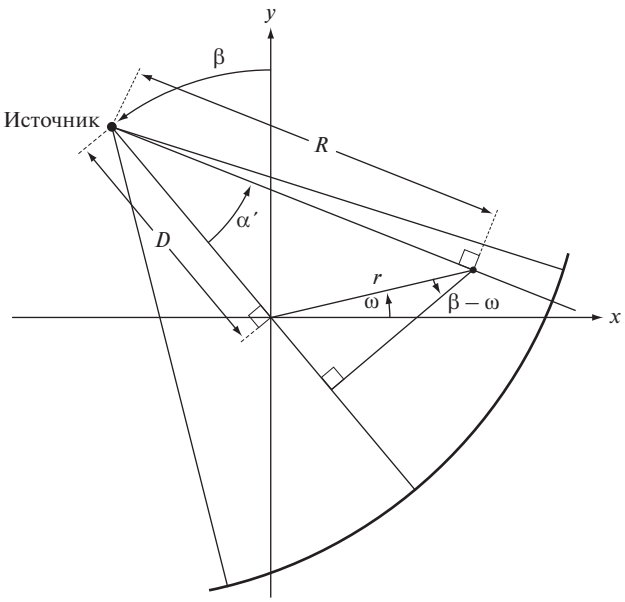


Рис. 5.47. Полярное представление произвольной точки луча из веерного пучка

где

$$h(\alpha)=\frac{1}{2}\left(\frac{\alpha}{\sin\alpha}\right)^2 s(\alpha) \tag{5.11-29}$$

и

$$q(\alpha,\beta)=p(\alpha,\beta)D\cos\alpha. \tag{5.11-30}$$

Внутренний интеграл в (5.11-28) представляет собой свертку; тем самым показано, что формула реконструкции изображения (5.11-24) может быть реализована как свертка функций $q(\alpha,\beta)$ и $h(\alpha)$. В отличие от формулы реконструкции для параллельных проекций, формула реконструкции для проекций в веерных пучках содержит член $1/R^2$, который является весовым коэффициентом, обратно пропорциональным квадрату расстояния от источника. Особенности вычислительной реализации формулы (5.11-28) остаются за рамками настоящего обсуждения; для подробного рассмотрения данного вопроса см. [Kak, Slaney, 2001]).

Вместо прямой реализации формулы (5.11-28) часто (особенно в компьютерном моделировании) применяется подход, состоящий в (1) преобразовании геометрии веерного пучка в геометрию параллельного пучка на основе соотношений (5.11-19) и (5.11-20) и (2) применении подхода к реконструкции на основе параллельного пучка, рассмотренного в разделе 5.11.5. В завершение данного раздела приведем пример того, как это можно сделать. Ранее уже отмечалось, что проекция на основе веерного пучка p, взятая под углом β, имеет соответственную проекцию в параллельном пучке g, взятую под соответствующим углом θ. Таким образом,

$$p(\alpha,\beta)=g(\rho,\theta)=g(D\sin\alpha,\alpha+\beta), \tag{5.11-31}$$

где второе равенство следует из (5.11-19) и (5.11-20).

Пусть $\Delta\beta$ означает угловое приращение между двумя соседними проекциями на основе веерных пучков, и пусть $\Delta\alpha$ — угловое приращение между лучами, которое задает число отсчетов в каждой проекции. Наложим следующее ограничение:

$$\Delta\beta = \Delta\alpha = \gamma. \tag{5.11-32}$$

Тогда $\beta = m\gamma$, а $\alpha = n\gamma$ для некоторых целых значений m и n, и можно переписать (5.11-31) в виде:

$$p(n\gamma, m\gamma) = g\left[D\sin n\gamma, (m+n)\gamma\right]. \tag{5.11-33}$$

Это выражение показывает, что n-й луч в m-й радиальной проекции равен n-му лучу в $(m+n)$-й параллельной проекции. Член $D \sin \gamma$ в правой части (5.11-33) означает, что параллельные проекции, полученные в результате преобразования из проекций веерных пучков, оказываются дискретизованы неоднородно. Данный факт может привести к появлению артефактов размывания, звона и наложения спектров, если интервалы $\Delta\alpha$ и $\Delta\beta$ оказываются слишком грубыми, что и иллюстрируется следующим примером.

Пример 5.20. Реконструкция изображения путем фильтрации и обратного проецирования веерных проекций.

■ На рис. 5.48(*а*) представлены результаты реконструкции изображения прямоугольника путем (1) построения веерных проекций с угловым шагом $\Delta\alpha = \Delta\beta = 1°$,

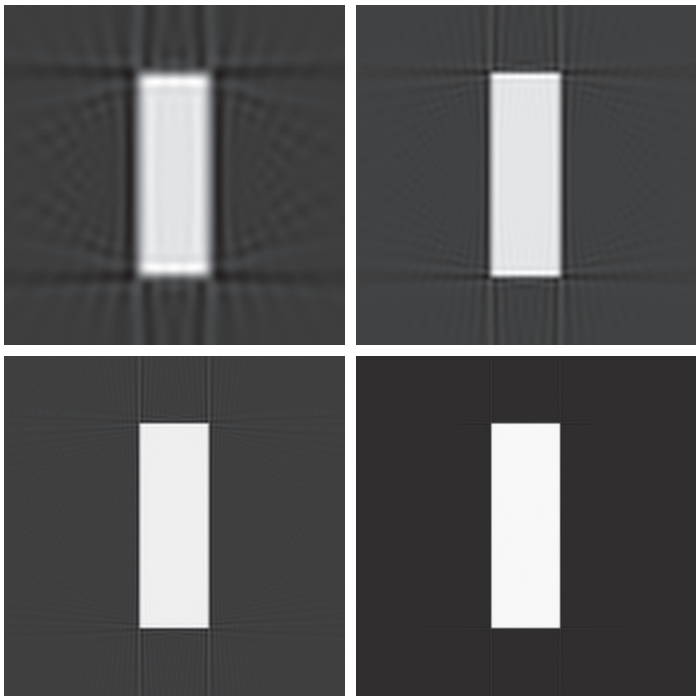


Рис. 5.48. Реконструкция изображения прямоугольника методом фильтрации и обратного проецирования веерных проекций с угловым шагом по α и β: (*а*) 1°, (*б*) 0,5°, (*в*) 0,25°, (*г*) 0,125°. Сравните (*г*) с рис. 5.43(*б*)

(2) преобразованием каждого луча веерного пучка в соответствующий луч параллельного пучка с использованием (5.11-33) и (3) использованием метода фильтрации и обратного проецирования, рассмотренного в разделе 5.11.5 для параллельных лучей. На рис. 5.48(*б*)—(*г*) показаны аналогичные результаты с угловым шагом, равным 0,5°, 0,25° и 0,125° соответственно. Во всех случаях использовалось спектральное окно Хэмминга. Использование различных значений углового шага позволяет продемонстрировать эффекты, связанные с недостаточным уровнем дискретизации.

Результат на рис. 5.48 ясно показывает, что приращение 1° является слишком грубым, поскольку и размывание, и звон совершенно очевидны. Результат (*б*) интересен в том отношении, что он заметно хуже изображения на рис. 5.43(*б*), которое было получено при использовании того же углового приращения в 0,5°. Фактически даже вариант реконструкции с приращением 0,25° не настолько хорош, как рис. 5.43(*б*). Только при использовании приращения в 0,125° удалось достичь сравнимых результатов, как видно на рис. 5.48(*д*). Угловое приращение в 0,25° дает в результате проекции с $180\times(1/0{,}25) = 720$ отсчетов, что уже близко к 849 лучам, использовавшимся при параллельной проекции в примере 5.19. Поэтому нет ничего неожиданного в том, что при $\Delta\alpha = 0{,}125°$ результаты получились сравнимыми.

Похожие результаты были получены и с фантомом головы, за тем исключением, что наложение спектров привело к значительно более сильным артефактам в виде синусоидальной интерференции. На рис. 5.49(*в*) можно видеть,

а *б*
в *г*

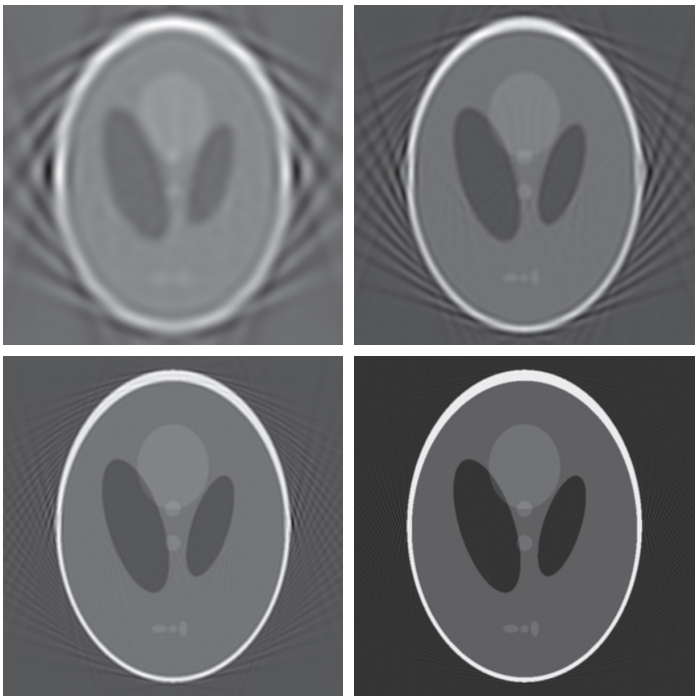


Рис. 5.49. Реконструкция изображения фантома головы методом фильтрации и обратного проецирования веерных проекций с угловым шагом по α и β: (*а*) 1°, (*б*) 0,5°, (*в*) 0,25°, (*г*) 0,125°. Сравните (*г*) с рис. 5.44(*б*)

что даже при $\Delta\alpha = \Delta\beta = 0{,}25^\circ$ еще присутствуют заметные искажения, особенно на периферии эллипса. Как и в случае прямоугольника, использование приращения в 0,125° наконец дало результат, сравнимый с изображением обратных проекций в параллельных лучах фантома головы на рис. 5.44(*б*). Эти результаты раскрывают причину, почему для того, чтобы уменьшить артефакты наложения спектров, в современных системах КТ на основе геометрии веерного пучка должны использоваться тысячи детекторов. ■

Заключение

Главные результаты этой главы, касающиеся вопросов восстановления изображений, получены в предположении, что процесс искажения может быть представлен в виде некоторой линейной трансляционно-инвариантной процедуры и последующего добавления аддитивного шума, не коррелированного с исходным изображением. Даже в тех случаях, когда эти предположения не вполне справедливы, часто бывает возможно получить полезные результаты с помощью методов, развитых в предыдущих разделах.

В основе некоторых из методов восстановления, полученных в этой главе, лежат различные критерии оптимальности. Слово «оптимальный» в данном контексте имеет строго математический смысл и не имеет отношения к оптимальности в смысле зрительного восприятия. По сути, отсутствие достаточного понимания процессов зрительного восприятия мешает сформулировать общую задачу восстановления таким образом, чтобы она учитывала возможности и особенности зрения наблюдателя. С точностью до отмеченного недостатка преимущество представленной в этой главе концепции заключается в том, что она позволяет выработать фундаментальные методы, которые твердо обоснованы с научной точки зрения и характеризуются корректным предсказуемым поведением.

Как и в главах 3 и 4, некоторые задачи восстановления, такие, например, как задача уменьшения случайного шума, решаются в пространственной области с использованием небольших масок. Частотная область оказалась идеально приспособленной для уменьшения периодического шума и для моделирования некоторых важных типов искажений, таких как смазывание изображения, обусловленное движением в процессе съемки. Мы также установили, что частотная область удобна при разработке ряда методов фильтрации, используемых для восстановления, таких как винеровская фильтрация и фильтрация по Тихонову.

Как указывалось в главе 4, частотное пространство предоставляет солидную и удобную с точки зрения использования нашей интуиции основу для проведения экспериментов. После того как для конкретной задачи найден подходящий метод решения (фильтр в частотной области), его реализация обычно осуществляется путем конструирования некоторого цифрового фильтра (в пространственной области), который приблизительно соответствует построенному в частотной области решению, но значительно быстрее работает на компьютере или на каком-то специальном оборудовании, как указывалось в конце главы 4.

Рассмотрение вопросов восстановления изображений по проекциям хотя и является вводным, представляет основы для различных аспектов обработ-

ки изображений в этой области. Как отмечалось в разделе 5.11, компьютерная томография (КТ) — главное направление приложений результатов восстановления изображений по проекциям. Несмотря на то, что наше рассмотрение было сфокусировано на рентгеновской томографии, основные принципы, сформулированные в разделе 5.11, применимы и к другим методам, таким как магнитно-резонансная томография (МРТ), позитронно-эмиссионная томография (ПЭТ), однофотонная эмиссионная компьютерная томография (ОФЭКТ), а также к методам формирования ультразвуковых изображений.

Ссылки и литература для дальнейшего изучения

В качестве дополнительного чтения по поводу представленной в разделе 5.1 линейной модели искажений рекомендуются [Castleman, 1996] и [Pratt, 1991]. В книге [Peebles, 1993] рассматриваются вопросы, связанные с функциями плотности распределения вероятностей шума и их свойствами (раздел 5.2). Книга [Papoulis, 1991] содержит более полное и детальное изложение этих вопросов на более высоком уровне. В качестве литературы по материалу раздела 5.3 можно рекомендовать [Umbaugh, 2005], [Boie, Cox, 1992], [Hwang, Haddad, 1995] и [Wilburn, 1998]. См. [Eng, Ma, 2001, 2006] касательно адаптивной медианной фильтрации. Рассмотренные в разделе 5.3 адаптивные фильтры, естественно, вписываются в общую теорию адаптивных фильтров, хорошим введением в которую является книга [Hayrin, 1996]. Фильтры раздела 5.4 являются прямым обобщением фильтров из главы 4. В качестве дополнительного чтения по материалу раздела 5.5 см. [Rosenfeld, Kak, 1982] и [Pratt, 1991].

Вопросы оценки искажающей функции (раздел 5.6) и сегодня представляют значительный интерес. Некоторые из давно известных методов оценки искажающей функции описаны в [Andrews, Hunt, 1977], [Rosenfeld, Kak, 1982], [Bates, McDonnell, 1986] и [Stark, 1987]. Поскольку искажающая функция редко бывает известна точно, в последние годы был предложен целый ряд методов, в которых особое значение придается определенным аспектам восстановления. Например, в работах [German, Reynolds, 1992] и [Hurn, Jennison, 1996] особое внимание уделено вопросам сохранения резких перепадов значений яркости для повышения резкости изображения, в то время как основной целью в работе [Boyd, Meloche, 1998] является восстановление мелких объектов на искаженных изображениях. В качестве примеров работы со смазанными изображениями укажем [Yitzhaky et al., 1998], [Harikumar, Bresler, 1999], [Mesarović, 2000] и [Giannakis, Heath, 2000]. Вопросы восстановления последовательностей изображений также представляют значительный интерес, и книга [Kokaram, 1998] может служить хорошим введением в эту область.

Методы фильтрации, рассмотренные в разделах 5.7—5.9, обсуждаются и исследуются с различных точек зрения во многих книгах и статьях по обработке изображений. Существуют два основных методологических подхода к разработке таких фильтров. Один из них, основанный на общей формулировке задачи как задачи линейной алгебры (в матричном виде), представлен в работе [Andrews, Hunt, 1977]. Этот подход характеризуется общностью и элегантностью, но недостаточно нагляден и потому труден для начинающих. Подходы, основанные

на работе непосредственно в частотной области (такой подход мы использовали в этой главе), обычно более просты для тех, кто впервые сталкивается с задачами восстановления, но они не обладают математической строгостью матричного подхода. Оба подхода приводят к одинаковым результатам, но, как показывает наш обширный опыт преподавания этого материала, студенты, впервые знакомящиеся с данной областью, предпочитают последний подход. В качестве литературы, которая дополняет материал разделов 5.7—5.10 и устраняет пробелы в нем, мы рекомендуем [Castleman, 1996], [Umbaugh, 2005] и [Petrou, Bosdogianni, 1999]. В последней работе устанавливается также красивая связь между двумерными фильтрами в частотной области и соответствующими цифровыми фильтрами. По поводу конструирования цифровых фильтров см. [Lu, Antoniou, 1992].

По вопросам компьютерной томографии важнейшими работами являются [Rosenfeld, Kak, 1982], [Kak, Slaney, 2001], и [Prince, Links, 2006]. Для дальнейшего изучения фантома Шепп—Логана см. [Shepp, Logan, 1974], а касательно дополнительных деталей возникновения фильтра Рам—Лака см. [Ramachandran, Lakshminarayanan, 1971]. Статья [O'Connor, Fessler, 2006] дает представление о современных исследованиях методов обработки изображений и сигналов в компьютерной томографии.

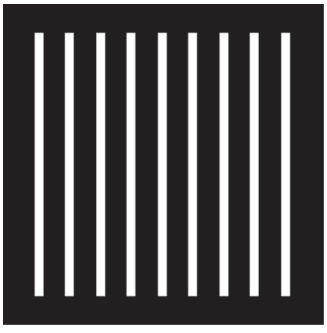
По поводу программной реализации многих приложений, рассмотренных в данной главе, см. [Gonzalez,Woods, Eddins, 2004].

Задачи

Детальное рассмотрение задач, помеченных звездочкой, можно найти на интернет-сайте книги. Также там содержатся примерные проекты, базирующиеся на материалах данной главы.

★5.1 Белые полосы на представленном тестовом изображении имеют ширину 7 пикселей и высоту 210 пикселей. Расстояние между полосами составляет 17 пикселей. Как будет выглядеть это изображение после применения

(а) Среднеарифметического фильтра размерами 3×3?
(б) Среднеарифметического фильтра размерами 7×7?
(в) Мреднеарифметического фильтра размерами 9×9?



Замечание: эта и следующие задачи, связанные с фильтрацией этого изображения, могут показаться несколько утомительными. Однако они стоят того, что-

бы потратить усилия на их решение, поскольку способствуют выработке настоящего понимания того, как действуют соответствующие фильтры. После того как Вам будет ясно, как именно конкретный фильтр видоизменяет данное изображение, Ваш ответ может представлять собой короткое словесное описание результата. Например, «результирующее изображение будет состоять из вертикальных полос шириной 3 пикселя и высотой 206 пикселей». Не забудьте описать изменение формы полос, такое как округление углов. Эффекты, возникающие на краях, где маски лишь частично накладываются на изображение, можно не принимать во внимание.

5.2 Решите задачу 5.1 для случая среднегеометрического фильтра.

★**5.3** Решите задачу 5.1 для случая среднегармонического фильтра.

5.4 Решите задачу 5.1 для случая среднего контрагармонического фильтра с $Q = 1{,}5$.

★**5.5** Решите задачу 5.1 для случая среднего контрагармонического фильтра с $Q = -1{,}5$.

5.6 Решите задачу 5.1 для случая медианного фильтра.

★**5.7** Решите задачу 5.1 для случая фильтра максимума.

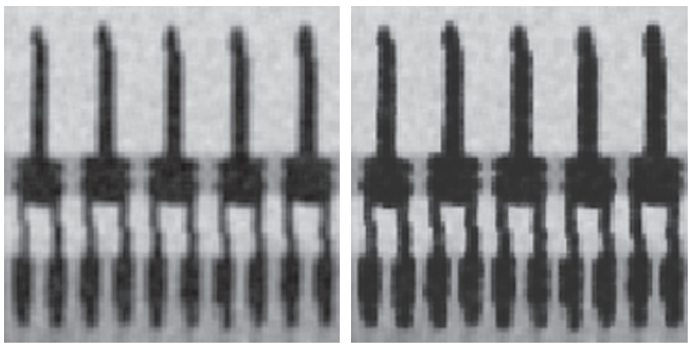
5.8 Решите задачу 5.1 для случая фильтра усеченного среднего.

★**5.9** Решите задачу 5.1 для случая фильтра срединной точки.

5.10 Два приведенных ниже изображения суть соответственно фрагменты правых верхних частей изображений на рис. 5.7(*в*) и (*г*). Таким образом, изображение слева представляет собой результат применения среднеарифметического фильтра размерами 3×3, а изображение справа — результат применения среднегеометрического фильтра тех же размеров.

★ **(а)** Объясните, почему изображение, полученное с помощью среднегеометрического фильтра, менее размыто. (*Совет*: для начала проанализируйте действие фильтров на одномерный ступенчатый яркостный перепад).

(б) Объясните, почему толщина черных деталей на правом изображении больше.



5.11 Применительно к заданному выражением (5.3-6) контрагармоническому фильтру:

(а) Объясните, почему фильтр эффективно устраняет «черный» униполярный импульсный шум при положительном значении параметра Q.

(б) Объясните, почему фильтр эффективно устраняет «белый» униполярный импульсный шум при отрицательном значении параметра Q.

(в) Объясните, почему фильтр дает плохие результаты (такие как на рис. 5.9), если знак параметра Q выбран неверно.

(г) Рассмотрите поведение фильтра при $Q = -1{,}5$.

(д) Рассмотрите поведение фильтра (при положительных и отрицательных значениях Q) в областях постоянной яркости.

★**5.12** Получите выражения для полосовых фильтров Гаусса и Баттерворта, соответствующих режекторным фильтрам в табл. 4.6.

5.13 Получите выражения для гауссова, Баттерворта и идеального узкополосного режекторного фильтра в форме выражения (4.10-5).

★**5.14** Покажите, что Фурье-преобразование непрерывной двумерной функции косинуса

$$f(x,y) = A\cos(u_0 x + v_0 y)$$

представляет собой пару комплексно-сопряженных δ-функций

$$F(u,v) = -\frac{A}{2}\left[\delta\left(u - \frac{u_0}{2\pi}, v - \frac{v_0}{2\pi}\right) + \delta\left(u + \frac{u_0}{2\pi}, v + \frac{v_0}{2\pi}\right)\right].$$

(*Совет*: используйте непрерывное преобразование Фурье в виде (4.5-7) и выразите косинус в экспоненциальной форме.)

5.15 Выведите формулу (5.4-13) из формулы (5.4-11).

★**5.16** Рассмотрим линейную трансляционно-инвариантную искажающую систему с искажающей функцией (ядром) вида

$$h(x-\alpha, y-\beta) = e^{-\left((x-\alpha)^2 + (y-\beta)^2\right)}.$$

Предположим, что на вход системы подается изображение, состоящее из линии бесконечно малой толщины, проходящей через точку $x = a$, $y = b$ и которая задается выражением $f(x,y) = \delta((x-a)—(y-b))$, где δ есть δ-функция. Какое изображение $g(x,y)$ получится на выходе, если шум отсутствует?

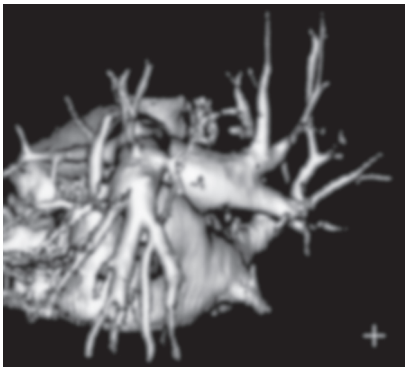
5.17 Во время съемки изображение в течении времени T_1 участвует в равномерном прямолинейном движении в вертикальном направлении. Затем движение на время T_2 меняет направление на горизонтальное. Предполагая, что время изменения направления движения пренебрежимо мало и затвор системы открывается и закрывается мгновенно, найдите выражение для искажающей функции $H(u,v)$.

★**5.18** Рассмотрите вопрос о размывании изображения в результате прямолинейного равноускоренного движения вдоль оси x. Предполагая, что изображение в начальный момент $t = 0$ покоилось, а затем двигалось равноускоренно $x_0(t) = at^2/2$ в течение времени T, найдите искажающую функцию $H(u,v)$. Можно предполагать, что затвор системы открывается и закрывается мгновенно.

5.19 Космический аппарат предназначен для передачи изображений поверхности планеты по мере ее приближения в процессе приземления. На последнем этапе приземления один из двигателей малой тяги вышел из строя, что привело к быстрому вращению аппарата вокруг вертикальной оси. Изображения, полученные во время последних двух секунд перед приземлением, оказались смазанными в результате этого кругового движения. Камера закреплена на нижней части космического аппарата вдоль его вертикальной оси и направлена вниз. По счастью, ось вращения аппарата совпадает с оптической осью

камеры, таким образом, изображения смазаны в результате движения, которое представляет собой равномерное вращение. За время съемки одного кадра аппарат поворачивается на угол $\pi/12$ радиан. Модельные предположения относительно процесса съемки состоят в том, что она осуществляется при помощи камеры с идеальным затвором, который открыт только во время поворота на угол $\pi/12$ радиан. Вертикальным сдвигом камеры за время экспозиции можно пренебречь. Разработайте метод решения задачи восстановления.

★**5.20** Изображение на рисунке ниже представляет собой размытую двумерную проекцию объемной реконструкции сердца. Известно, что каждая из пересекающихся нитей в правой нижней части рисунка имела до размывания ширину 4 пикселя, длину 20 пикселей и значение яркости, равное 255. Сформулируйте последовательность действий, при помощи которой можно определить искажающую функцию $H(u,v)$ на основе данной информации.



(Изображение предоставлено компанией G. E. Medical System)

5.21 Геометрия некоторой рентгеновской системы формирования изображений приводит к искажениям, которые могут быть смоделированы как свертка неискаженного изображения с центрально-симметричной функцией в пространственной области

$$h(x,y)=\frac{x^2+y^2-\sigma^2}{\sigma^4}e^{-\frac{x^2+y^2}{2\sigma^2}},$$

Предполагая переменные непрерывными, покажите, что в частотной области искажающая функция имеет вид

$$H(u,v)=-\sqrt{2\pi}\sigma(u^2+v^2)e^{-2\pi^2\sigma^2(u^2+v^2)}.$$

(*Совет*: обратитесь к разделу 4.9.4, строке 13 табл. 4.3 и задаче 4.26.)

★**5.22** Используя передаточную функцию задачи 5.21, найдите выражение для среднегеометрического фильтра в предположении, что отношение энергетических спектров шума и неискаженного изображения постоянно. Затем найдите выражение для винеровского фильтра.

5.23 Используя передаточную функцию задачи 5.21, найдите окончательное выражение для тихоновского фильтра.

5.24 Предполагая, что модель на рис. 5.1 является линейной и трансляционно-инвариантной, а шум и изображение некоррелированы, покажите, что энергетический спектр изображения на выходе задается выражением

$$|G(u,v)|^2=|H(u,v)|^2|F(u,v)|^2+|N(u,v)|^2.$$

Используйте (5.5-17) и (4.6-18).

5.25 В работе [Cannon, 1974] был рассмотрен восстанавливающий фильтр $R(u,v)$, удовлетворяющий условию

$$|F(u,v)|^2=|R(u,v)|^2|G(u,v)|^2,$$

и автор исходил из предпосылки об усилении энергетического спектра восстановленного изображения $|F(u,v)|^2$ так, чтобы он стал равным энергетическому спектру исходного изображения $|F(u,v)|^2$. Предполагая, что шум и изображение некоррелированы:

(а) Выразите $R(u,v)$ через $|F(u,v)|^2$, $|H(u,v)|^2$ и $|N(u,v)|^2$. (*Совет*: используйте модель на рис. 5.1, формулу (5.5-17) и задачу 5.24.)

(б) Представьте результат (*а*) в виде, аналогичном виду формулы (5.8-2).

5.26 Работая на большом телескопе, астроном замечает, что получаемые изображения слегка расфокусированы. Изготовитель сообщает астроному, что работа телескопа отвечает техническим требованиям. Линзы телескопа фокусируют изображение на ПЗС-матрицу, и затем изображение оцифровывается при помощи электроники телескопа. Невозможно попытаться улучшить ситуацию путем проведения лабораторных экспериментов с линзами и сенсорами из-за размера и веса составных частей телескопа. Астроном, наслышанный о Ваших успехах в области обработки изображений, просит Вас помочь найти подходящий метод цифровой обработки с целью небольшого повышения резкости изображений. Как бы Вы приступили к решению этой проблемы, с учетом того, что единственный доступный Вам тип изображений — это изображения звезд?

5.27 Профессор археологии, занимающийся исследованиями вопросов денежного обращения времен Римской империи, недавно узнал, что четыре римские монеты, играющие ключевую роль в его исследованиях, зарегистрированы в каталоге Британского музея в Лондоне. Посетив музей, профессор с сожалением узнает, что эти монеты недавно были украдены. Дальнейшие расспросы показали, что музей хранит фотографии всех своих экспонатов. К несчастью, фотографии интересующих профессора монет нечеткие (размыты), поэтому датировка и другие надписи маленького размера не читаются. Размывание вызвано тем, что в процессе съемки объект не находился в фокусе камеры. Вас как специалиста по обработке изображений и друга профессора просят оказать любезность и определить, можно ли с помощью компьютерной обработки восстановить изображение так, чтобы профессор смог прочитать интересующие его надписи. Вам сообщили, что та камера, с помощью которой производилась съемка монет, по-прежнему доступна, так же как и другие образцы монет того же периода. Предложите последовательность шагов для решения этой задачи.

5.28 Опишите в общих чертах преобразования Радона следующих квадратных изображений. Количественно отметьте все важнейшие характеристики Вашего описания. На рисунке (*а*) имеется одна точка в центре, на (*б*) — две точки по диагонали. Опишите Ваше решение касательно (*в*) при помощи профиля яркости. Предполагайте геометрию параллельного пучка лучей.

(а) ⋆ **(б)** **(в)**

5.29 Покажите, что преобразование Радона (5.11-3) сигнала в форме гауссиана $f(x,y) = A\exp\left(\frac{-x^2-y^2}{2\sigma^2}\right)$ будет $g(\rho,\theta) = A\sqrt{2\pi}\cdot\sigma\cdot\exp(-\rho^2)$. (Совет: обратитесь к примеру 5.17, где для упрощения интегрирования использовалась симметрия.)

5.30 ⋆ **(а)** Покажите, что преобразование Радона (5.11-3) единичного импульса $\delta(x,y)$ на плоскости $\rho\theta$ является прямой вертикальной линией, проходящей через начало координат.

(б) Покажите, что преобразование Радона импульса $\delta(x - x_0, y - y_0)$ на плоскости $\rho\theta$ является синусоидальной кривой.

5.31 Докажите верность следующих свойств преобразования Радона (5.11-3):

⋆ **(а)** Линейность: преобразование Радона является линейным оператором. (См. раздел 2.6.2 касательно определения линейного оператора.)

(б) Свойство сдвига: преобразование Радона от $f(x - x_0, y - y_0)$ будет $g(\rho - x_0\cos\theta - y_0\sin\theta, \theta)$.

⋆ **(в)** Свойство свертки: покажите, что преобразование Радона от свертки двух функций равно свертке преобразований Радона этих двух функций.

5.32 Сделайте необходимые выкладки, приводящие из (5.11-14) к (5.11-15). Вам потребуется использовать свойство $G(\omega, \theta+180°) = G(-\omega, \theta)$.

⋆**5.33** Докажите верность выражения (5.11-25).

5.34 Докажите верность выражения (5.11-27).

ГЛАВА 6

ОБРАБОТКА ЦВЕТНЫХ ИЗОБРАЖЕНИЙ

Только после долгих лет подготовки молодой художник имеет право прикоснуться к цвету — не как к средству описания, а как к средству самовыражения.

Анри Матисс

Долгое время я ограничивал себя лишь одним цветом — в качестве дисциплины.

Пабло Пикассо

Введение

Использование цвета в обработке изображений обусловлено двумя основными причинами. Во-первых, цвет является тем важным признаком, который часто облегчает распознавание и выделение объекта на изображении. Во-вторых, человек в состоянии различать тысячи различных оттенков цвета и всего лишь порядка двух десятков оттенков серого. Второе обстоятельство особенно важно при визуальном (т. е. выполняемом непосредственно человеком) анализе изображений.

Обработку цветных изображений можно условно разделить на две основные области: обработку изображений в *натуральных цветах* и обработку изображений в *псевдоцветах*. В первом случае рассматриваемые изображения обычно формируются цветными устройствами регистрации изображения, такими как цветная телевизионная камера или цветной сканер. Во втором случае задача состоит в присвоении цветов некоторым значениям интенсивности монохромного сигнала или некоторым диапазонам изменения его интенсивности. До сравнительно недавнего времени цифровая обработка цветных изображений осуществлялась по большей части на уровне псевдоцветов. За последнее десятилетие, однако, цветные устройства ввода и аппаратные средства обработки цветных изображений стали вполне доступны по ценам. Как результат, в настоящее время техника обработки изображений в натуральных цветах используется в широком диапазоне приложений, включая издательские системы, системы визуализации и Интернет.

Из последующих обсуждений станет ясно, что некоторые методы обработки полутоновых изображений, развитые в предыдущих главах, непосредственно применимы и к цветным изображениям. Другие же методы должны быть пере-

формулированы и согласованы со свойствами цветового пространства, которое строится далее в этой главе. Описанные в настоящей главе методы далеко не являются исчерпывающими; они иллюстрируют разнообразие методов, применимых в обработке цветных изображений.

6.1. Основы теории цвета

Хотя процесс восприятия и интерпретации цвета человеческим мозгом представляет собой еще не до конца исследованное психофизиологическое явление, физическая природа цвета может быть точно описана на основе экспериментальных и теоретических результатов.

В 1666 г. сэр Исаак Ньютон обнаружил, что при прохождении луча солнечного света через стеклянную призму выходящий поток лучей не является белым, но состоит из непрерывного спектра цветов, простирающихся от фиолетового цвета на одном конце до красного на другом. Как показывает рис. 6.1, спектр белого света (видимый) может быть разделен на шесть широких цветовых диапазонов: фиолетовый, синий, зеленый, желтый, оранжевый и красный[1]. При рассмотрении полного спектра на рис. 6.2 видно, что ни один цветовой диапазон не имеет ярко выраженных границ; вместо этого каждый цвет плавно переходит в другой.

Цвет, воспринимаемый человеком и некоторыми другими животными как цвет объекта, определяется, по существу, характером отраженного от объекта света. Как показано на рис. 6.2, видимый свет составляет относительно узкую часть всего диапазона длин волн электромагнитного спектра. Тело, которое равномерно отражает свет во всем видимом диапазоне длин волн, выглядит для наблюдателя как белое. Однако тело, которое отражает свет преимущественно в некотором ограниченном диапазоне видимого спектра, приобретает некоторый цвет. Так например объект зеленого цвета отражает главным образом свет с длиной волны в диапазоне от 500 до 570 нм и поглощает бóльшую часть энергии в диапазонах других длин волн.

Важную роль в науке о цвете играет выбор параметров, характеризующих свет. Когда свет является ахроматическим (неокрашенным), в роли единственной такой характеристики выступает *интенсивность*. Ахроматический свет — это то, что видит зритель на экране черно-белого телевизора, и именно такой свет был до сих пор неявной составной частью всех наших рассмотрений, связанных с обработкой изображений. Определенный в главе 2 и многократно использованный впоследствии термин *яркость* (полутоновая яркость или уровень серого) обозначает количественную меру интенсивности, которая принимает значения в диапазоне от черного до белого, с промежуточными серыми оттенками.

Хроматический (окрашенный) свет охватывает диапазон электромагнитного спектра приблизительно от 400 нм до 700 нм. Хроматические источники света характеризуются тремя основными величинами: потоком лучистой энергии,

[1] Деление спектра белого света на спектральные цвета достаточно условно. Например, иногда говорят о пяти цветах, объединяя красно-оранжевый или сине-фиолетовый участки спектра, или же о семи цветах, когда между зеленым и синим выделяется голубой участок спектра. — *Прим. перев.*

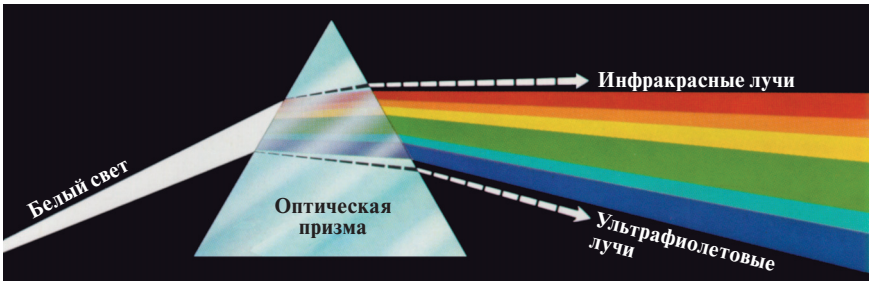


Рис. 6.1. Разложение белого света на спектральные составляющие при прохождении через оптическую призму. (Изображение предоставлено General Electric Co., Lamp Business Division)

световым потоком и светлотой. *Поток лучистой энергии*, обычно измеряемый в ваттах (Вт), — это общее количество энергии, излучаемой источником света в единицу времени. *Световой поток*, измеряемый в люменах (лм), — это поток лучистой энергии, оцениваемой по зрительному ощущению. Например, световой источник, работающий в дальнем инфракрасном диапазоне, может давать значительный поток энергии, но наблюдатель его практически не ощущает, так что световой поток такого источника почти равен нулю. Наконец, как уже обсуждалось в разделе 2.1, *светлота* является субъективной характеристикой, которая практически не поддается измерению. Она отражает уровень зрительного ощущения, производимого интенсивностью (т. е. световым потоком), и является одним из ключевых параметров для описания цветового восприятия.

Как указано в разделе 2.1.1, рецепторами глаза, отвечающими за восприятие цветов, являются колбочки. В результате всесторонних экспериментов было установлено, что все 6—7 миллионов колбочек человеческого глаза могут быть разделены по их восприимчивости к спектральному составу света на три основные группы, которые приблизительно соответствуют чувствительности к красному, зеленому и синему цветам. Примерно 65 % всех колбочек воспринимают красный свет, 33 % колбочек воспринимают зеленый свет, и только 2 % воспринимают синий цвет (однако эти колбочки являются наиболее чувствительными). На рис. 6.3 представлены экспериментальные кривые спектральной чувствительности колбочек каждой из трех групп для среднего нормального глаза. Вследствие таких спектральных характеристик человеческий глаз воспринимает цвета как различные сочетания так называемых *первичных основных цветов*: красного (R), зеленого (G) и синего (B). В 1931 г. Международная комиссия

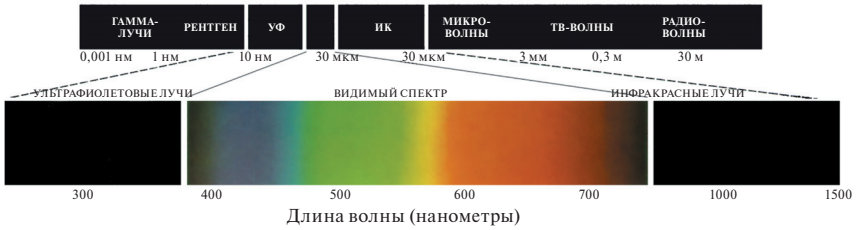


Рис. 6.2. Длины волн видимой части электромагнитного спектра. (Изображение предоставлено General Electric Co., Lamp Business Division)

по освещению (МКО) разработала стандартный набор монохроматических первичных основных цветов: синий с длиной волны 435,8 нм, зеленый — 546,1 нм и красный — 700 нм. Этот стандарт был установлен до того, как в 1965 г. стали доступны представленные на рис. 6.3 кривые спектральной чувствительности. Поэтому стандарт МКО лишь приблизительно соответствует экспериментальным данным. Как показывают рис. 6.2 и 6.3, никакой монохроматический цвет в отдельности не может быть назван красным, зеленым или синим. Кроме того, важно понимать, что наличие стандартного набора монохроматических первичных основных цветов не означает, что все цвета спектра могут быть получены на основе этих фиксированных RGB-цветов. Использование термина «*основные*» часто приводит к тому заблуждению, что *все* видимые цвета могут быть воспроизведены при смешении основных первичных цветов в различных пропорциях. Как вскоре будет видно, такое утверждение неверно, за исключением того случая, когда длина волны основных цветов также может изменяться. В этом последнем случае, однако, мы уже не будем иметь трех стандартных первичных основных цветов.

Первичные основные цвета могут складываться, что дает *вторичные основные* цвета: пурпурный (красный плюс синий), голубой (зеленый плюс синий) и желтый (красный плюс зеленый). Смешение трех первичных основных цветов или вторичного основного цвета и противоположного ему первичного в правильных пропорциях дает белый цвет. Результат такого смешения представлен на рис. 6.4(*а*), где также показаны три первичных основных цвета и их сочетания, дающие вторичные основные цвета.

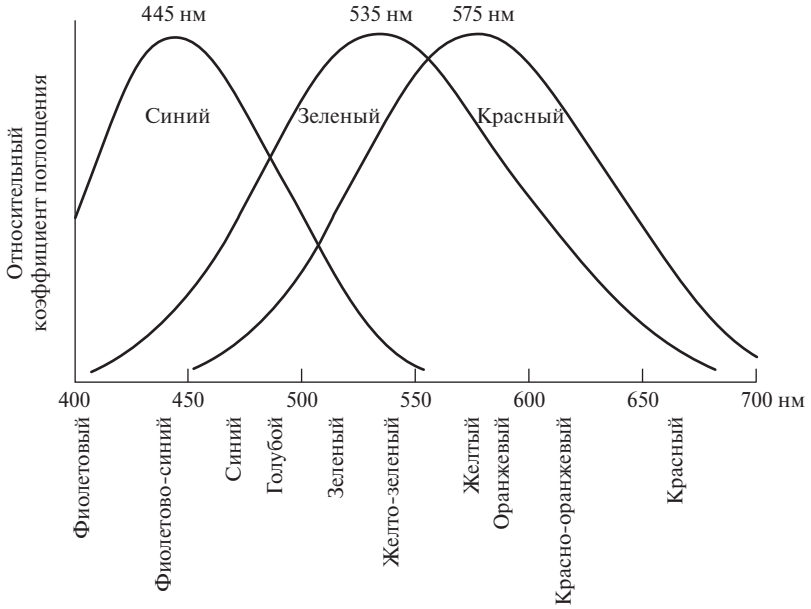


Рис. 6.3. Кривые спектральной чувствительности колбочек человеческого глаза (зависимость относительного коэффициента поглощения от длины волны)

Рис. 6.4. Первичные и вторичные основные цвета световых источников и красителей. (Изображение предоставлено General Electric Co., Lamp Business Division)

Важно различать первичные основные цвета световых источников и первичные основные цвета красителей (светофильтров). В последнем случае первичный основной цвет определяется как цвет красителя, который поглощает, или вычитает, некоторый один первичный основной цвет светового источника и отражает либо пропускает два оставшихся. Поэтому для красителей первичными основными цветами являются пурпурный, голубой и желтый, а вторичными — красный, зеленый и синий. Эти цвета показаны на рис. 6.4(*б*). Правильная комбинация трех первичных основных цветов красителей или вторичного основного цвета и противоположного ему первичного дает черный цвет.

Цветное телевидение дает пример аддитивного цветовоспроизведения (т. е. основанного на сложении первичных основных цветов световых источников). Внутренняя поверхность экрана электронно-лучевых трубок (ЭЛТ) цветных телевизоров составлена из большого числа триад — расположенных треугольником точек люминофоров. При возбуждении электронным лучом каждая точка триады способна излучать свет одного из первичных основных цветов. Интенсивность свечения красного люминофора модулируется с помощью расположенной внутри кинескопа электронной пушки, которая генерирует импульсы в соответствии с энергией, измеренной телекамерой в красном диапазоне. Управление излучением зеленого и синего люминофоров осуществляется аналогично с использованием своих электронных пушек. Эффект, наблюдаемый

на экране телевизионного приемника, состоит в том, что три первичных основных цвета от каждого люминофора смешиваются и воспринимаются чувствительными к цветам колбочками глаза как полноценное цветное изображение. Смена изображений со скоростью тридцать кадров в секунду[2] делает иллюзию непрерывного воспроизведения цветного изображения на экране полной.

В последнее время ЭЛТ заменяются цифровыми технологиями «плоских панелей», такими как *жидкокристаллические дисплеи* (ЖКД) и *плазменные* экраны. Хотя по конструктивному принципу они и отличаются кардинально от ЭЛТ, эти и другие технологии используют тот же самый принцип восприятия — все они для формирования одного цветового пикселя требуют наличия трех элементарных «подпикселей» (красного, зеленого и синего). ЖКД используют поляризационные фильтры, позволяющие задерживать или пропускать поляризованный свет через экран. В дисплеях, использующих активные матрицы, для адресации и передачи сигнала каждому из пикселей экрана используются тонкопленочные транзисторы. Для формирования трех первичных цветов в триадах каждого из пикселей используются цветные фильтры. В плазменных экранах пиксели представляют собой крошечные капсулы с газом, покрытые фосфоресцирующим веществом, излучающим один из трех первичных цветов. Каждая из капсул адресуется способом, аналогичным используемому в ЖКД. Эта возможность индивидуальной адресации триады каждого из пикселей является основой цифровых дисплеев.

Параметрами, обычно используемыми для различения цветов, являются *светлота*, *цветовой тон* и *насыщенность*. Как указывалось ранее в этом параграфе, светлота отображает ахроматическое ощущение интенсивности. Цветовой тон характеризует доминирующий цвет, воспринимаемый наблюдателем, причем большинство цветовых тонов в своем восприятии эквивалентны тому или иному спектральному цвету (см. рис. 6.5). Таким образом, когда мы называем некоторый объект красным, оранжевым или желтым, мы тем самым обозначаем его цветовой тон. Насыщенность цвета связана с его относительной белизной или с количеством белого цвета в нем. Спектрально чистые (монохроматические) цвета являются полностью насыщенными. Такие цвета, как розовый (смесь красного и белого) или бледно-лиловый (смесь фиолетового и белого), менее насыщены, причем величина насыщенности цвета обратно пропорциональна количеству белого цвета в смеси.

Цветовой тон и насыщенность вместе называются *цветностью*, и поэтому цвет может быть охарактеризован своей светлотой и цветностью. Величины красного, зеленого и синего, необходимые для получения некоторого конкретного цвета, называются *координатами цвета* и обозначаются соответственно X, Y и Z. Часто при описании цвета светлота не представляет интереса, и в таком случае цветовой тон и насыщенность можно выразить в *координатах цветности*, которые определяются как

$$x = \frac{X}{X+Y+Z}, \tag{6.1-1}$$

[2] Такая частота кадров используется в стандарте NTSC; в стандарте SECAM, распространенном в России, используется кадровая развертка с частотой 25 кадров (или 50 полукадров) в секунду. — *Прим. перев.*

$$y = \frac{Y}{X+Y+Z}, \tag{6.1-2}$$

$$z = \frac{Z}{X+Y+Z}. \tag{6.1-3}$$

Из приведенных выражений видно, что[3]

$$x + y + z = 1. \tag{6.1-4}$$

Для любой длины волны в диапазоне видимого спектра соответствующие координаты цвета могут быть найдены непосредственно при помощи кривых или таблиц, которые были составлены на основе обширного экспериментального материала [Poynton, 1996]. Отметим также более ранние работы [Walsh, 1958], [Kiver, 1965].

Другой способ задавать цвета основан на использовании *диаграммы цветностей* МКО (см. рис. 6.5), на которой вся совокупность цветов представлена как функция x (красной) и y (зеленой) координат цветности. Для любых значений координат x и y соответствующее значение z (синей) координаты цветности может быть получено из выражения (6.1-4): $z = 1 - (x + y)$. Например, точка, отмеченная на рис. 6.5 как зеленая, содержит приблизительно 62 % зеленого и 25 % красного. Из (6.1-4) следует, что содержание синего равно приблизительно 13 %.

Вдоль границы диаграммы цветностей, имеющей форму языка, расположены различные цвета спектра — от фиолетового с длиной волны 380 нм до красного с длиной волны 780 нм. Эти чистые (монохроматические) цвета показаны в спектре на рис. 6.2. Любая точка, расположенная не на границе, а внутри диаграммы, представляет некоторую смесь цветов. Точка равной энергии на рис. 6.5 соответствует равным долям трех первичных основных цветов; она представляет опорный белый цвет стандарта МКО. Любая точка, расположенная на границе диаграммы цветностей, имеет максимальную цветовую насыщенность. По мере того как точка смещается от границы к точке равной энергии, соответствующий ей цвет содержит в своем составе все бóльшую долю белого и становится все менее насыщенным. Цветовая насыщенность точки равной энергии равна нулю.

Диаграмма цветности полезна при рассмотрении процедуры смешения цветов, поскольку отрезок, соединяющий любые две точки диаграммы, определяет всевозможные различные цвета, которые могут быть получены при смешении двух данных цветов. Рассмотрим, например, отрезок, который соединяет точки, отмеченные на рис. 6.5 как красная и зеленая. Точки, представляющие смесь этих цветов, будут лежать на рассматриваемом отрезке. Если в смеси больше красного, чем зеленого, то новому цвету такой смеси будет соответствовать точка, находящаяся ближе к красной точке. Аналогично отрезок, проведенный от точки равной энергии к любой точке границы диаграммы, определяет все оттенки выбранного цвета.

Рассмотренная процедура непосредственно обобщается на случай смешения трех цветов. Для того чтобы определить диапазон цветов, которые могут

[3] Для обозначения координат цветности мы используем общепринятые обозначения x, y, z. Не следует путать их с обозначениями (x, y) для пространственных координат, которые используются в этой и других частях книги.

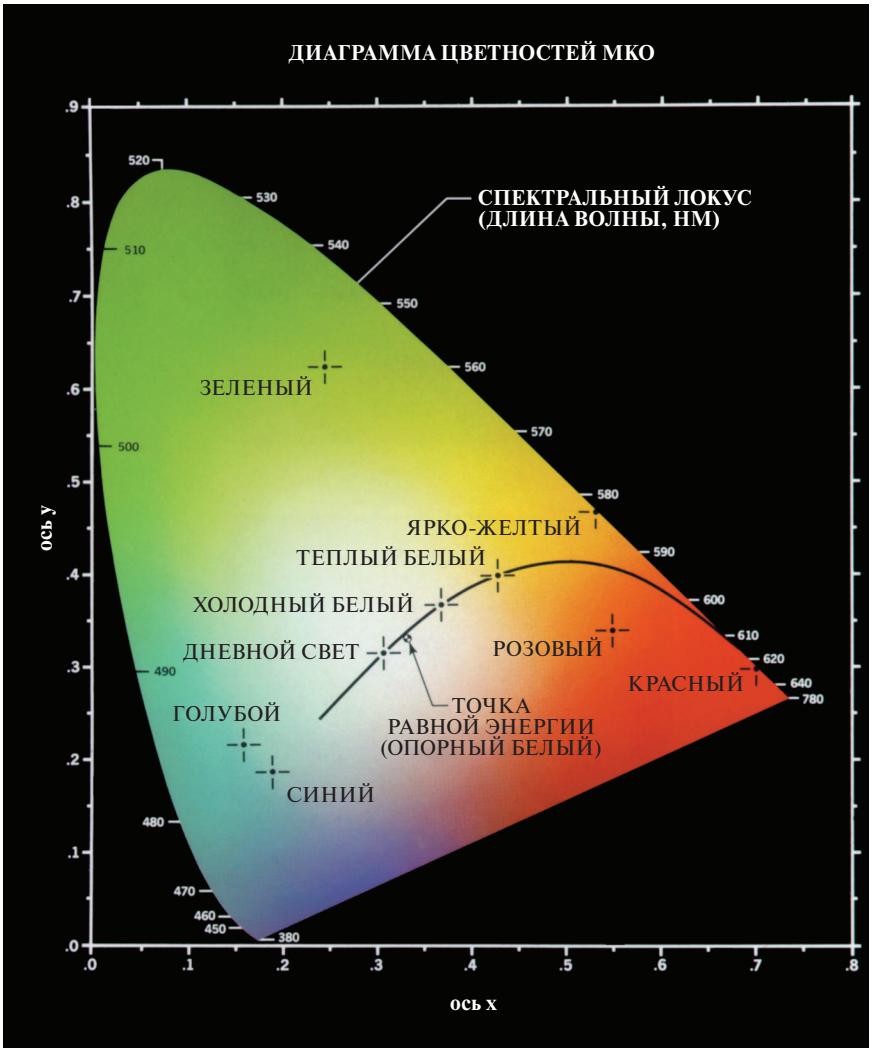


Рис. 6.5. Диаграмма цветностей. (Изображение предоставлено General Electric Co., Lamp Business Division)

быть получены комбинацией трех любых заданных цветов, нужно просто соединить между собой отрезками соответствующие точки на диаграмме цветности. В результате получится треугольник, и все цвета, соответствующие точкам внутри и на границе этого треугольника, могут быть получены как различные комбинации трех первоначальных цветов. Никакой треугольник с вершинами в трех точках с *неизменными* цветами не может включать весь диапазон цветов на рис. 6.5. Это геометрическое наблюдение подтверждает сделанное выше замечание о том, что не все цвета могут быть получены с помощью трех фиксированных первичных основных цветов.

Треугольник на рис. 6.6 представляет типичный для RGB-мониторов диапазон воспроизводимых цветов, называемый *цветовым охватом* (*цветовым га-*

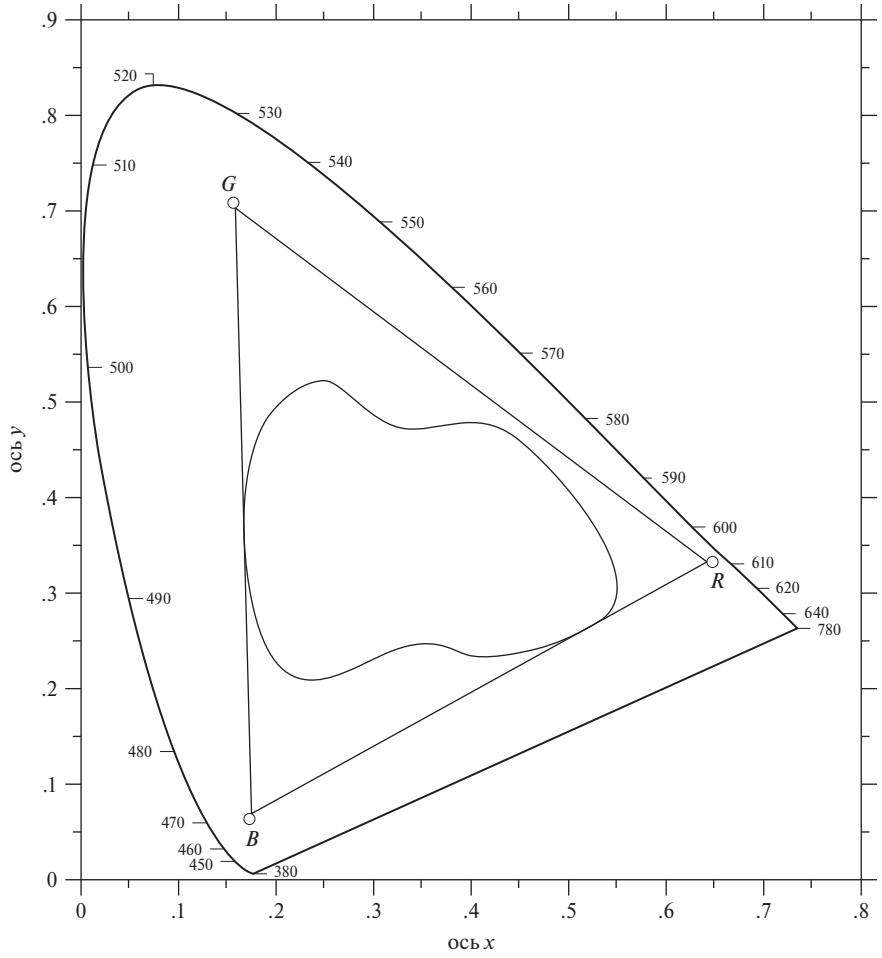


Рис. 6.6. Типичный цветовой охват цветного монитора (треугольная область) и цветного печатающего устройства (внутренняя область сложной формы)

мутом). Область сложной формы внутри этого треугольника представляет цветовой охват современных печатающих устройств высокого качества. Граница области охвата для печатающих устройств имеет сложную форму, потому что в процессе цветной печати одновременно используются аддитивные и субтрактивные процедуры смешения цветов. Управлять таким процессом намного труднее, чем процессом воспроизведения цветов на экране монитора, основанном на смешении трех очень хорошо контролируемых первичных основных цветов.

6.2. Цветовые модели

Назначение цветовой модели (называемой также *цветовым пространством* или *системой цветов*) состоит в том, чтобы сделать возможным описание цветов не-

которым стандартным, общепринятым образом. По существу, цветовая модель определяет некоторую систему координат и подпространство внутри этой системы, в котором каждый цвет представляется единственной точкой.

Большинство современных цветовых моделей ориентированы либо на устройства цветовоспроизведения (например цветные мониторы или принтеры), либо на определенные прикладные задачи (такие как создание цветной графики в анимации), когда работа с цветом является непосредственной целью. Аппаратно-ориентированными цветовыми моделями, наиболее часто используемыми на практике, являются модель RGB для цветных мониторов и широкого класса цветных видеокамер, модели CMY и CMYK для цветных принтеров и модель HSI[4], которая хорошо соответствует цветовосприятию человека. Последняя модель обладает также тем преимуществом, что она разделяет цветовую и яркостную (полутоновую) информацию на изображении и поэтому дает возможность применять многие из полутоновых методов обработки изображений, развитых в этой книге. В настоящее время используется множество различных цветовых моделей; это обусловлено тем, что наука о цвете представляет собой широкую область, включающую многочисленные приложения. Хотелось бы остановиться здесь на некоторых из этих моделей, поскольку они интересны и информативны. Однако, чтобы не выходить за рамки задач настоящей книги, мы ограничимся рассмотрением лишь вышеперечисленных моделей, которые играют ведущую роль в обработке изображений. Овладев материалом этой главы, читатель не будет испытывать трудностей в понимании других используемых цветовых моделей.

6.2.1. Цветовая модель RGB

В модели RGB каждый цвет представляется красным, зеленым и синим первичными основными цветами (компонентами). В основе модели лежит декартова система координат. Цветовое пространство представляет собой куб, показанный на рис. 6.7. В трех вершинах куба, лежащих на координатных осях, расположены точки, отвечающие первичным цветам, — красному, зеленому и синему. Вторичные цвета — голубой, пурпурный и желтый —расположены в трех других вершинах куба. Черный цвет находится в начале координат, а белый — в наиболее удаленной от начала координат вершине. В рассматриваемой модели оттенки серого цвета (точки с равными RGB-значениями) лежат на главной диагонали, соединяющей черную и белую вершины. Различные цвета в этой модели представляют собой точки на поверхности или внутри куба и определяются вектором, проведенным в данную точку из начала координат. Для удобства предполагается, что все значения цвета нормированы таким образом, чтобы куб на рис. 6.7 был единичным кубом, т.е. все значения R, G и B лежат в диапазоне [0,1].

---

[4] Названия цветовых моделей представляют собой английские аббревиатуры: RGB — *Red* (красный), *Green* (зеленый), *Blue* (синий); CMY — *Cyan* (голубой), *Magenta* (пурпурный), *Yellow* (желтый); CMYK — *Cyan* (голубой), *Magenta* (пурпурный), *Yellow* (желтый), *blacK* (черный); HSI — *Hue* (цветовой тон), *Saturation* (насыщенность), *Intensity* (интенсивность). Последнюю модель иногда называют также BHS-моделью — *Brightness* (яркость), *Hue* (цветовой тон), *Saturation* (насыщенность). — *Прим. перев.*

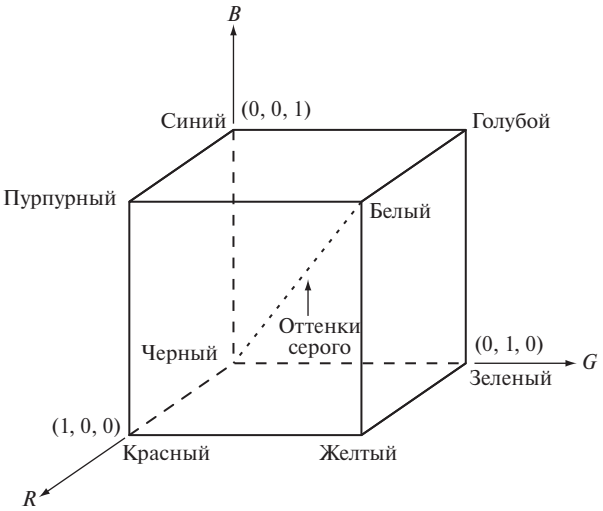


Рис. 6.7. Схематическое изображение цветового куба RGB. Точки на главной диагонали представляют оттенки серого цвета: от черного цвета в начале координат до белого цвета в точке (1,1,1)

Представляемые в цветовой модели RGB-изображения состоят из трех отдельных изображений-компонент, по одному для каждого первичного основного цвета. При воспроизведении RGB-монитором эти три изображения смешиваются на люминесцирующем экране и образуют составное цветное изображение, как это объяснялось в разделе 6.1. Число битов, используемых для представления каждого пикселя в RGB-пространстве, называется *глубиной цвета*. Рассмотрим RGB-изображение, в котором каждая из компонент — красная, зеленая и синяя — является 8-битовой. В таком случае говорят, что каждый *цветной* RGB-пиксель (т. е. триплет значений (R, G, B)) имеет глубину 24 бита (три цветовые плоскости умножить на число битов на каждую плоскость); для такого изображения часто используется термин *полноцве́тное* изображение. Суммарное число всевозможных цветов в 24-битовом RGB-изображении составляет $(2^8)^3 = 16777216$. На рис. 6.8 изображен 24-битовый цветовой куб RGB, соответствующий схеме на рис. 6.7.

Пример 6.1. Формирование скрытых граней и сечения цветового куба RGB.

■ На рис. 6.8 показан сплошной куб, который составлен из (28)3 = 16777216 указанных выше цветов. Удобным способом просмотра этих цветов является формирование цветных плоскостей (граней и сечений данного куба). Это достигается фиксированием одного из трех цветов и изменением двух оставшихся. Например, сечение, проходящее через центр куба и параллельное плоскости GB на рис. 6.8 и 6.7, представляет собой плоскость (127, G, B), G, B = 0, 1, 2, ..., 255. Здесь вместо нормализованных значений в диапазоне [0, 1], удобных с математической точки зрения, мы использовали реальные значения пикселей, поскольку именно последние используются в компьютере для формирования цветов. Рис. 6.9(*a*) показывает, что если три отдельные компоненты подать

на цветной монитор, то в результате получится цветное изображение сечения куба. Заметим, что компоненты являются полутоновыми (черно-белыми) изображениями, причем значение 0 соответствует черному, а 255 — белому. На рис. 6.9(*б*) представлены те грани цветового куба, которые не видны на рис. 6.8; изображения сформированы аналогичным образом.

Рис. 6.8. 24-битовый цветовой куб RGB

Интересно отметить, что процесс формирования (регистрации) цветного изображения является, по существу, обратным к показанному на рис. 6.9 процессу. Цветное изображение может быть получено с помощью трех фильтров: красного, зеленого и синего. Снимая цветную сцену на черно-белую фотокамеру, оснащенную одним из этих фильтров, мы получаем в результате полутоновое (монохромное) изображение, яркость которого пропорциональна интенсивности света, проходящего через фильтр. Повторение этой процедуры с каждым из фильтров дает три монохромных изображения, которые являются

а
б

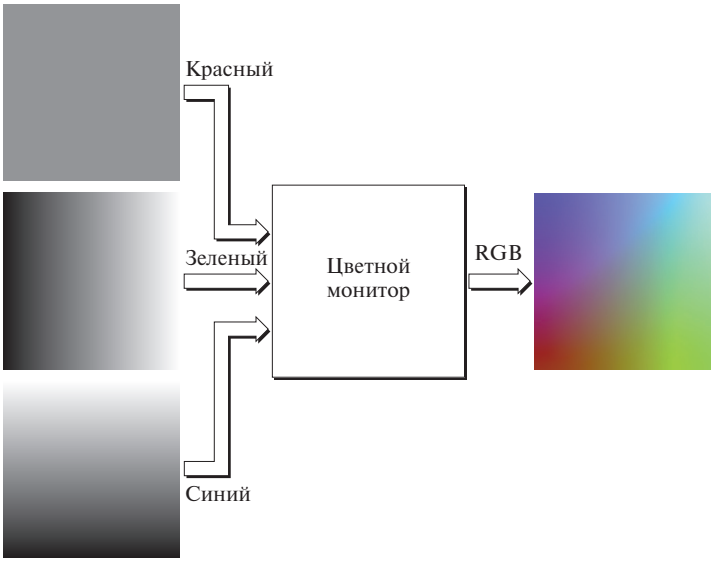


Рис. 6.9. (*а*) Формирование RGB-изображения сечения цветового куба плоскостью (127, G, B). (*б*) Три скрытые грани цветового куба на рис. 6.8

RGB-компонентами изображения цветной сцены. На практике использование цветных RGB-сенсоров дает возможность объединить весь процесс в одном устройстве. Показанная на рис. 6.9(*a*) процедура воспроизведения этих трех RGB-компонент даст цветное RGB-изображение исходной сцены. ■

Хотя высококачественные графические адаптеры и мониторы обеспечивают хорошее воспроизведение цветов 24-битовых RGB-изображений, многие используемые в настоящее время системы ограничены количеством цветов, равным 256. Кроме того, существует целый ряд приложений, в которых имеет смысл использовать не более сотни, а то и меньшее количество цветов. Хорошим примером здесь могут служить методы обработки изображений в псевдоцветах, которые обсуждаются в разделе 6.3. Также желательно иметь подмножество цветов, которые бы воспроизводились точно во всех используемых графических системах вне зависимости от их особенностей. Такое подмножество цветов называется палитрой *фиксированных RGB-цветов*[5] или набором цветов, одинаково воспроизводимых всеми системами. Применительно к интернет-приложениям это подмножество цветов называется палитрой *фиксированных Web-цветов* или набором цветов, одинаково воспроизводимых всеми программами просмотра интернет-сайтов.

Если исходить из предположения, что 256 цветов — это тот минимальный набор цветов, которые точно воспроизводятся любым графическим устройством, то полезно иметь общепринятый стандарт записи этих цветов. Известно, что сорок из этих 256 цветов воспроизводятся различными операционными системами по-разному; при этом остается 216 цветов, которые являются общими для большинства систем. Эти 216 цветов стали *de facto* стандартом фиксированных цветов, особенно для интернет-приложений. Они используются всегда, когда требуется, чтобы воспроизводимые цвета выглядели одинаково для большинства пользователей.

Обратим внимание, что $216 = (6)^3$, а значит, каждый из рассматриваемых 216 вариантов цвета можно формировать по-прежнему из трех RGB-компонент, но каждая из которых может принимать лишь 6 возможных значений: 0, 51, 102, 153, 204 или 255. Обычно указанные значения выражают в шестнадцатеричной системе счисления, как показано в табл. 6.1. Напомним, что шестнадцатеричные цифры 0, 1, 2,..., 9, A, B, C, D, E, F соответствуют десятичным числам 0, 1, 2,..., 9, 10, 11, 12, 13, 14, 15. Напомним также, что $(0)_{16} = (0000)_2$ и $(F)_{16} = (1111)_2$. Таким образом, например, $(FF)_{16} = (255)_{10} = (11111111)_2$ и видно, что двузначное шестнадцатеричное число соответствует восьмизначному двоичному, т. е. одному байту.

Таблица 6.1. Допустимые значения RGB-компонент в палитре фиксированных цветов

Система счисления	Значения RGB-компонент					
Шестнадцатеричная	00	33	66	99	CC	FF
Десятичная	0	51	102	153	204	255

[5] В оригинале — *safe RGB colors. — Прим. перев.*

Поскольку для формирования RGB-цвета требуется три числа, каждый цвет из палитры фиксированных цветов задается тремя двузначными шестнадцатеричными числами из табл. 6.1. Например, чистому красному цвету отвечает FF0000. Значения 000000 и FFFFFF отвечают черному и белому цветам соответственно. Те же результаты получаются, конечно, и при использовании более привычной десятичной системы счисления. Например, самый яркий красный цвет имеет компоненты $R = 255(FF)$ и $G = B = 0$.

На рис. 6.10(*а*) изображена палитра 216 фиксированных цветов, которые расположены в порядке убывания RGB-значений. Первый квадрат в верхнем левом массиве имеет значение FFFFFF (белый цвет), второй квадрат справа от него имеет значение FFFFCC, третий квадрат — FFFF99 и так далее для первой строки массива. Значения во второй строке того же массива равны FFCCFF, FFCCCC, FFCC99 и т.д. Последний квадрат этого массива имеет значение FF0000 (самый яркий красный цвет). Второй массив справа от только что рассмотренного начинается со значения CCFFFF, а его значения изменяются аналогично, как и значения оставшихся четырех массивов. Последний (нижний правый) квадрат последнего массива имеет значение 000000 (черный цвет). Важно отметить, что палитра из 216 фиксированных цветов включает далеко не все возможные оттенки 8-битового серого цвета. На рис. 6.10(*б*) показаны коды шестнадцати равноотстоящих оттенков серого цвета в RGB-модели. Некоторые из этих значений не принадлежат множеству фиксированных цветов, но большинство графических систем воспроизводит их ближайшим (не серым) цветом правильной

а
б

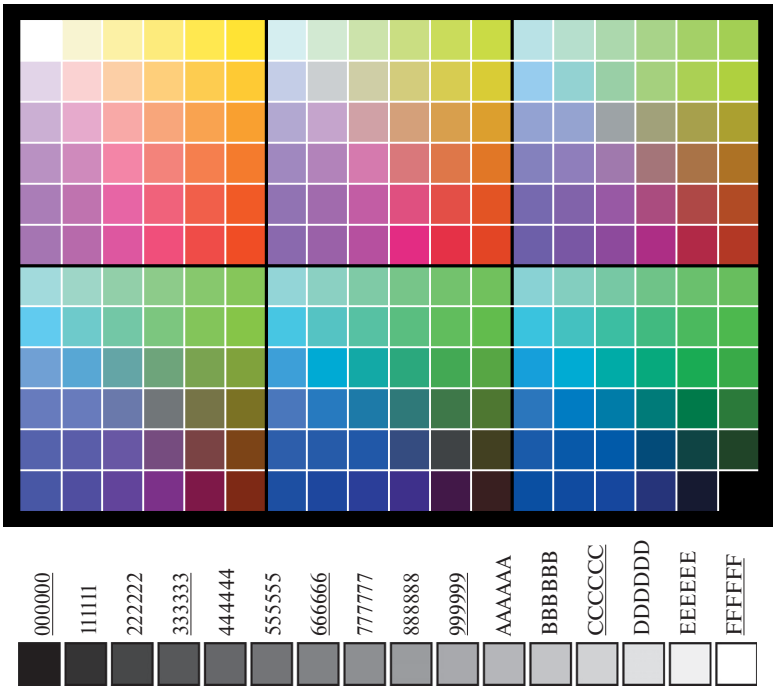


Рис. 6.10. (*а*) Палитра 216 фиксированных RGB-цветов. (*б*) 16 равноотстоящих оттенков серого цвета RGB-модели (серые цвета, входящие в палитру фиксированных цветов, выделены подчеркиванием)

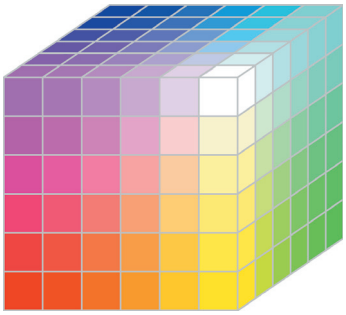


Рис. 6.11. RGB-куб фиксированных цветов

интенсивности. Серые цвета из палитры фиксированных цветов вида $(KKKKKKKK)_{16}$, $K = 0, 3, 6, 9, C, F$, выделены на рис. 6.10(*б*) подчеркиванием.

На рис. 6.11 представлен RGB-куб фиксированных цветов. Как видно из рис. 6.10(*а*), он состоит из 216 допустимых цветов (которые изображены маленькими кубиками), причем каждый слой содержит 36 цветов. Совмещая его с 24-битовым кубом, показанным на рис. 6.8, можно сказать, что каждый маленький кубик, имеющий единственное значение, соответствует множеству попадающих в него цветов 24-битового RGB-пространства.

6.2.2. Цветовые модели CMY и CMYK

Как указано в разделе 6.1, голубой, пурпурный и желтый цвета являются вторичными основными цветами световых источников или альтернативно первичными основными цветами красителей. Например, если поверхность, покрытая голубой краской, освещается белым светом, то красный свет от такой поверхности не отражается. Таким образом, голубой краситель вычитает красный свет из отражаемого белого, который сам по себе состоит из одинаковых количеств красного, зеленого и синего света.

Большинство устройств для нанесения цветных красителей на бумагу, такие как цветные принтеры и копировальные устройства, либо требуют представления входных данных в виде CMY, либо осуществляют преобразование данных из RGB в CMY. Это преобразование выполняется с помощью следующей простой операции

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}, \qquad (6.2\text{-}1)$$

где снова предполагается, что все значения цвета нормированы так, чтобы они находились в диапазоне [0, 1]. Выражение (6.2-1) показывает, что свет, отраженный от поверхности чисто голубого цвета, не содержит красного (поскольку в этом выражении $C = 1 - R$). Аналогично поверхность чисто пурпурного цвета не отражает зеленого, а поверхность чисто желтого цвета — синего. Кроме того, как видно из (6.2-1), набор значений RGB может быть легко получен из значений CMY вычитанием их из единицы. Как указывалось выше, цветовая модель CMY используется в процессе получения печатных копий, поэтому обратный переход от CMY к RGB, вообще говоря, не представляет большого практического интереса.

В соответствии с рис. 6.4 взятые в равном количестве первичные основные цвета красителей — голубой, пурпурный и желтый — должны давать черный цвет. На практике смешение этих цветов в процессе печати приводит к появлению черного цвета, который выглядит тусклым по сравнению с оригиналом. Поэтому для получения при печати чистого черного цвета (который зачастую

доминирует) цветовая модель CMY расширяется до модели CMYK, содержащей четвертый основной цвет — черный. Таким образом, когда полиграфисты говорят о четырехцветной печати, они имеют в виду три основных цвета модели CMY плюс черный цвет.

6.2.3. Цветовая модель HSI

Как мы видели, создание цветов в RGB- и CMY-моделях, а также переход из одной модели в другую являются простыми процедурами. Как отмечалось ранее, эти цветовые системы идеально приспособлены для аппаратной реализации. Кроме того, система RGB удачно согласована со зрительной системой человека в том смысле, что человеческий глаз восприимчив к красному, зеленому и синему первичным основным цветам. К сожалению, цветовые системы RGB, CMY и другие подобные плохо приспособлены для *описания* цветов таким образом, как это свойственно человеку. Например, описывая цвет автомобиля, человек не говорит о процентном содержании в нем каждого из основных цветов. Более того, рассматривая цветное изображение, мы не думаем о том, что оно составлено из трех отдельных изображений — по одному для каждого первичного основного цвета.

Глядя на окрашенный объект, человек описывает его с помощью цвета (цветового тона), насыщенности и светлоты. Напомним, что, как это обсуждалось в разделе 6.1, цветовой тон является характеристикой, которая описывает собственно цвет (чистый желтый, оранжевый, красный и т. д.), тогда как насыщенность дает меру того, в какой степени некоторый чистый цвет разбавлен белым. Светлота является субъективной характеристикой, которая практически не поддается измерению. Она соответствует понятию интенсивности (полутоновой яркости) в ахроматическом случае и является одним из ключевых параметров для описания цветового восприятия. Как известно, интенсивность (яркость) — основная характеристика монохромных (полутоновых) изображений. Эта величина может быть измерена и легко поддается интерпретации. В модели, которая носит название *цветовая модель HSI* (цветовой тон, насыщенность, интенсивность) и к рассмотрению которой мы приступаем, яркостная информация (интенсивность) отделена от цветовой информации (цветовой тон, насыщенность). В результате модель HSI представляет собой идеальное средство для построения алгоритмов обработки изображений, поскольку в основе модели лежит естественное и интуитивно близкое человеку описание цвета, ведь именно человек в конечном счете является и разработчиком, и пользователем этих алгоритмов. Можно подвести итог и сказать, что модель RGB идеальна для создания цветных изображений (как при их регистрации цветной камерой, так и при воспроизведении на экране монитора), но весьма ограничена в том, что касается *описания* цвета. Последующий материал предоставляет эффективный способ такого описания.

Как обсуждалось в примере 6.1, цветное RGB-изображение может рассматриваться как совокупность трех полутоновых изображений, яркость которых соответствует интенсивности красного, зеленого и синего цветов. Поэтому неудивительно, что существует возможность выделить интенсивность из цветного RGB-изображения. Это становится достаточно ясно, если взять

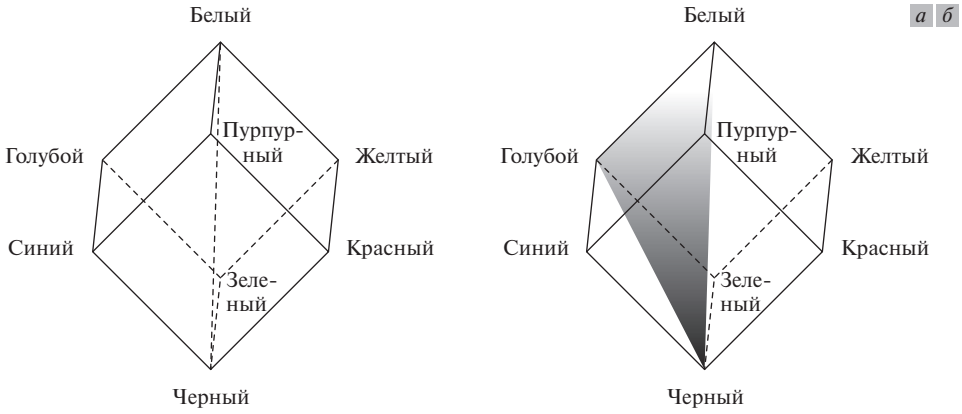


Рис. 6.12. Концептуальные взаимосвязи между цветовыми моделями RGB и HSI

изображенный на рис. 6.7 цветовой куб и расположить его так, чтобы «черная» вершина (0, 0, 0) находилась внизу, а «белая» вершина — прямо над ней, как показано на рис. 6.12(*а*). Как указывалось при обсуждении рис. 6.7, все оттенки серого цвета (которым отвечают определенные полутоновые яркости) расположены вдоль прямой (оси интенсивности), соединяющей черную и белую вершины. Для показанного на рис. 6.12 расположения эта ось является вертикальной. Таким образом, для того чтобы определить интенсивность (яркость) любой точки цветового куба на рис. 6.12, нужно провести плоскость *перпендикулярно* оси интенсивности так, чтобы она проходила через данную точку. Пересечение проведенной плоскости с осью интенсивности даст точку с искомым значением интенсивности (яркости) в диапазоне [0, 1]. Можно также заметить, что насыщенность цвета возрастает при увеличении расстояния точки от оси интенсивности. Действительно, очевидно, что цветовая насыщенность точек на оси интенсивности равна нулю, поскольку все эти точки являются серыми.

Для того чтобы понять, как определить цветовой тон для заданной RGB-точки, рассмотрим рис. 6.12(*б*). На этом рисунке изображена плоскость, которая определяется тремя точками (черной, белой и голубой). Поскольку плоскость содержит черную и белую точки, то она содержит и всю ось интенсивности. Более того, ясно, что *все* точки плоскости, которые принадлежат треугольнику, образованному осью интенсивности и отрезками пересечения плоскости с гранями куба (этот треугольник выделен на рис. 6.12(*б*) серым цветом), имеют *одинаковый* цветовой тон (в данном случае голубой). К такому выводу можно прийти, если вспомнить (см. раздел 6.1), что все цвета, порождаемые тремя любыми цветами, лежат в треугольнике, вершины которого суть точки цветового куба, отвечающие этим цветам. Если двумя точками из трех являются черная и белая точки, а третья точка отвечает некоторому цвету, то все точки треугольника будут иметь одинаковый цветовой тон, поскольку черная и белая составляющие не могут его изменить. (Разумеется, интенсивность и насыщенность точек в треугольнике будут различаться.) Поворачивая рассматриваемую плоскость вокруг вертикальной оси интенсивности, мы будем получать различные цве-

товые тона. Проведенное концептуальное рассмотрение приводит нас к выводу о том, что необходимые для построения пространства HSI значения цветового тона, насыщенности и интенсивности могут быть получены из цветовых координат модели RGB. Таким образом, если придать приведенным выше рассуждениям вид точных геометрических формул, мы получим возможность преобразовать RGB-пространство в HSI-пространство.

При выбранном на рис. 6.12 расположении RGB-куба пространство HSI представляется множеством цветовых плоскостей, *перпендикулярных* вертикальной оси интенсивности. В зависимости от положения точки пересечения этих плоскостей с осью интенсивности сечение куба плоскостью может иметь треугольную или шестиугольную форму. Это легче представить себе, если смотреть на куб сверху вниз вдоль оси интенсивности, как показано на рис. 6.13(*а*). В представленной плоскости сечения мы видим, что первичные основные цвета расположены под углом 120° друг относительно друга. Вторичные основные цвета находятся под углом 60° относительно первичных цветов. Это означает, что вторичные основные цвета также расположены под углом 120° друг относительно друга. На рис. 6.13(*б*) представлено сечение той же шестиугольной формы вместе с произвольной цветовой точкой (показана на рисунке черной). Цветовой тон этой точки определяется углом между направлениями из центра шестиугольника на данную точку и на некоторую опорную точку. В качестве опорной обычно (хотя и не всегда) выбирается точка пересечения плоскости с осью, соответствующей красному цвету. Значение угла отсчитывается против часовой стрелки. Насыщенность задается длиной вектора, проведенного из центра в данную точку (т. е. расстоянием от вертикальной оси). Отметим,

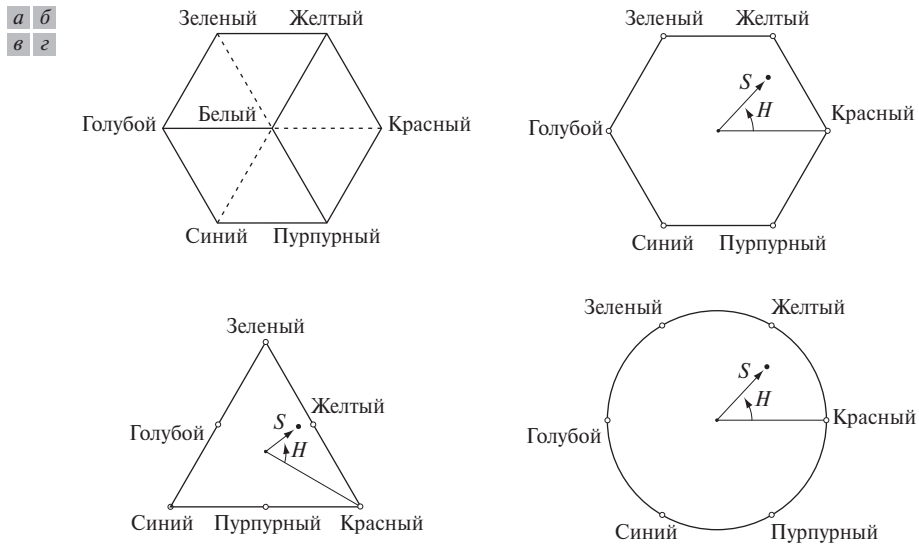


Рис. 6.13. Цветовой тон и насыщенность в цветовой модели HSI. Конец вектора на рисунках соответствует произвольной точке цветового пространства. Угол, отсчитываемый от красной оси, определяет цветовой тон, а длина вектора — насыщенность. Интенсивность всех цветов в любой из изображенных плоскостей задается положением этой плоскости относительно вертикальной оси интенсивности

что центр является точкой пересечения данной цветовой плоскости и вертикальной оси интенсивности. Таким образом, пространство HSI составляется из цветовых областей в плоскостях, перпендикулярных вертикальной оси интенсивности. Положение точки в цветовой области характеризуется длиной вектора, проведенного в эту точку из центра соответствующего сечения,

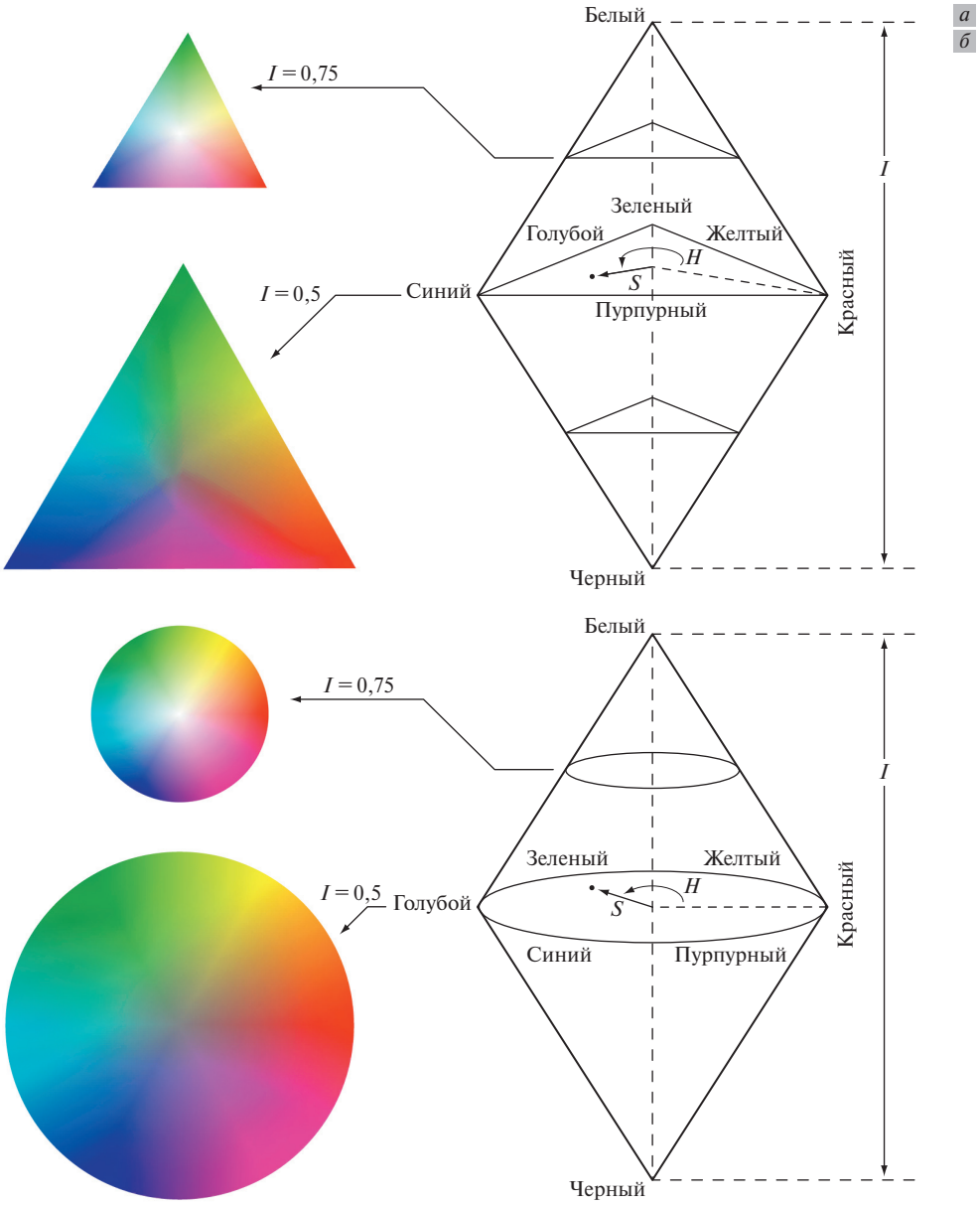


Рис. 6.14. Цветовая модель HSI, в основе которой лежат (*а*) цветовые треугольники и (*б*) цветовые круги. Треугольники и круги перпендикулярны вертикальной оси интенсивности

и углом, образуемым этим вектором с красной осью. В принципе, при задании цветовой области можно использовать шестиугольник, как это было сделано выше, треугольник или даже круг, как это показано на рис. 6.13(*в*) и (*г*). Конкретный выбор формы области в действительности не имеет значения, поскольку любая область указанного вида может быть отображена на любую из двух других при помощи геометрического преобразования. На рис. 6.14 представлены две HSI-модели, в основе одной из которых лежат цветовые треугольники, а другой — круги.

Преобразование цветов из системы RGB в систему HSI

Преобразования из RGB в HSI и обратно выполняются поэлементно. Поэтому для простоты выражений координатные параметры (x, y) опущены.

Цветовой тон H для каждого пикселя, заданного в RGB-формате изображения, определяется по формуле

$$H = \begin{cases} \theta & \text{при } B \le G, \\ 360° - \theta & \text{при } B > G, \end{cases} \tag{6.2-2}$$

где[6]

$$\theta = \arccos\left\{ \frac{\frac{1}{2}[(R-G)+(R-B)]}{\left[(R-G)^2+(R-B)(G-B)\right]^{1/2}} \right\}.$$

Насыщенность S дается выражением

$$S = 1 - \frac{3}{(R+G+B)}[\min(R,G,B)]. \tag{6.2-3}$$

Наконец, интенсивность I дается выражением

$$I = \frac{1}{3}(R+G+B). \tag{6.2-4}$$

Формулы написаны в предположении, что RGB-координаты нормированы так, чтобы их значения лежали в диапазоне [0, 1], и угол θ отсчитывается от красной оси пространства HSI, как показано на рис. 6.13. Цветовой тон может быть нормирован таким образом, чтобы его значения попадали в диапазон [0, 1]. Для этого нужно значения H, получаемые по формуле (6.2-2), поделить на 360°. Значения двух других HSI-компонент автоматически попадают в этот диапазон при условии, что RGB-значения лежат в интервале [0, 1].

Формулы (6.2-2)—(6.2-4) могут быть выведены из геометрических соображений на основе рис. 6.12 и 6.13. Этот вывод является утомительным и не добавляет ничего существенного к проведенному рассмотрению. Интересующийся

[6] При вычислении данного выражения рекомендуется прибавлять малую добавку к значению знаменателя, чтобы избежать деления на 0, когда $R = G = B$; при этом $\theta = 90°$. Заметим также, что в этом случае уравнение (6.2.3) дает в результате 0. Добавим также, что при обратном преобразовании из HSI в RGB в результате вычисления уравнений (6.2.5)—(6.2.7) получим $R = G = B = I$, как и следует ожидать, поскольку случай $R = G = B$ означает, что элемент является серым.

читатель может найти доказательство этих формул, также как и нижеследующих формул обратного преобразования из HSI в RGB, по приведенным литературным ссылкам или на веб-сайте книги.

Преобразование цветов из системы HSI в систему RGB

Теперь по заданным значениям HSI в интервале [0, 1] мы хотим найти соответствующие RGB-значения в том же интервале. Для этого, в зависимости от значения H, необходимо использовать различные формулы. В области изменения цветового тона существуют три различных сектора величиной в 120°, которые разделены направлениями на первичные основные цвета (см. рис. 6.13). Прежде всего мы умножаем величину H на 360°, чтобы восстановить исходный диапазон изменений цветового тона [0, 360°].

RG-сектор ($0° \le H < 120°$). Если значение H находится в этом секторе, то RGB-координаты определяются по формулам

$$B = I(1-S), \tag{6.2-5}$$

$$R = I\left[1+\frac{S\cos H}{\cos(60°-H)}\right] \tag{6.2-6}$$

и

$$G = 3I-(R+B). \tag{6.2-7}$$

GB-сектор ($120° \le H < 240°$). Если значение H находится в этом секторе, то мы сначала вычитаем из него 120°:

$$H = H-120°. \tag{6.2-8}$$

После этого RGB-координаты определяются по формулам

$$R = I(1-S), \tag{6.2-9}$$

$$G = I\left[1+\frac{S\cos H}{\cos(60^{\circ}-H)}\right] \tag{6.2-10}$$

и

$$B = 3I-(R+G). \tag{6.2-11}$$

BR-сектор ($240° \le H \le 360°$). Наконец, если значение H находится в этом секторе, то мы вычитаем из него 240°:

$$H = H-240°. \tag{6.2-12}$$

Затем RGB-координаты определяются по формулам

$$G = I(1-S), \tag{6.2-13}$$

$$B = I\left[1+\frac{S\cos H}{\cos(60°-H)}\right] \tag{6.2-14}$$

и

$$R = 3I-(G+B). \tag{6.2-15}$$

Использование этих формул для обработки изображений обсуждается в ряде последующих разделов.

а б в

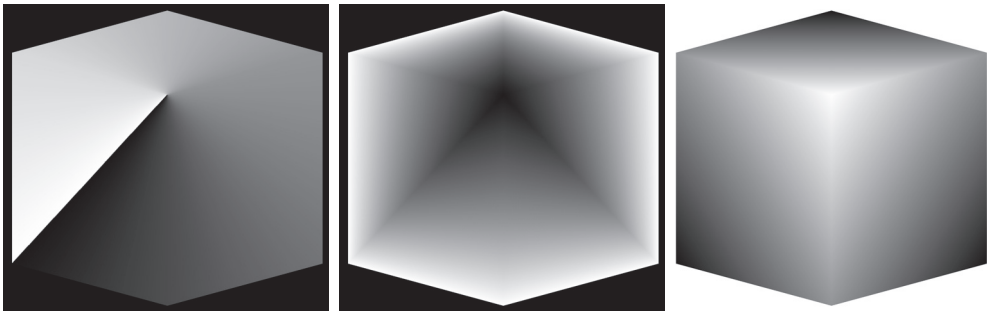


Рис. 6.15. HSI-компоненты изображения цветового куба на рис. 6.8: (*а*) цветовой тон, (*б*) насыщенность и (*в*) интенсивность

Пример 6.2. Значения HSI, соответствующие изображению цветового куба RGB.

■ На рис. 6.15 представлены изображения цветового тона, насыщенности и интенсивности, соответствующие изображению цветового куба RGB на рис. 6.8. Изображение на рис. 6.15(*а*) является изображением цветового тона. Наиболее примечательная особенность этого изображения — это разрыв значений вдоль диагонали передней (красной) грани куба. Для того чтобы понять причину этого разрыва, обратимся к рис. 6.8, нарисуем диагональ, соединяющую красную и белую вершины куба, и выберем точку на середине этой диагонали. Рассмотрим замкнутый путь на поверхности куба, который начинается из этой точки, проходит последовательно через желтую, зеленую, голубую, синюю и пурпурную вершины и возвращается обратно в исходную красную точку. В соответствии с рис. 6.13 значения цветового тона вдоль этого пути должны возрастать от 0° до 360° (т. е. от наибольшего до наименьшего возможного значения цветового тона). Именно это и демонстрирует рис. 6.15(*а*), поскольку на этом полутоновом рисунке наименьшему значению соответствует черный цвет, а наибольшему — белый. В действительности изначально цветовой тон принимал значения в диапазоне [0, 1], но для воспроизведения изображение цветового тона было подвергнуто масштабному преобразованию так, чтобы его значения оказались в диапазоне [0, 255].

Представленное на рис. 6.15(*б*) изображение насыщенности демонстрирует постепенное уменьшение значений в направлении белой вершины RGB-куба. Это показывает, что по мере приближения к белому цвет становится все менее и менее насыщенным. Наконец, значение каждой точки изображения интенсивности, которое представлено на рис. 6.15(*в*), есть среднее значений RGB-компонент в соответствующей точке изображения куба на рис. 6.8. ■

Работа с HSI-изображениями

Остановимся теперь кратко на некоторых простых методах работы с изображениями, которые заданы своими HSI-компонентами. Это позволит лучше познакомиться с этими компонентами и будет способствовать более глубокому пониманию цветовой модели HSI. На рис. 6.16(*а*) представлено изображение, составленное из первичных и вторичных основных RGB-цветов. На рис. 6.16(*б*)—

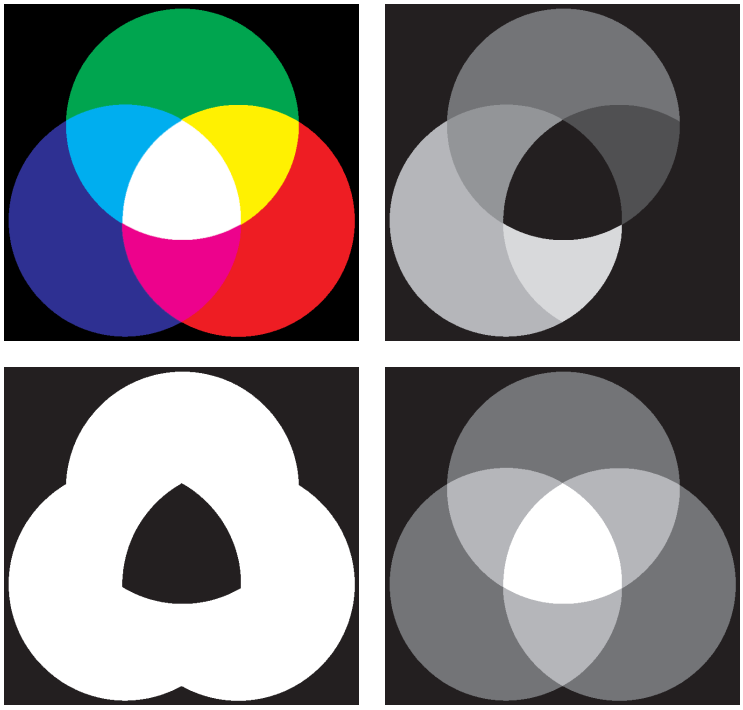


Рис. 6.16. (*а*) RGB-изображение и компоненты соответствующего HSI-изображения: (*б*) цветовой тон, (*в*) насыщенность и (*г*) интенсивность

(*г*) представлены *H*-, *S*- и *I*-компоненты этого изображения. Эти изображения были получены с использованием формул (6.2-2)—(6.2-4). Напомним, что, согласно проведенному выше в этом параграфе рассмотрению, значение яркости на рис. 6.16(*б*) соответствует углу. Например, поскольку красный цвет соответствует углу 0°, то области красного цвета на рис. 6.16(*а*) отвечает черная область на изображении цветового тона. Аналогично яркость на рис. 6.16(*в*) соответствует насыщенности (значения которой для воспроизведения растянуты на диапазон [0, 255]), а значение яркости на рис. 6.16(*г*) равно интенсивности.

Чтобы изменить отдельный цвет некоторой области RGB-изображения, мы изменяем значения в соответствующей области изображения цветового тона на рис. 6.16(*б*). Затем мы преобразуем новые значения цветового тона *H* и оставленные без изменений значения насыщенности *S* и интенсивности *I* обратно в RGB-систему так, как это было объяснено при рассмотрении формул (6.2-5)—(6.2-15). Чтобы изменить насыщенность (чистоту цвета) в некоторой области, мы действуем подобным же образом, за исключением того, что теперь мы изменяем значения в компоненте насыщенности HSI-пространства. Аналогично можно изменить и среднюю интенсивность любой области. Разумеется, эти изменения могут быть осуществлены одновременно. Например, изображение на рис. 6.17(*а*) было получено из изображения на рис. 6.16(*б*) путем изменения значений в точках, соответствующих синей и зеленой областям; новые значения в этих точках полагались равными нулю. На рис. 6.17(*б*) насыщенность голубой области была уменьшена вдвое по сравнению с первоначальным значением на-

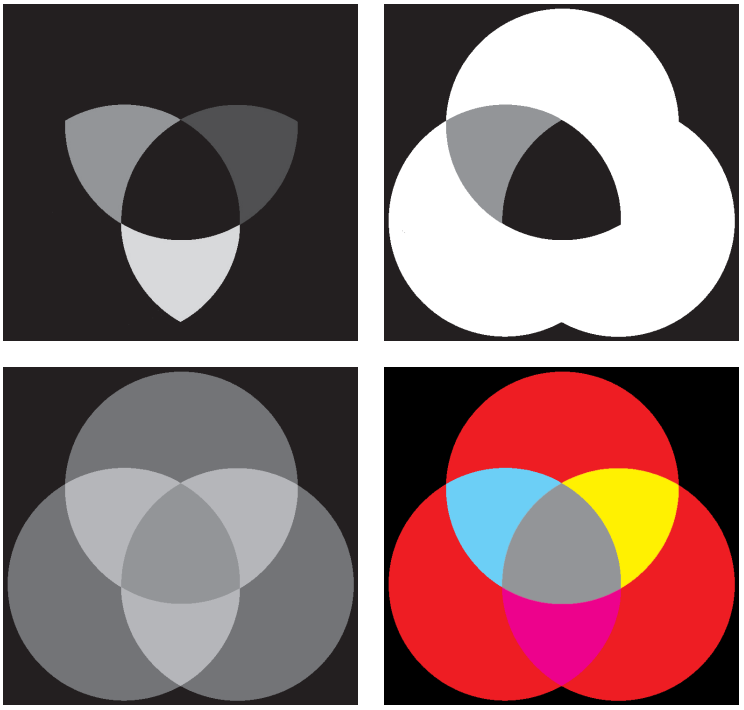


Рис. 6.17. (*а*)—(*в*) Изображения модифицированных HSI-компонент. (*г*) Результирующее RGB-изображение. (Исходные изображения HSI-компонент представлены на рис. 6.16)

сыщенности на рис. 6.16(*в*). На рис. 6.17(*в*) интенсивность белой области в центре была уменьшена вдвое по сравнению с интенсивностью на рис. 6.16(*г*). Результат обратного преобразования в систему RGB модифицированного таким образом HSI-изображения представлен на рис. 6.17(*г*). Как и следовало ожидать, внешние части всех кругов имеют теперь красный цвет, насыщенность цвета голубой области уменьшилась, а центральная область из белой превратилась в серую. Хотя полученные результаты элементарны, они демонстрируют те возможности, которые предоставляет HSI-модель в части *независимого* контроля над цветовым тоном, насыщенностью и интенсивностью — привычными характеристиками, используемыми при описании цвета.

6.3. Обработка изображений в псевдоцветах

Обработка изображения в *псевдоцветах* (называемых также *условными цветами*) подразумевает присвоение цветов пикселям полутонового изображения на основе некоторого определенного правила. Термин «псевдоцвета», или «условные цвета», используется для того, чтобы отличать цветные изображения, полученные в результате присвоения цветов точкам монохромного изображения, от изображений в натуральных цветах, задача обработки которых будет обсуждаться начиная с раздела 6.4. Основное применение псевдоцветов — это

визуализация и интерпретация человеком той информации, которая содержится в полутоновых изображениях или видеопоследовательностях. Как уже отмечалось в начале настоящей главы, основным стимулом к использованию псевдоцветов является то обстоятельство, что человек способен различать тысячи оттенков цвета и только около двух десятков оттенков серого.

6.3.1. Квантование по яркости

Метод *квантования по яркости* (иногда называемый *квантованием по оптической плотности*) и цветового кодирования (т. е. присвоения пикселям тех или иных цветов в качестве уровней квантования) является одним из простейших примеров обработки изображения в псевдоцветах. Если рассматривать изображение как поверхность в трехмерном пространстве (см. рис. 2.18(*а*)), то обсуждаемый метод можно представлять себе как метод, основанный на проведении плоскостей, параллельных координатной плоскости изображения. Каждая такая плоскость «разрезает» поверхность по области пересечения. На рис. 6.18 представлен пример, в котором секущая плоскость $f(x, y) = l_i$ используется для разделения образованной изображением поверхности на два уровня.

Присвоим каждой из сторон показанной на рис. 6.18 плоскости свой цвет и будем кодировать пиксели, которым отвечают точки поверхности, лежащие выше плоскости, одним цветом, а пиксели, которым отвечают точки поверхности, лежащие ниже плоскости, — другим. Точкам, которые соответствуют пересечению плоскости с поверхностью, можно присвоить любой из двух выбранных цветов. В результате мы получим двухцветное изображение, видом которого можно управлять, двигая секущую плоскость вверх и вниз вдоль оси яркости.

В общем случае рассматриваемый метод может быть сформулирован следующим образом. Пусть яркость принимает значения в диапазоне $[0, L - 1]$, уровень l_0 соответствует черному ($f(x, y) = 0$), а уровень l_{L-1} — белому ($f(x, y) = L - 1$). Предположим, что P плоскостям, перпендикулярным оси яркости, соответствуют уровни $l_1, l_2, ..., l_P$, причем $0 < P < L - 1$. Таким образом, P плоскостей

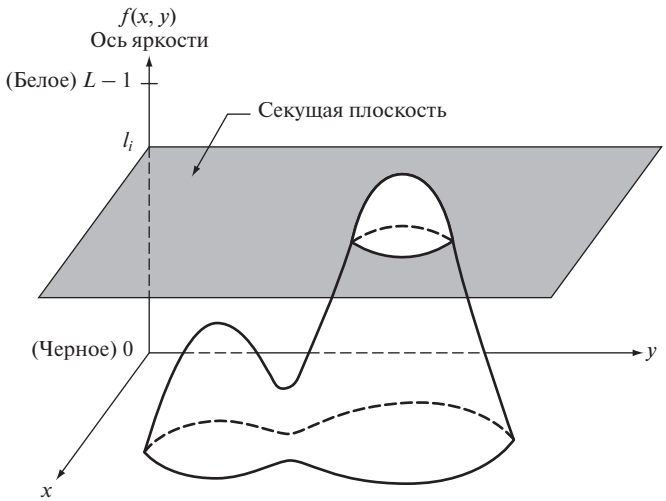


Рис. 6.18. Геометрическое объяснение метода квантования по яркости

Рис. 6.19. Альтернативный способ определения метода квантования по яркости

делят весь диапазон яркости на $P + 1$ интервал V_1, V_2, ..., V_{P+2}. Сопоставление цвета значению яркости осуществляется по правилу

$$f(x,y) = c_k, \quad \text{если } f(x,y) \in V_k, \tag{6.3-1}$$

где c_k — цвет, соответствующий k-му интервалу яркости V_k, который определяется секущими плоскостями при $l = k - 1$ и $l = k$.

Концепция секущих плоскостей полезна, главным образом, для геометрической интерпретации метода квантования по яркости. Рис. 6.19 иллюстрирует альтернативный способ определения того же преобразования, которое ранее было определено с помощью рис. 6.18. Преобразование входного значения яркости в один из двух цветов, в зависимости от того, превышает это значение данную величину l_i или нет, осуществляется при помощи представленной на рис. 6.19 функции преобразования. Если используется большее число уровней, то функция преобразования имеет ступенчатую форму.

а *б*

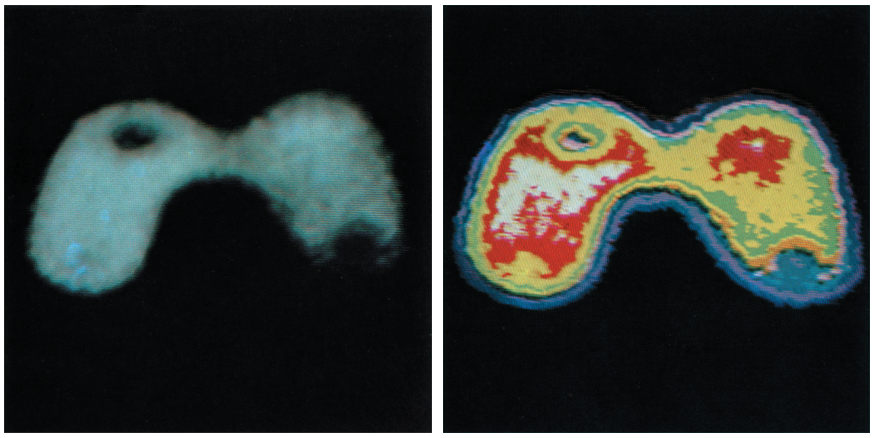


Рис. 6.20. (*а*) Монохромное изображение фантома щитовидной железы. (*б*) Результат квантования по яркости (по плотности) на восемь цветов. (Исходное изображение предоставил др. Дж.Л. Бланкеншип, Instrumentation and Controls Division, Oak Ridge National Laboratory)

Пример 6.3. Квантование по яркости.

■ Простой практический пример использования квантования по яркости представлен на рис. 6.20. Изображение на рис. 6.20(*а*) представляет собой монохромное изображение фантома щитовидной железы (радиационного тестового образца), а изображение на рис. 6.20(*б*) — результат квантования по яркости этого изображения на восемь псевдоцветов. Области, которые на монохромном изображении выглядят как области постоянной яркости, в действительности таковыми не являются, что показывают различия в цвете на квантованном изображении. На монохромном изображении левая доля, например, имеет тусклый серый цвет, и заметить изменения яркости в этой области изображения достаточно трудно. На цветном изображении, напротив, четко выделяются восемь различных областей постоянной яркости, по одной для каждого использованного цвета. ■

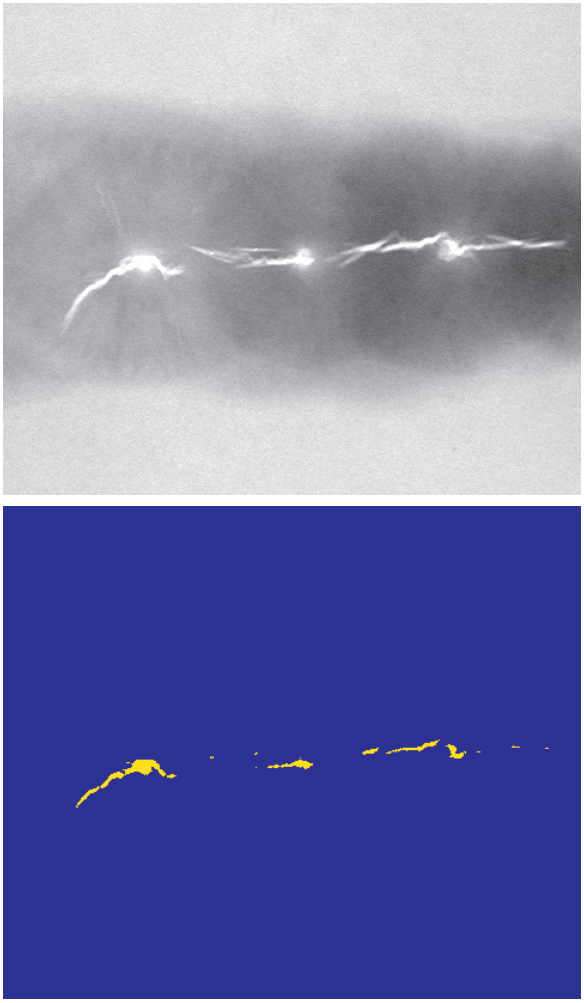


Рис. 6.21. Монохромное рентгеновское изображение сварного шва. (*б*) Результат цветового кодирования. (Исходное изображение предоставлено компанией X-TEK Systems, Ltd.)

В предыдущем простом примере яркость была разделена на диапазоны, которым были поставлены в соответствие различные цвета без учета того, какую информацию несут значения яркости на конкретном изображении. В этом случае интерес представляло лишь выделение областей с различными уровнями яркости, которые составляли изображение. Квантование по яркости играет гораздо более значимую и полезную роль в том случае, если выбор диапазонов яркости на изображении основан на физических характеристиках изображаемых объектов. Так, например, на рис. 6.21(*а*) представлено рентгеновское изображение сварного шва (горизонтальная темная область), который содержит несколько трещин и каверн (яркие белые прожилки, проходящие горизонтально по центру изображения). Известно, что если сварной шов содержит каверну или трещину, то полная интенсивность потока рентгеновских лучей, проходящих через объект, приводит к насыщению детектора, находящегося по другую сторону от объекта. Таким образом, значения, равные 255 на выходе такой системы, использующей 8-битовые изображения, автоматически указывают на дефект сварного шва. Если контроль качества сварного шва осуществляется человеком на основе просмотра изображения (такая процедура контроля широко распространена по сей день), то простое цветовое кодирование, которое присваивает один цвет значению 255 и другой цвет всем остальным значениям, может значительно облегчить работу эксперта. На рис. 6.21(*б*) показан результат такого кодирования. Без дополнительных объяснений ясно, что вероятность экспертной ошибки будет меньше в том случае, если изображение представлено в таком виде, как на рис. 6.21(*б*), а не на рис. 6.21(*а*). Другими словами, когда точно известны значения или диапазон значений яркости, которые представляют интерес, квантование по яркости является простым, но мощным средством визуализации, особенно если речь идет о большом числе изображений. Следующий пример значительно более сложный.

Пример 6.4. Использование цвета для изображения количества осадков.

■ Измерение количества осадков, особенно в тропических областях Земли, представляет интерес в разнообразных приложениях, связанных с изучением окружающей среды. Получение точных данных с использованием приборов наземного базирования является непростым и дорогостоящим делом, а получение полной картины осадков еще более затруднено, поскольку значительное количество осадков выпадает над океанами. Один из возможных способов получения полной картины осадков связан с использованием спутников. Спутник TRMM (Tropical Rainfall Measuring Mission — Программа измерения количества осадков в тропических областях) в числе прочего оснащен тремя устройствами, разработанными для слежения за осадками: специальным радаром, микроволновым сканером и сканером видимого и инфракрасного диапазонов. (По поводу регистрации изображений различной природы см. разделы 1.3 и 2.3.)

Результатом обработки данных со спутника являются оценки среднего количества осадков в исследуемой области за определенный период времени. Имея такие оценки, нетрудно построить полутоновое изображение, элементы которого соответствуют областям земной поверхности (размер этих областей определяется разрешением используемых устройств регистрации данных), а значения

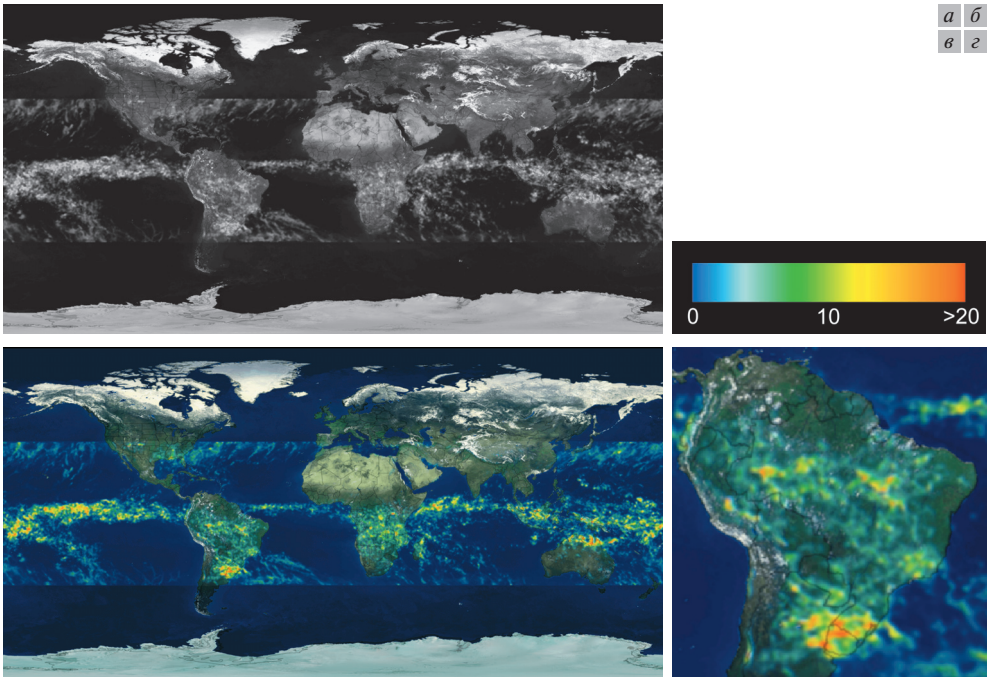


Рис. 6.22. (*а*) Полутоновое изображение, яркость которого (в выделенной чуть более светлой полосе) соответствует среднемесячному количеству осадков. (*б*) Шкала соответствия цветов и значений яркости. (*в*) Изображение, полученное в результате цветового кодирования. (*г*) Увеличенное изображение части Южной Америки. (Изображения предоставлены NASA)

элементов соответствуют количеству осадков. Такое изображение представлено на рис. 6.22(*а*), причем обследуемые спутником области находятся в чуть более светлой горизонтальной полосе, занимающей среднюю треть картинки (тропические области). В данном конкретном примере представлено среднемесячное количество осадков (в дюймах) за трехлетний период.

Визуальный анализ этой картины осадков представляет весьма трудную, если не сказать невыполнимую задачу. Предположим, однако, что мы кодируем уровни яркости от 0 до 255 с помощью цветов, как показано на рис. 6.22(*б*). Синие оттенки цвета означают малое количество осадков, а красные — большое. Заметим, что шкала оканчивается чистым красным цветом, который отвечает количеству осадков, превышающему 20 дюймов. На рис. 6.22(*в*) показан результат цветового кодирования исходного полутонового изображения с использованием приведенной цветовой шкалы. Как показывает это изображение, а также увеличенное изображение области на рис. 6.22(*г*), представление результатов таким способом существенно облегчает их интерпретацию. Помимо того, что полученные с помощью спутников результаты обеспечивают построение полной карты осадков земной поверхности, они также позволяют метеорологам с большей, чем ранее, точностью калибровать устройства слежения за осадками наземного базирования. ■

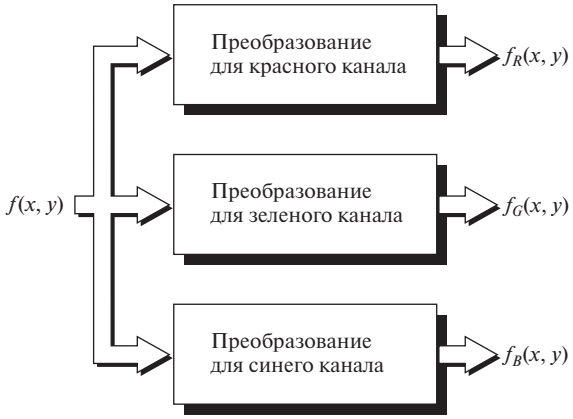


Рис. 6.23. Функциональная блок-схема формирования изображения в псевдоцветах. Величины f_R, f_G и f_B подаются в качестве входных сигналов соответственно в красный, зеленый и синий каналы цветного RGB-монитора

6.3.2. Преобразование яркости в цвет

Существуют другие преобразования более общего вида, применение которых позволяет достичь более разнообразных результатов по сравнению с простым методом квантования по яркости, рассмотренным в предыдущем разделе. Один из таких методов, привлекательный с практической точки зрения, представлен на рис. 6.23. Основная идея, лежащая в основе этого метода, состоит в том, чтобы осуществить три независимых преобразования значений яркости для каждого пикселя входного изображения. Затем три полученных изображения подаются в красный, зеленый и синий каналы цветного монитора. В результате формируется составное изображение, цветовое содержание которого определяется природой используемых функций преобразования. Отметим, что эти преобразования затрагивают лишь значения яркости изображения и не зависят от положения точки на изображении.

Частным случаем только что описанной методики является метод, рассмотренный в предыдущем параграфе. В нем для формирования цветов используются кусочно-постоянные функции яркости (см. рис. 6.19). Метод, обсуждаемый в настоящем параграфе, может использовать гладкие и нелинейные функции, что, естественно, придает всей методике значительную гибкость.

Пример 6.5. Использование псевдоцветов для обнаружения взрывчатых веществ в багаже.

■ На рис. 6.24(*a*) представлены два монохромных изображения, полученные с помощью используемой в аэропортах рентгеновской сканирующей системы контроля багажа. Багаж на изображении слева содержит обычные предметы. Багаж на изображении справа содержит те же предметы, а также некоторый блок, имитирующий пластиковую взрывчатку. Цель данного примера состоит в том, чтобы проиллюстрировать использование метода преобразования яркости в цвет для реализации различных вариантов контроля.

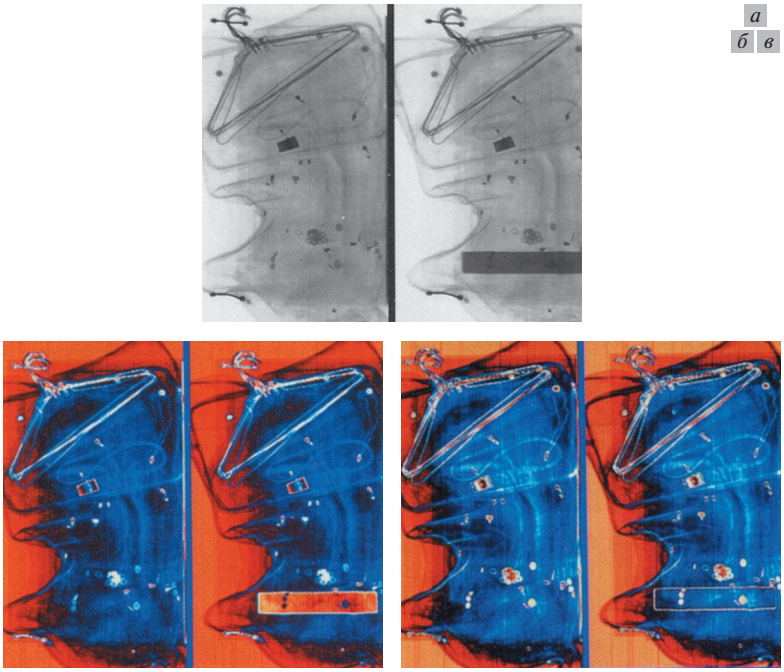


Рис. 6.24. Обработка в псевдоцветах с использованием представленных на рис. 6.25 преобразований яркости в цвет. (Исходное изображение предоставил д-р. М. Гурвиц, Вестингауз)

На рис. 6.25 изображены используемые функции преобразований. Эти синусоидальные функции содержат области вблизи максимумов, в которых их значения относительно постоянны, а также области вблизи минимумов, где их значения быстро изменяются. Изменяя фазу и частоту каждой из синусоидальных функций, можно добиться выделения (цветом) некоторых диапазонов значений полутоновой яркости. Например, если фазы и частоты всех синусоид совпадают, то изображение на выходе также будет монохромным. Небольшие различия в фазах между тремя функциями преобразования приводят к незначительным изменениям для тех пикселей, яркость которых соответствует положениям максимумов синусоид, особенно если синусоиды имеют широкие профили (низкие частоты). Пиксели, яркость которых соответствует крутым участкам синусоид, приобретают намного более ярко выраженную цветовую окраску. Причина этого в том, что в результате сдвига фаз между синусоидами их амплитуды в этих областях заметно различаются.

Представленное на рис. 6.24(*б*) изображение получено с использованием функций преобразования, изображенных на рис. 6.25(*а*), где также показаны диапазоны яркости, которые соответствуют взрывчатому веществу, сумке с вещами и фону. Заметим, что, хотя взрывчатое вещество и фон достаточно сильно различаются по яркости, они оказались изображены почти одинаковым цветом. Это произошло из-за периодичности синусоидальных функций. Представленное на рис. 6.24(*в*) изображение получено с использованием функций преобразования, изображенных на рис. 6.25(*б*). В этом случае преобразование действует

а
б

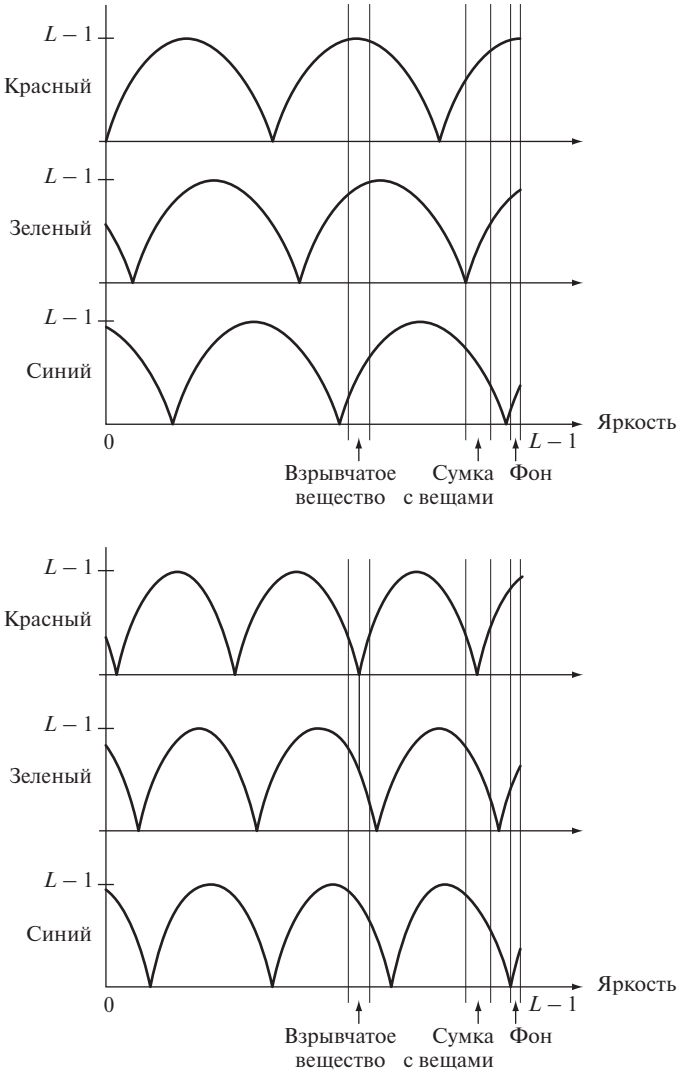


Рис. 6.25. Функции преобразований, использованные для получения изображений на рис. 6.24

почти одинаково в диапазонах яркости, соответствующих взрывчатому веществу и сумке с вещами. Поэтому этим диапазонам был присвоен по существу один и тот же цвет. Отметим, что такое отображение позволяет наблюдателю «видеть» сквозь взрывчатое вещество. Отображение для фона практически не изменилось по сравнению с рис. 6.24(*б*), соответственно почти не изменился и цвет фона. ■

Подход, схематически представленный на рис. 6.23, предполагает работу с одним монохромным изображением. Однако часто интерес представляет объединение нескольких монохромных изображений в одно составное цветное изображение, как показано на рис. 6.26. Такой подход (он проиллюстрирован

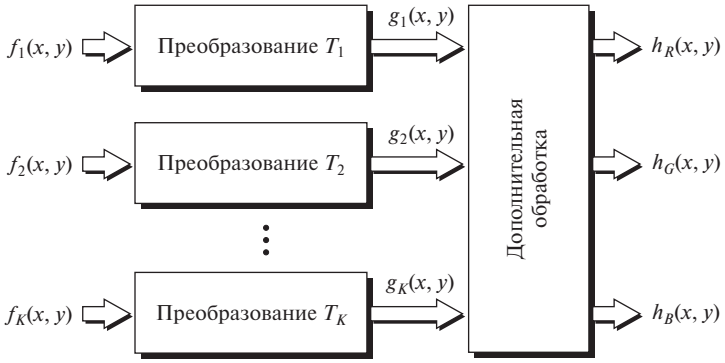


Рис. 6.26. Метод формирования псевдоцветов при наличии нескольких монохромных изображений

примером 6.6 ниже) используется при обработке спектрозональных изображений, когда различные сенсоры формируют отдельные монохромные изображения, каждое в своем спектральном диапазоне. В качестве указанной на рис. 6.26 дополнительной обработки могут выступать такие процедуры, как цветовая коррекция (см. раздел 6.5.4), объединение изображений и выбор трех изображений для воспроизведения на основе информации о приборных функциях сенсоров, использованных при формировании изображений.

Пример 6.6. Цветовое кодирование спектрозональных изображений.
■ На рис. 6.27(*а*)—(*г*) представлены полученные со спутника четыре изображения г. Вашингтона, которые включают часть реки Потомак. Первые три изображения получены в видимых красном, зеленом и синем диапазонах, а четвертое — в ближнем инфракрасном диапазоне (см. табл. 1.1 и рис. 1.10). На рис. 6.27(*д*) представлено изображение в натуральных цветах, полученное при объединении первых трех изображений в RGB-изображение. Изображения густонаселенных областей в натуральных цветах с трудом поддаются интерпретации, но одной из примечательных особенностей этого изображения является изменение цвета в различных частях реки Потомак. Изображение на рис. 6.27(*е*) представляет несколько больший интерес. Это изображение было получено заменой красной компоненты изображения на рис. 6.27(*д*) на изображение в ближнем инфракрасном диапазоне. Из табл. 1.1 нам известно, что этот диапазон очень чувствителен к наличию биомассы в сцене. Рис. 6.27(*е*) достаточно ясно демонстрирует различие между биомассой (красное) и объектами сцены искусственного происхождения, которые состоят в основном из бетона и асфальта и выглядят на изображении голубоватыми.

Только что продемонстрированный вид обработки весьма полезен при визуализации представляющих интерес событий, особенно когда эти события находятся за пределами нашего чувственного восприятия. Рис. 6.28 прекрасно иллюстрирует сказанное. На этом рисунке представлены изображения спутника Юпитера Ио в псевдоцветах, полученные объединением нескольких спектрозональных изображений с космического аппарата «Галилео». Некоторые из компонент получены в невидимых диапазонах спектра. Однако понимание того, как физические и химические процессы могут влиять на ре-

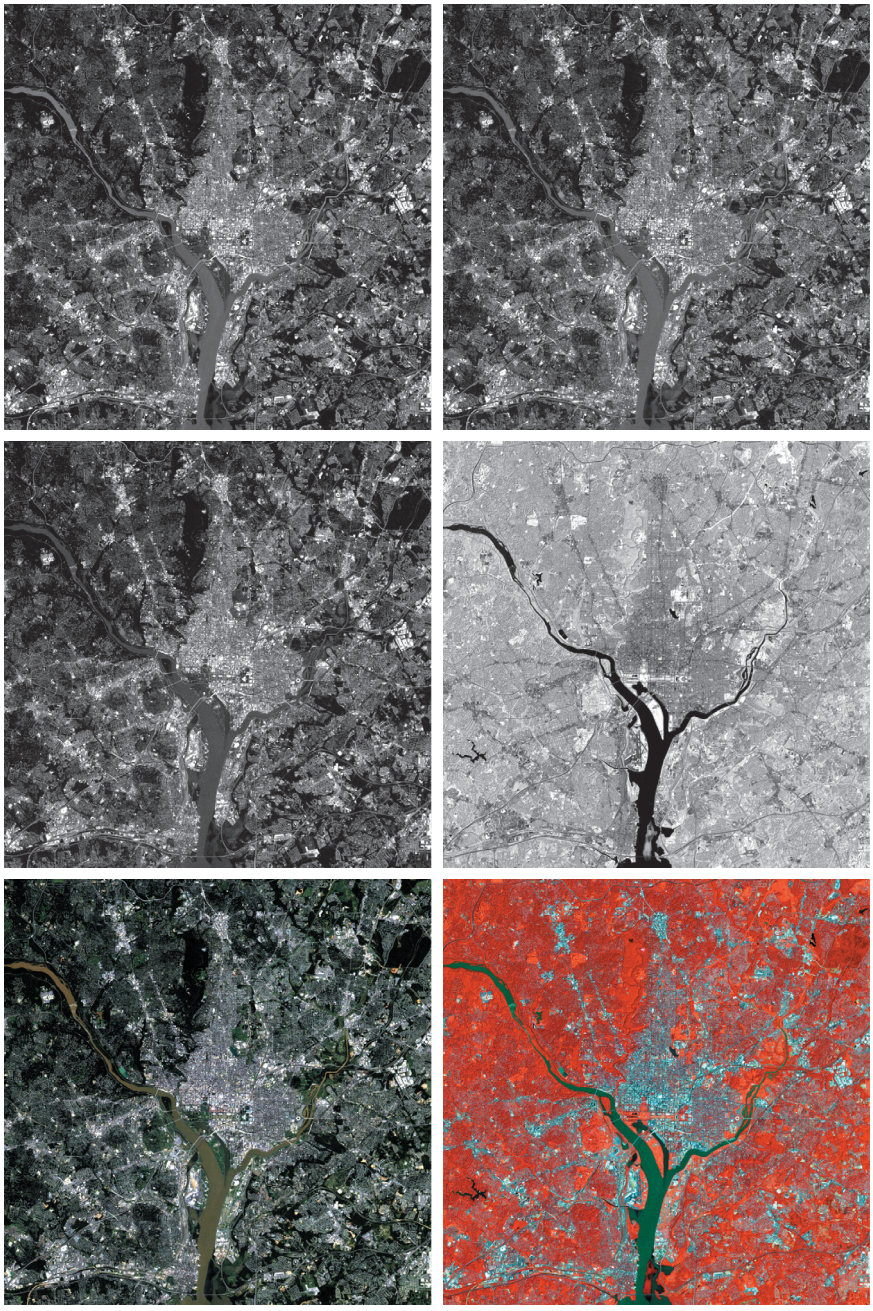
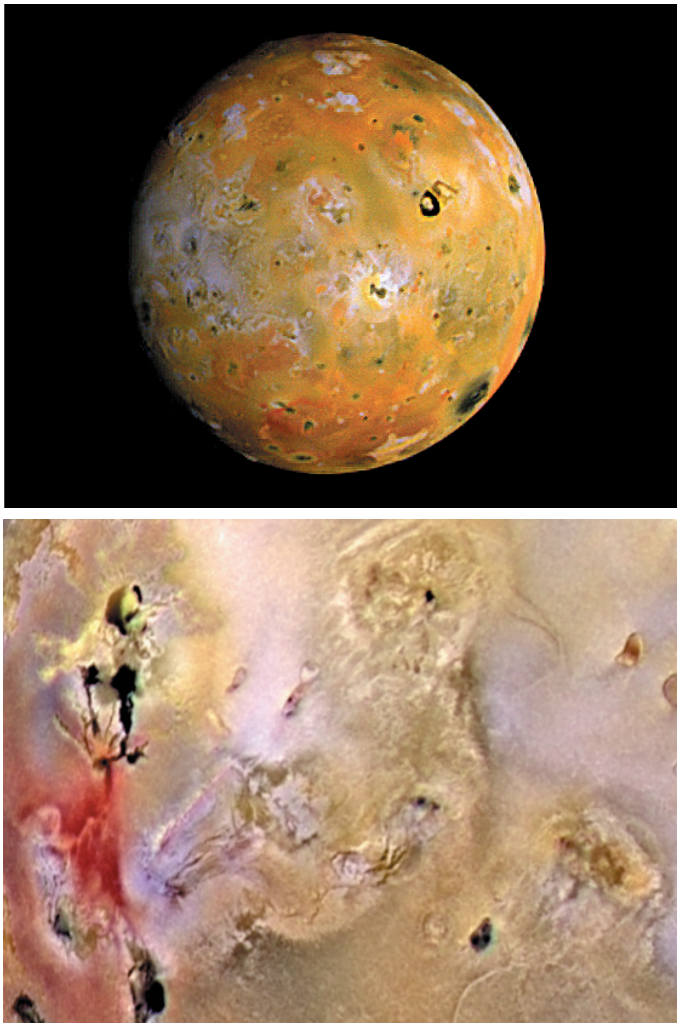


Рис. 6.27. (*а*)—(*г*) Изображения в диапазонах 1—4 на рис. 1.10 (см. табл. 1.1). (*д*) Цветное изображение, полученное при использовании изображений (*а*), (*б*) и (*в*) в качестве красной, зеленой и синей компонент RGB-изображения. (*е*) Изображение, полученное аналогично, с той разницей, что в качестве красной компоненты использовано изображение (*г*) в ближнем инфракрасном диапазоне. (Исходное спектрозональное изображение предоставлено NASA)



а
б

Рис. 6.28. (*а*) Изображение спутника Юпитера Ио в условных цветах. (*б*) Крупный план. (Изображения предоставлены NASA)

зультаты измерений, дает возможность соединить отдельные изображения в поддающуюся интерпретации карту в условных цветах. Один из способов объединения наблюдаемых данных, представленных в виде изображений, может быть основан на том, как эти данные отображают различия в химическом составе поверхности или изменения отражательных свойств поверхности по отношению к солнечному свету. Например, на изображении в условных цветах на рис. 6.28(*б*) ярко-красный цвет соответствует веществу, которое недавно было выброшено на поверхность из активных вулканов Ио, а окружающий желтый цвет представляет более старые серные отложения. Это изображение позволяет намного легче отследить указанные свойства, чем это было бы возможно при исследовании каждой из компонент, составляющих изображение, по отдельности. ■

6.4. Основы обработки цветных изображений

В этом разделе мы приступаем к изучению методов, применимых при обработке изображений в натуральных цветах, или просто цветных изображений. Хотя набор рассматриваемых в последующих разделах методов далеко не является исчерпывающим, эти методы служат иллюстрацией того, как следует обращаться с такими изображениями при решении различных задач их обработки. Используемые при обработке цветных изображений подходы распадаются на две основные категории. Подходы первой категории предполагают, что каждая цветовая компонента изображения обрабатывается отдельно, а затем результирующее цветное изображение составляется из компонент, обработанных по отдельности. Для подходов второй категории характерна непосредственная работа с цветными пикселями. Поскольку цветное изображение содержит, как минимум три составные части, то значение цветного пикселя представляет собой вектор. В RGB-модели, например, значение пикселя изображения может рассматриваться как вектор, проведенный из начала координат в соответствующую точку цветового пространства (см. рис. 6.7).

Пусть **c** есть произвольный вектор в цветовом пространстве RGB:

$$\mathbf{c} = \begin{bmatrix} c_R \\ c_G \\ c_B \end{bmatrix} = \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \tag{6.4-1}$$

Это выражение показывает, что компонентами вектора **c** являются RGB-координаты точки в цветовом пространстве. Запись

$$\mathbf{c}(x,y) = \begin{bmatrix} c_R(x,y) \\ c_G(x,y) \\ c_B(x,y) \end{bmatrix} = \begin{bmatrix} R(x,y) \\ G(x,y) \\ B(x,y) \end{bmatrix} \tag{6.4-2}$$

отражает тот факт, что компоненты вектора **c** зависят от пространственных координат (x, y). Для изображения размерами $M \times N$ существуют MN таких векторов $\mathbf{c}(x, y)$, $x = 0, 1, 2, ..., M - 1$, $y = 0, 1, 2, ..., N - 1$.

Важно ясно отдавать себе отчет в том, что выражение (6.4-2) описывает вектор, компоненты которого являются функциями *пространственных* переменных x и y. Это обстоятельство часто является источником недоразумений, которых можно избежать, если сконцентрироваться на том, что наш интерес лежит в области пространственных методов обработки. А именно Кацухико Токунага нас интересуют неоднородные методы обработки изображений, т. е. зависящие от пространственных координат x и y. Тот факт, что пиксели изображения являются цветными, приводит в простейшем случае к тому, что мы имеем возможность обрабатывать независимо каждую цветовую компоненту изображения, используя обычные методы обработки полутоновых изображений. Однако получаемые таким способом результаты не всегда совпадают с результатами обработки, выполняемой непосредственно в цветовом векторном пространстве; в этом случае требуется разработка новых подходов.

Для того чтобы методы покомпонентной обработки и векторной обработки были эквивалентны, необходимо выполнение двух условий. Во-первых, метод

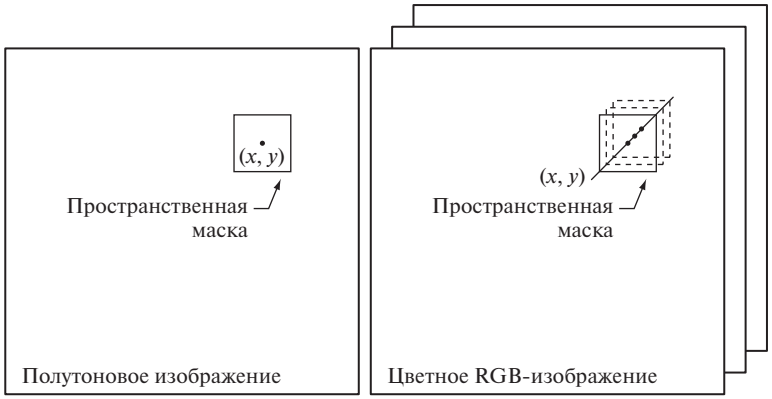


Рис. 6.29. Пространственные маски для полутоновых и цветных RGB-изображений

должен быть применим как к векторам, так и к скалярам. Во-вторых, операции над каждой компонентой вектора не должны зависеть от других компонент. В качестве иллюстрации на рис. 6.29 показана процедура обработки полутонового и цветного изображений по пространственной окрестности. Допустим, что эта процедура состоит в усреднении по окрестности. Для полутонового изображения на рис. 6.29(*а*) процедура усреднения будет заключаться в суммировании значений яркости всех пикселей в заданной окрестности и делении полученного значения на полное число таких пикселей. Для RGB-изображения на рис. 6.29(*а*) процедура усреднения будет заключаться в суммировании всех векторов, отвечающих точкам заданной окрестности, и делении полученного вектора на полное число векторов в окрестности. Но каждая компонента усредненного вектора равна среднему по окрестности от значений изображения этой компоненты. Поэтому тот же самый результат будет получен, если выполнить усреднение отдельно по каждой компоненте и затем из усредненных компонент сформировать искомый вектор. Более детальное рассмотрение приведено в последующих разделах. Мы также рассмотрим методы, для которых два указанных подхода дают различные результаты.

6.5. Цветовые преобразования

Предметом рассматриваемых в настоящем разделе методов, называемых в совокупности *цветовыми преобразованиями*, является обработка компонент цветного изображения в рамках одной отдельно взятой цветовой модели. Это отличает данные преобразования от преобразований координат цвета при переходе из одной цветовой модели в другую (таких как рассмотренные в разделе 6.2.3 преобразования цветов из модели RGB в модель HSI и обратно).

6.5.1. Постановка задачи

Как и для рассмотренных в главе 3 градационных методов преобразования изображений, мы задаем преобразование цветных изображений следующим выражением:

$$g(x,y) = T\left[f(x,y)\right], \tag{6.5-1}$$

где $f(x,y)$ — цветное изображение на входе, $g(x,y)$ — преобразованное, или обработанное, цветное изображение на выходе и T — действующий на изображение f оператор обработки по пространственной окрестности точки (x,y). Принципиальное отличие этого выражения от выражения (3.1-1) заключается в интерпретации последнего. Теперь значение пикселя представляет собой трехмерный или многомерный вектор, т. е. набор из трех или более координат того цветового пространства, которое используется для представления изображения (см. рис. 6.29(*б*)).

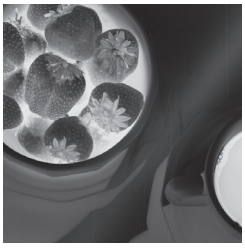
Так же как и при определении основных преобразований яркости для полутоновых изображений в разделе 3.2, мы в этом разделе ограничимся рассмотрением цветовых преобразований вида

$$s_i = T_i(r_1, r_2, \ldots, r_n), \quad i = 1, 2, \ldots, n, \tag{6.5-2}$$

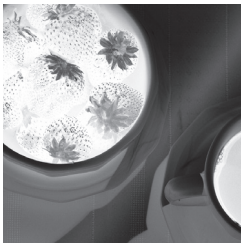
где для упрощения записи переменные r_i и s_i используются для обозначения цветовых компонент изображений $f(x,y)$ и $g(x,y)$ в произвольной точке (x,y), n обозначает число цветовых компонент, а $\{T_1, T_2, \ldots, T_n\}$ — множество *функций преобразования* или *цветового отображения*, которые, действуя на величины r_i, дают величины s_i. Заметим, что вся совокупность n функций преобразования T_i определяет единственное отображение T в выражении (6.5-1). Значение n определяется цветовым пространством, выбранным для описания пикселей изображений f и g. Например, если используется цветовое пространство RGB, то $n = 3$ и переменные r_1, r_2 и r_3 обозначают красную, зеленую и синюю компоненты входного изображения. Для цветовых пространств CMYK и HSI имеем соответственно $n = 4$ и $n = 3$.

В верхней части рис. 6.30 представлено изображение в натуральных цветах вазы с клубникой и чашки кофе. Это изображение высокого разрешения было получено в результате сканирования цветного негатива большого формата (10×12,5 см). Во втором ряду на этом рисунке изображены компоненты исходного CMYK-изображения, полученного в результате сканирования. На этих изображениях в каждой из цветовых компонент CMYK-модели черному соответствует значение 0, а белому — значение 1. Таким образом, мы видим, что цвет клубники состоит из большого количества пурпурного и желтого цветов, так как соответствующие клубнике области являются наиболее яркими на изображениях, отвечающих этим двум CMYK-компонентам. Черный цвет присутствует в малом количестве и в основном ограничен изображением кофе и теней в вазе с клубникой. В третьем ряду представлены результаты преобразования цветов из системы CMYK в систему RGB, из которых видно, что цвет клубники состоит из большого количества красного и содержит зеленый и синий цвета в очень небольшом (но все же в некотором) количестве. В последнем ряду на рис. 6.30 представлены HSI-компоненты исходного изображения, которые вычислялись по формулам (6.2-2)—(6.2-4). Как и следовало ожидать, компонента интенсивности представляет собой полутоновое изображение оригинала. Далее цвет клубники является наиболее насыщенным среди всех цветов изображения, т. е. он менее остальных цветов разбавлен белым цветом; этот цвет обладает самой большой насыщенностью. Отметим, наконец, определенную трудность

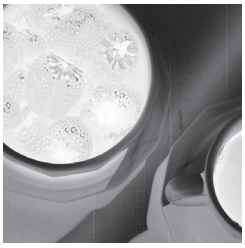
Цветное изображение



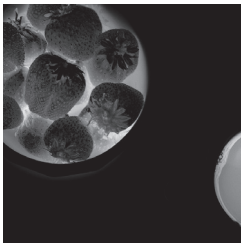
Голубой (C)



Пурпурный (M)



Желтый (Y)

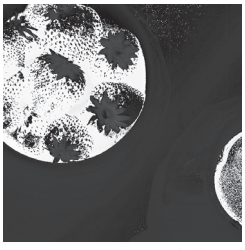


Черный (K)

Красный (R)

Зеленый (G)

Синий (B)



Цветовой тон (H)

Насыщенность (S)

Интенсивность (I)

Рис. 6.30. Цветное изображение и его компоненты, соответствующие различным цветовым пространствам. (Исходное изображение предоставлено MedData Interactive)

в интерпретации компоненты цветового тона, которая обусловлена сочетанием следующих факторов: (1) значение цветового тона в модели HSI претерпевает разрыв в тех точках, которым соответствуют углы 0° и 360° (см. рис. 6.15), и (2) для цветовых точек с нулевой насыщенностью (т. е. белых, черных и серых) значение цветового тона не определено. Разрывность выбранной модели цветового тона наиболее заметна на области, соответствующей клубнике. В точках этой области составляющая цветового тона принимает значения, близкие как к минимуму (0), так и к максимуму (1). В результате мы имеем область с непред-

сказуемо чередующимися крайними значениями диапазона, представляющими один и тот же цвет — красный.

Любой из представленных на рис. 6.30 наборов цветовых компонент может быть использован в преобразовании (6.5-2). Теоретически в рамках каждой из цветовых моделей может быть осуществлено любое преобразование. На практике, однако, некоторые операции лучше приспособлены для реализации в рамках той или иной модели. При принятии решения относительно того, в каком цветовом пространстве реализовывать данное преобразование, необходимо учитывать затраты на переход от одного цветового представления в другое. Предположим, например, что мы хотим изменить интенсивность цветного изображения на рис. 6.30 с помощью преобразования

$$g(x,y) = kf(x,y), \tag{6.5-3}$$

где $0 < k < 1$. В цветовом пространстве HSI этого можно достичь при помощи простого преобразования

$$s_3 = kr_3, \tag{6.5-4}$$

причем $s_1 = r_1$ и $s_2 = r_2$. Изменению подлежит только компонента интенсивности. В цветовом пространстве RGB изменению должны быть подвергнуты все три компоненты:

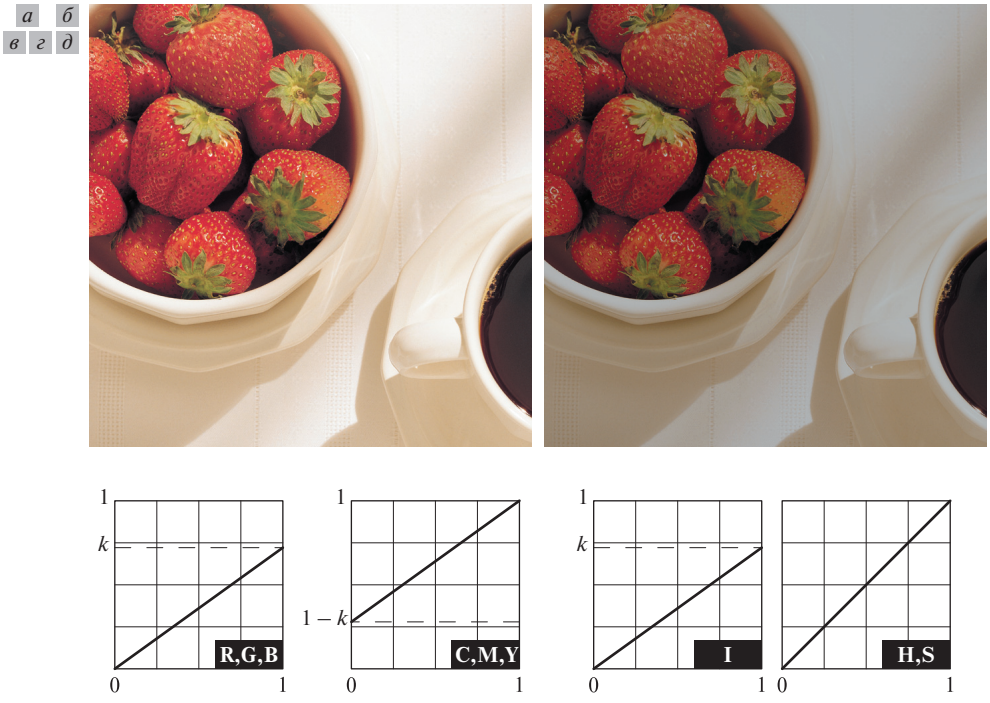


Рис. 6.31. Изменение интенсивности изображения при помощи цветовых преобразований. (*а*) Исходное изображение. (*б*) Результат увеличения его интенсивности на 30 % ($k = 0{,}7$). (*в*)—(*д*) Функции преобразования для моделей RGB, CMY и HSI. (Исходное изображение предоставлено MedData Interactive)

$$s_i = k r_i, \quad i = 1, 2, 3. \tag{6.5-5}$$

Пространство CMY требует применения похожего набора линейных преобразований:

$$s_i = k r_i + (1 - k), \quad i = 1, 2, 3. \tag{6.5-6}$$

Хотя в модели HSI преобразование осуществляется с помощью меньшего числа операций, вычисления, необходимые для перехода из пространства RGB или CMY(K) в пространство HSI, не просто нивелируют (в данном конкретном случае) это преимущество, но и делают такой способ вычисления совершенно неэффективным. Вычислительная сложность перехода в пространство HSI значительно превосходит сложность самого преобразования. Результат преобразования, однако, не зависит от выбранной для его реализации цветовой системы. На рис. 6.31(*б*) представлен результат применения любого из рассматриваемых преобразований (6.5-4)—(6.5-6) со значением $k = 0{,}7$ к изображению на рис. 6.31(*а*), которое является увеличенной копией цветного изображения на рис. 6.30. Графики соответствующих функций преобразования изображены на рис. 6.31(*в*)—(*д*).

Важно отметить, что преобразование каждой из компонент цветового пространства по формулам (6.5-4)—(6.5-6) зависит только от одной этой компоненты. Например, красная компонента s_1 на выходе, в соответствии с (6.5-5), зависит только от красной компоненты r_1 на входе и не зависит от зеленой (r_2) и синей (r_3) компонент на входе. Преобразования такого типа относятся к числу наиболее простых и часто используемых средств цветовой обработки и могут быть осуществлены независимо для каждой отдельной цветовой компоненты, о чем уже шла речь ранее в нашем обсуждении. В оставшейся части этого раздела мы рассмотрим несколько преобразований такого рода и обсудим случаи, когда функции преобразования зависят от всех цветовых компонент входного изображения, и поэтому такое преобразование не может быть осуществлено отдельно по каждой цветовой компоненте.

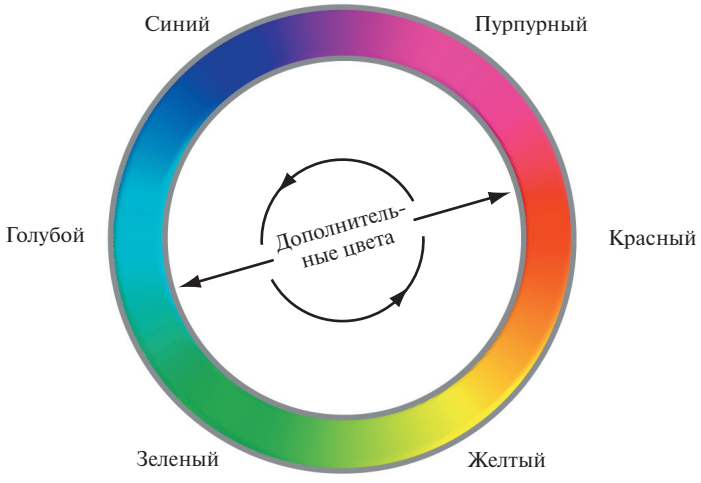


Рис. 6.32. Дополнительные цвета на цветовом круге

6.5.2. Цветовое дополнение

Цвета, расположенные друг напротив друга на изображенном на рис. 6.32 *цветовом круге*[7], называются *дополнительными цветами*. Наш интерес к цветовому дополнению, т. е. к переходу от данных цветов к соответствующим дополнительным цветам, связан с тем обстоятельством, что эта операция аналогична преобразованию полутонового изображения в негатив (см. раздел 3.2.1). Как и в полутоновом случае, цветовое дополнение полезно для выявления деталей внутри темных областей цветного изображения, особенно когда размеры областей заметно превосходят размеры деталей.

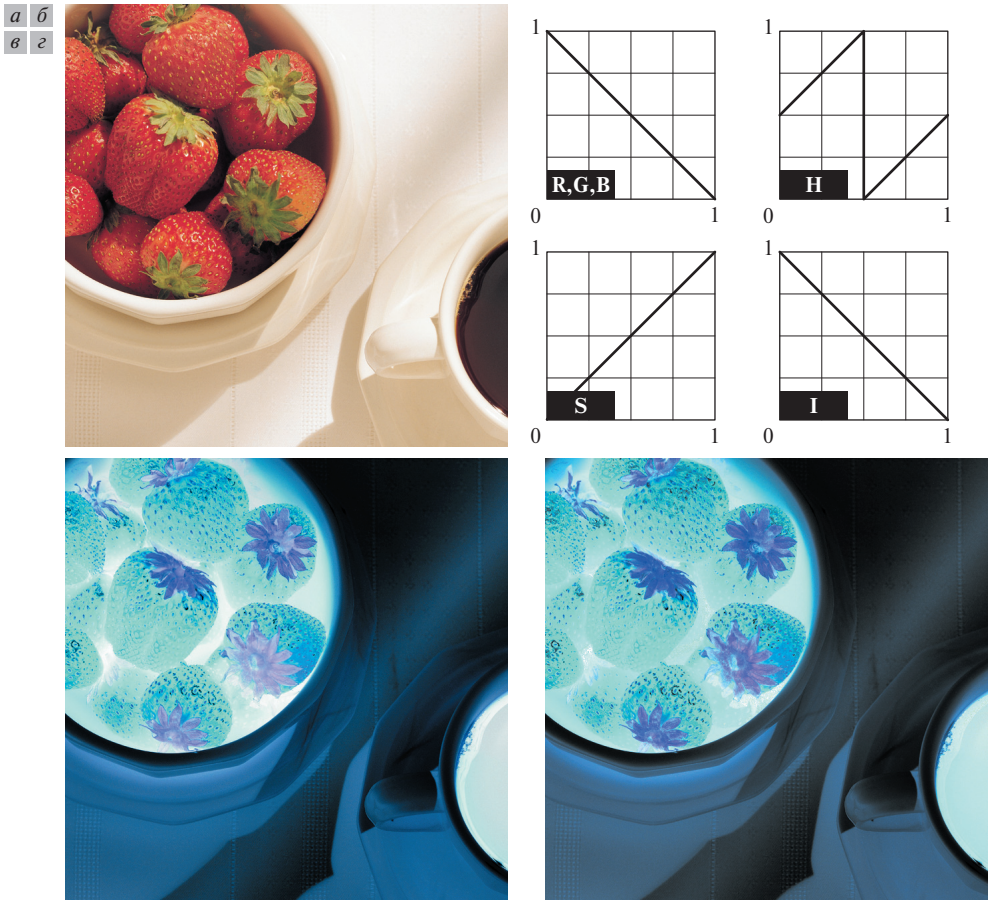


Рис. 6.33. Цветовое дополнение. (*а*) Исходное изображение. (*б*) Функции преобразования цветового дополнения. (*в*) Цветовое дополнение изображения (*а*), посчитанное на основе функций преобразования RGB-модели. (*г*) Приближение для RGB-изображения цветового преобразования, посчитанное на основе функций преобразования HSI-модели

[7] Это понятие появилось в 17 веке и восходит к сэру Исааку Ньютону, который впервые соединил концы видимого спектра и образовал цветовой круг.

Пример 6.7. Вычисление цветового дополнения.

■ На рис. 6.33(*а*) и (*в*) представлены цветное изображение рис. 6.30 и его цветовое дополнение. График функций преобразования, использованных для вычисления цветового дополнения в системе RGB, показан на левом верхнем графике рис. 6.33(*б*). Эти функции совпадают с определенной в разделе 3.2.1 функцией преобразования полутонового изображения в негатив. Заметим, что вычисленное цветовое дополнение напоминает обычный цветной негатив, получаемый на цветной фотопленке. Красные цвета на исходном изображении в цветовом дополнении оказались замененными на голубые. Там, где исходное изображение белое, его цветовое дополнение черное и т. д. Каждый цвет на изображении цветового дополнения может быть предсказан по цвету на исходном изображении с помощью цветового круга на рис. 6.32. И каждая из функций преобразования RGB-компонент, используемая при вычислении цветового дополнения, зависит только от соответствующей компоненты входного изображения.

В отличие от преобразования интенсивности на рис. 6.31, функции преобразования в пространстве RGB для цветового дополнения не имеют простого аналога в пространстве HSI. В качестве упражнения (см. задачу 6.18) читателю предлагается показать, что компонента насыщенности для цветового дополнения не может быть вычислена только на основе насыщенности исходного изображения. На рис. 6.33(*г*) представлено приближение для цветового дополнения, полученное с использованием показанных на рис. 6.33(*б*) функций преобразования для цветового тона, насыщенности и интенсивности (H, S, I). Заметим, что компонента насыщенности не претерпевает изменений; этим и обусловлена некоторая разница между изображениями на рис. 6.33(*в*) и (*г*). ■

6.5.3. Вырезание цветового диапазона

Вырезание определенного диапазона цветов на изображении применяется для выделения некоторых объектов из их окружения. Основная идея заключается в том, чтобы либо (1) воспроизвести интересующие цвета так, чтобы они выступали на общем фоне, либо (2) использовать определяемые цветом области в качестве маски при дальнейшей обработке. Наиболее простой подход состоит в том, чтобы обобщить рассмотренный в разделе 3.2.4 метод вырезания яркостного диапазона. Однако, поскольку значение цветного пикселя представляет собой n-мерный вектор, функции преобразования для вырезания цветового диапазона являются более сложными, чем аналогичные функции для вырезания яркостного диапазона. В действительности интересующие нас преобразования превосходят по сложности все цветовые преобразования, рассмотренные до сих пор. Это связано с тем, что при любом применяемом на практике методе вырезания цветового диапазона каждая цветовая компонента преобразованного пикселя зависит от всех n цветовых компонент исходного пикселя.

Один из простейших способов «разделить» цветное изображение состоит в том, чтобы отобразить все цвета, лежащие вне области интереса, в некоторый нейтральный, не бросающийся в глаза цвет. Если представляющие интерес цвета заключены в некотором кубе (или *гиперкубе* при $n > 3$) с длиной ребра W и с центром в точке $(a_1, a_2, ..., a_n)$ цветового пространства, которая соответствует

некоторому заданному цвету-прототипу, то совокупность необходимых функций преобразования задается выражением

$$s_i = \begin{cases} 0{,}5, & \text{если } |r_j - a_j| > \dfrac{W}{2} \text{ для любого } 1 \le j \le n; \\ r_i, & \text{в остальных случаях;} \end{cases} \quad i = 1, 2, \ldots, n. \qquad (6.5\text{-}7)$$

Эти преобразования выделяют цвета вокруг заданного, заменяя все остальные на цвет средней точки используемого цветового пространства (произвольно выбранный нейтральный цвет). В случае цветового пространства RGB подходящей точкой является средина отрезка серых цветов, т. е. цвет (0,5, 0,5, 0,5).

Если для задания цветовой области, представляющей интерес, используется сфера, то формула (6.5-7) принимает вид

$$s_i = \begin{cases} 0{,}5, & \text{если } \sum_{j=1}^{n} (r_j - a_j)^2 > R_0^2; \\ r_i, & \text{в остальных случаях;} \end{cases} \quad i = 1, 2, \ldots, n. \qquad (6.5\text{-}8)$$

Здесь R_0 — радиус ограничивающей цвет сферы (или *гиперсферы* при $n > 3$), а $(a_1, a_2, \ldots, a_n)$ — координаты центра сферы в цветовом пространстве, определяющие цвет-прототип. В других полезных модификациях формул (6.5-7) и (6.5-8) используются нескольких цветов-прототипов, а интенсивности цветов, лежащих вне области интереса, уменьшаются — вместо замены этих цветов на некоторый нейтральный цвет.

Пример 6.8. Вырезание цветового диапазона.

■ Преобразования (6.5-7) и (6.5-8) могут быть использованы для выделения областей, отвечающих съедобным частям клубники, на рис. 6.31(*а*) из остального фона: чашки, вазы, кофе и поверхности стола. На рис. 6.34(*а*) и (*б*) приведены

а *б*

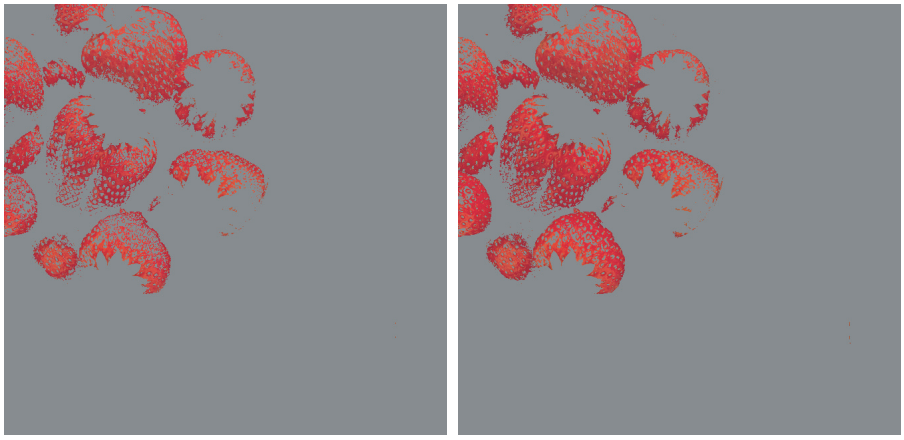


Рис. 6.34. Преобразование, которое вырезает диапазон красных цветов внутри (*а*) куба в RGB-пространстве с длиной грани $W = 0{,}2549$ и с центром в точке (0,6863, 0,1608, 0,1922), и (*б*) сферы в RGB-пространстве радиусом $R_0 = 0{,}1765$ и центром в той же точке. Цвет пикселей, значения которых лежат вне данных куба или шара, заменен серым, равным (0,5, 0,5, 0,5)

результаты применения обоих преобразований. В обоих случаях красный цвет-прототип с RGB-координатами (0,6863, 0,1608, 0,1922) выбирался как доминирующий цвет клубники; значения W и R_0 выбирались таким образом, чтобы наличие светлых областей на изображении не приводило к нежелательному расширению области интереса. Подходящие значения — $W = 0{,}2549$ и $R_0 = 0{,}1765$ — были подобраны экспериментально. Отметим, что преобразование вида (6.5-8), основанное на использовании сферы, приводит к несколько лучшему результату в том смысле, что выделенным оказывается большее количество красных областей, отвечающих клубнике. Сфера радиусом 0,1765 не содержит целиком куба с длиной грани 0,2549, но и сама не содержится целиком в этом кубе. ■

6.5.4. Яркостная и цветовая коррекция

Цветовые преобразования осуществимы на большинстве из персональных компьютеров. В соединении с цифровыми фотокамерами, планшетными сканерами и цветными струйными принтерами они превращают персональный компьютер в *цифровую фотолабораторию*, которая позволяет осуществлять выравнивание яркости и цветовую коррекцию (две операции, составляющие основу любой системы высококачественной цветной репродукции) без применения традиционных средств химической обработки, используемых в обычной фотолаборатории. Хотя яркостная и цветовая коррекции полезны в различных областях обработки изображений, в центре нашего обсуждения будет находиться их наиболее распространенное применение — улучшение фотографий и цветное репродуцирование.

В этом параграфе эффективность преобразований оценивается исключительно по результатам печати. При разработке и совершенствовании этих преобразований оценка осуществляется с использованием монитора, поэтому необходимо обеспечить высокую степень цветового соответствия между используемыми мониторами и возможными устройствами вывода. В действительности монитор должен точно воспроизводить цвета исходного изображения, представленного в цифровом формате, равно как и окончательные цвета изображения в том виде, как они появятся на печати. Наилучшим образом это достигается при использовании *не зависящей от устройства цветовой модели*, которая связывает между собой цветовые охваты (см. раздел 6.1) мониторов, устройств вывода, а также других используемых устройств. Успех такого подхода определяется как качеством *цветовых профилей*, используемых для отображения каждого из устройств в цветовую модель, так и моделью как таковой. В роли такой модели во многих *системах управления цветом* выступает МКО (CIE) модель $L^*a^*b^*$, называемая также моделью CIELAB ([CIE, 1978], [Robertson, 1977]). Цветовые координаты в модели $L^*a^*b^*$ задаются следующими выражениями:

$$L^* = 116 \cdot h\left(\frac{Y}{Y_W}\right) - 16, \tag{6.5-9}$$

$$a^* = 500\left[h\left(\frac{X}{X_W}\right) - h\left(\frac{Y}{Y_W}\right)\right], \tag{6.5-10}$$

$$b^* = 200\left[h\left(\frac{Y}{Y_W}\right) - h\left(\frac{Z}{Z_W}\right)\right], \tag{6.5-11}$$

где

$$h(q) = \begin{cases} \sqrt[3]{q}, & \text{если } q > 0{,}008856, \\ 7{,}787q + 16/116, & \text{если } q \le 0{,}008856, \end{cases} \tag{6.5-12}$$

а величины X_w, Y_W и Z_W представляют собой координаты опорного белого цвета. В качестве такового обычно используется белый свет, отраженный идеальной диффузной поверхностью, освещенной источником *D*65 стандарта МКО (которому на диаграмме цветностей МКО Рис. 6.5 соответствуют координаты цветности $x = 0{,}3127$ и $y = 0{,}3290$). Цветовое пространство $L^*a^*b^*$ является *колориметрическим* (т. е. одинаково воспринимаемые цвета имеют одинаковые цветовые координаты), *равноконтрастным* (т. е. равным изменениям координат цветности соответствуют равные изменения в ощущении цвета — см. классическую работу Мак-Адама [MacAdams, 1942]) и *не зависящим от устройства*. Хотя это цветовое пространство не может быть отображено напрямую (для отображения на экране или при печати необходим переход в другое цветовое пространство), его цветовой охват включает весь видимый спектр и позволяет точно представить цвета любых мониторов, принтеров и других устройств ввода—вывода. Подобно системе HSI, система $L^*a^*b^*$ превосходно разделяет интенсивность (которая представлена яркостью L^*) и цветность (которая представлена двумя цветоразностями: a^* — красный минус зеленый и b^* — зеленый минус синий). Это свойство делает систему $L^*a^*b^*$ весьма удобной как для улучшения изображений (тональной и цветовой коррекции), так и для их сжатия[8].

Главное преимущество калиброванных систем обработки изображений состоит в том, что они позволяют осуществлять яркостную и цветовую коррекции в интерактивном режиме независимо, т. е. в виде двух последовательных операций. Вначале обычно производится коррекция яркостного диапазона изображения, а затем устраняется цветовой дисбаланс, такой как недостаточная или избыточная насыщенность цветов. Яркостной диапазон цветного изображения связан с общим распределением интенсивности его цветов. Как и в монохромном случае, часто желательно распределить значения интенсивности цветного изображения равномерно между наиболее светлыми и наиболее темными значениями. В приводимых ниже примерах рассматриваются различные цветовые преобразования, используемые для яркостной и цветовой коррекции.

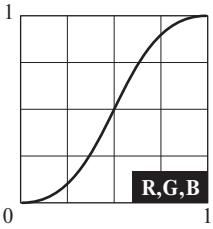
Пример 6.9. Яркостная коррекция.

■ Преобразования для изменения яркости изображения обычно выбираются интерактивно. Задача — так подобрать яркость и контраст изображения, чтобы обеспечить максимальную детализацию изображения в нужном диапазоне интенсивностей. При этом сами цвета на изображении не изменяются. В пространствах RGB и CMY(K) это означает использование одной и той же функции

[8] Исследования показывают, что при использовании модели *L*a*b** достигается наивысшая степень разделения яркостной и цветовой информации по сравнению с другими цветовыми системами, такими как CIELUV, YIQ, YUV, YCC и XYZ [Kasson, Plouffe, 1992].

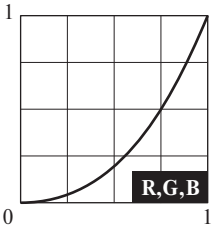
Малоконтрастное изображение

Результат коррекции



Светлое изображение

Результат коррекции



Темное изображение

Результат коррекции

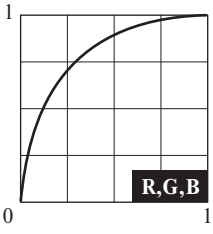


Рис. 6.35. Яркостная коррекция для малоконтрастного, светлого и темного цветных изображений. При одинаковом изменении красной, зеленой и синей компонент цветовой тон изображения не всегда заметно изменяется

преобразования для каждой их трех (или четырех) цветовых компонент; в цветовом пространстве HSI преобразованию подвергается только компонента интенсивности.

На рис. 6.35 представлены типичные преобразования, используемые для коррекции трех основных типов яркостного дисбаланса, когда изображение имеет низкий контраст, является слишком светлым или слишком темным. S-образная кривая в верхнем ряду на этом рисунке идеально подходит для усиления контраста (ср. рис. 3.2(*a*)). Центральная точка этой кривой располагается таким образом, чтобы значения в светлых и темных областях становились соответственно еще более светлыми и темными. (Кривая, обратная по отношению к рассматриваемой, может быть использована для уменьшения контраста.) Преобразования во втором и третьем ряду на рисунке осуществляют коррекцию светлого и темного изображений и напоминают степенные преобразования на рис. 3.6. Хотя в действительности функции преобразования, как и цветовые компоненты, являются дискретными, принято изображать их и работать с ними как с непрерывными функциями; при этом обычно они составлены из кусочно-линейных функций или (для более гладких отображений) из полиномов более высокого порядка. Заметим, что тип яркостного дисбаланса для изображений на рис. 6.35 виден непосредственно, однако его также можно определить исходя из гистограмм значений цветовых компонент изображения. ■

Пример 6.10. Цветовая коррекция.

■ После того как яркостные свойства изображения правильно откорректированы, можно приступать к устранению цветового дисбаланса. Хотя объективно наличие цветового дисбаланса может быть установлено путем анализа на изображении какого-нибудь участка известного цвета с помощью колориметра, однако в том случае, когда на изображении имеются белые области, где значения RGB- или CMY(K)-компонент должны быть равны между собой, возможна и достаточно точная визуальная оценка. Как можно видеть на рис. 6.36, оттенки кожи также являются прекрасным объектом для визуальных оценок цвета, поскольку человеческое зрение крайне чувствительно по отношению к цвету кожи. Яркие цвета, такие как ярко-красный цвет, не слишком полезны, когда речь идет о визуальном оценивании качества цветопередачи.

Существуют различные способы коррекции, применяемые при наличии цветового дисбаланса. При изменении цветовых компонент изображения важно понимать, что каждое действие влияет на весь цветовой баланс на изображении. Другими словами, восприятие каждого цвета зависит от всех окружающих его цветов. Тем не менее, чтобы предсказать, как одна цветовая компонента влияет на остальные, можно использовать цветовой круг, представленный на рис. 6.32. Основываясь на этом цветовом круге можно, например, прийти к заключению, что относительное количество некоторого цвета на изображении может быть увеличено, если уменьшить количество противоположного (дополнительного) цвета. Аналогично это количество увеличивается при одновременном увеличении доли двух цветов, смежных с рассматриваемым цветом, или при одновременном уменьшении доли двух цветов, смежных с дополнительным по отношению к рассматриваемому цвету. Предположим, например,

Исходное изображение

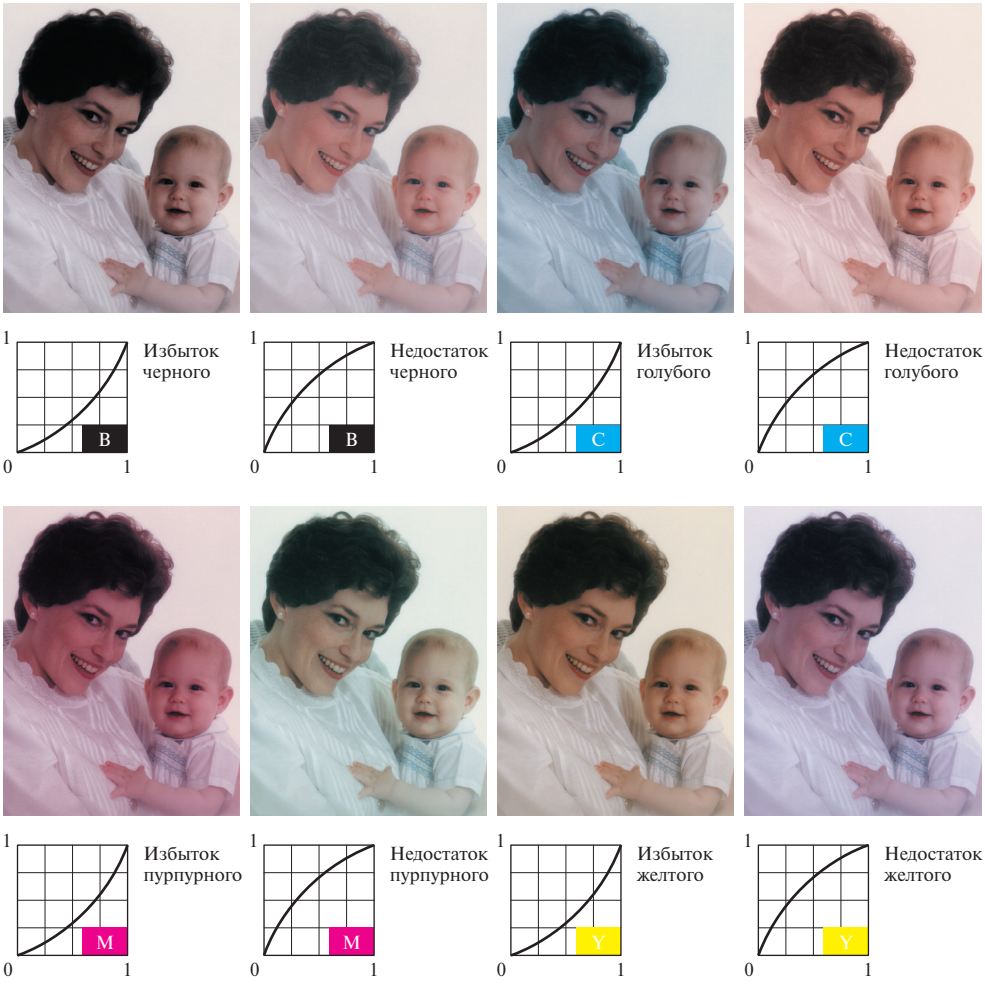


Рис. 6.36. Цветовая коррекция для CMYK-изображений

что на RGB-изображении наблюдается избыток пурпурного цвета. Количество пурпурного может быть уменьшено путем (1) одновременного уменьшения красного и синего или (2) увеличением зеленого.

На рис. 6.36 представлены преобразования, использованные для коррекции простых CMYK-изображений, цветовой баланс которых был предварительно нарушен. Отметим, что на рисунке изображены функции преобразования, необходимые для коррекции изображений (при формировании изображений с нарушенным цветовым балансом были использованы обратные к ним функции). Все вместе эти изображения напоминают калейдоскоп отпечатков, полученных в фотолаборатории, и могут быть полезны как атлас возможных искажений в процессе цветной печати. Заметим, например, что переизбыток красного цвета может быть вызван избытком пурпурного (согласно левому изображению в нижнем ряду) или недостатком голубого (как показывает крайне правое изображение во втором ряду). ■

6.5.5. Обработка гистограмм

В отличие от интерактивных методов обработки предыдущего параграфа, преобразования, рассмотренные в разделе 3.3 и основанные на обработке гистограмм значений яркости, могут применяться к цветным изображениям в автоматическом режиме. Напомним, что целью преобразования эквализации гистограммы является получение изображения с равномерной гистограммой значений яркости. В случае монохромных изображений эффективность метода эквализации гистограмм была продемонстрирована (см. рис. 3.20) для темных, светлых и занимающих промежуточное положение изображений. Однако поскольку цветные изображения состоят из нескольких компонент, то для того чтобы приспособить полутоновую технику для обработки более чем одной компоненты и/или гистограммы, необходимо специальное рассмотрение. Очевидно, выравнивание гистограммы каждой компоненты цветного изображения по отдельности является опрометчивым подходом и может приводить к появлению неверных цветов. Более логичный подход заключается в однородном растяжении значений интенсивности цветов, не меняющем сами цвета (т. е. значения цветового тона). Следующий пример показывает, что цветовое пространство HSI идеально приспособлено для реализации такого подхода.

Пример 6.11. Эквализация гистограммы в цветовом пространстве HSI.

■ На рис. 6.37(*а*) показано цветное изображение вращающейся подставки для специй с находящимися на ней графинчиками и солонками. Значения интенсивности этого изображения занимают весь (нормализованный) диапазон возможных значений [0, 1]. Как можно видеть по гистограмме значений интенсивности до обработки (см. рис. 6.37(*б*)), изображение содержит большое количество темных цветов, что сдвигает медиану в область меньших значений до уровня 0,36. Применение метода эквализации гистограммы к компоненте интенсивности без изменения компонент цветового тона и насыщенности дает изображение, представленное на рис. 6.37(*в*). Заметим, что изображение в целом выглядит

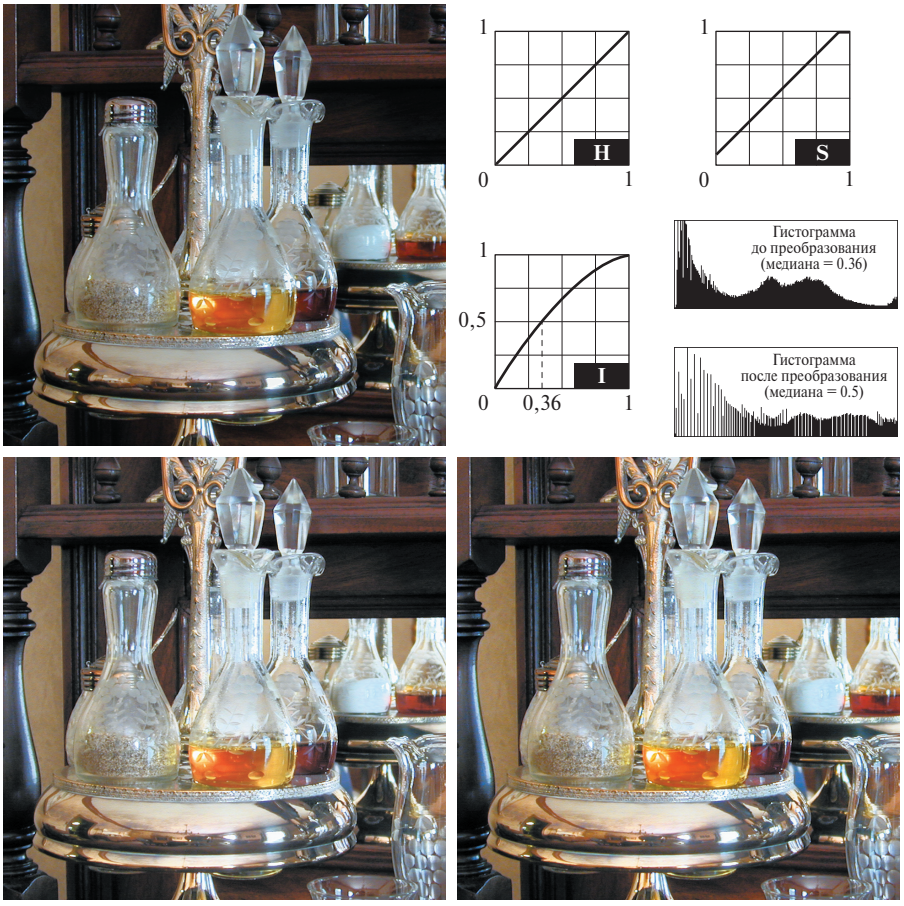


Рис. 6.37. Эквализация гистограммы (с последующей коррекцией насыщенности) в цветовом пространстве HSI

значительно более ярким и что теперь стали видны некоторые формы и текстура деревянного стола, на котором расположена стойка для специй. На рис. 6.37(*б*) представлена гистограмма значений интенсивности этого нового изображения, а также функция преобразования интенсивности, использованная при эквализации (см. формулу (3.3-8)).

Хотя использованный метод эквализации интенсивности не изменил значения цветового тона и насыщенности изображения, он сильно повлиял на цветовосприятие изображения в целом. Отметим, в частности, что оказались утеряны переливы цвета, характерные для масла и уксуса в графинчиках. Частично это удается скомпенсировать последующим увеличением значений компоненты насыщенности изображения с помощью функции преобразования, показанной на рис. 6.37(*б*). Соответствующий результат приведен на рис. 6.37(*г*). Такой тип коррекции часто используется при работе с компонентой интенсивности в пространстве HSI, поскольку изменения в интенсивности обычно влияют на то, как выглядят цвета на изображении. ■

6.6. Сглаживание и повышение резкости

В предыдущем разделе мы рассматривали преобразования цветных изображений, при которых значение каждого пикселя изменялось независимо от значений соседних пикселей. Наш следующий шаг состоит в выходе за рамки таких преобразований и рассмотрении преобразований, при которых значение каждого пикселя изображения изменяется в соответствии с характеристиками окружающих пикселей. В настоящем разделе основные принципы такого рода локальной обработки рассматриваются на примерах сглаживания и повышения резкости цветных изображений.

6.6.1. Сглаживание цветных изображений

В соответствии с обсуждениями, проведенными в разделах 3.4 и 3.5, и рис. 6.29(*а*), процедура сглаживания черно-белого полутонового изображения может рассматриваться как операция пространственной фильтрации, при которой все коэффициенты фильтрующей маски имеют одинаковые значения. По мере того как маска перемещается по сглаживаемому изображению, значение каждого пикселя заменяется средним значением пикселей в окрестности, определяемой маской. Как показывает рис. 6.29(*б*), данная концепция легко распространяется на цветные изображения. Основная разница состоит в том, что вместо скалярных значений яркости мы должны оперировать компонентами векторов вида (6.4-2).

Пусть S_{xy} обозначает совокупность координат, определяемых некоторой окрестностью с центром в точке (x, y) в плоскости цветного RGB-изображения. Среднее значение трехкомпонентного вектора RGB по этой окрестности равно

$$\bar{\mathbf{c}}(x,y) = \frac{1}{K} \sum_{(s,t) \in S_{xy}} \mathbf{c}(s,t). \tag{6.6-1}$$

Из выражения (6.4-2) и свойств векторного сложения следует, что

$$\bar{\mathbf{c}}(x,y) = \begin{bmatrix} \dfrac{1}{K} \displaystyle\sum_{(s,t) \in S_{xy}} R(s,t) \\ \dfrac{1}{K} \displaystyle\sum_{(s,t) \in S_{xy}} G(s,t) \\ \dfrac{1}{K} \displaystyle\sum_{(s,t) \in S_{xy}} B(s,t) \end{bmatrix}. \tag{6.6-2}$$

Для ознакомления с операциями с векторами и матрицами обратитесь к интернет-сайту книги.

Мы видим, что компоненты полученного среднего вектора представляют собой значения скалярных изображений, которые могут быть вычислены независимым сглаживанием каждой цветовой компоненты исходного RGB-изображения с использованием определенной «полутоновой» процедуры обработки

по окрестности. Таким образом, мы приходим к заключению, что сглаживание методом усреднения по окрестности может быть осуществлено отдельно в каждой цветовой плоскости (по каждой цветовой компоненте). При этом получаемый результат совпадает с результатом векторного усреднения, выполняемого в RGB-пространстве.

Пример 6.12. Сглаживание цветного изображения с помощью усреднения по окрестности.

■ Рассмотрим цветное RGB-изображение, представленное на рис. 6.38(*а*). Его красная, зеленая и синяя цветовые компоненты изображены на рис. 6.38(*б*)—(*г*). На рис. 6.39(*а*)—(*в*) представлены HSI-компоненты этого изображения. Согласно сказанному выше, было осуществлено независимое покомпонентное сглаживание RGB-изображения на рис. 6.38 при помощи пространственной усредняющей маски размерами 5×5. Затем эти независимо сглаженные изображения были объединены и в результате сформировано сглаженное цветное изображение. Посчитанное таким образом изображение представлено на рис. 6.40(*а*).

а б
в г

Рис. 6.38. (*а*) RGB-изображение. Компоненты изображения: (*б*) красная, (*в*) зеленая и (*г*) синяя

а б в

Рис. 6.39. HSI-компоненты цветного RGB-изображения на рис. 6.38(*а*). (*а*) Цветовой тон. (*б*) Насыщенность. (*в*) Интенсивность

а б в

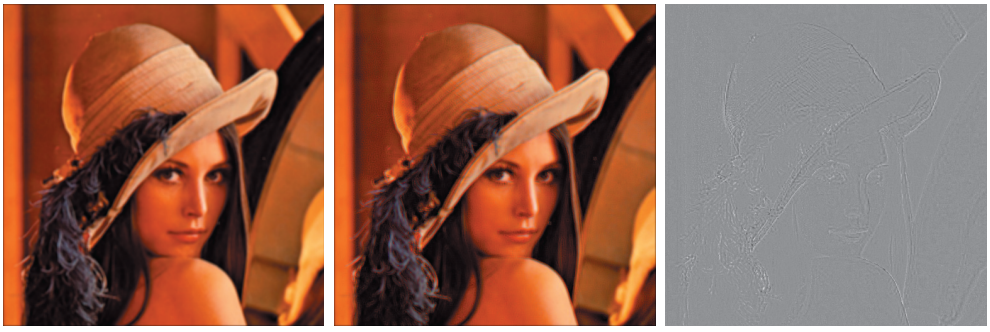


Рис. 6.40. Сглаживание изображения с помощью усредняющей маски размерами 5×5. (*а*) Результат независимой обработки каждой из RGB-компонент изображения. (*б*) Результат обработки компоненты интенсивности в HSI-пространстве с последующим переходом в RGB-пространство. (*в*) Разность между этими двумя результатами

Отметим, что оно выглядит так, как и можно было ожидать от операции пространственного сглаживания и в соответствии с примерами раздела 3.5.

В разделе 6.2 отмечалось, что важным преимуществом цветовой модели HSI является то, что в ней происходит разделение интенсивности (которая тесно связана с полутоновой яркостью) и цветовой информации. Это делает данную модель удобной для применения многих методов полутоновой обработки и наводит на мысль о том, что может оказаться более эффективным осуществлять сглаживание только компоненты интенсивности HSI-представления, которое изображено на рис. 6.39. Для того чтобы продемонстрировать достоинства и/или последствия такого подхода, осуществим теперь сглаживание только компоненты интенсивности (оставив без изменений компоненты цветового тона и насыщенности), а для воспроизведения преобразуем полученный результат обратно в RGB-изображение. Сглаженное изображение представлено на рис. 6.40(*б*). Заметим, что это изображение весьма близко к изображению на рис. 6.40(*а*), но, как показывает изображение их разности на рис. 6.40(*в*), два данных сглаженных изображения не идентич-

ны. Причина этого в том, что у изображения на рис. 6.40(*а*) цвет каждого пикселя является средним цветом пикселей в окрестности. С другой стороны, сглаживание только компоненты интенсивности изображения на рис. 6.40(*б*) не меняет значений исходного цветового тона и насыщенности, тем самым сохраняя исходный цвет пикселей. Из данного высказывания следует, что разница между двумя вариантами сглаживания будет возрастать с увеличением размера сглаживающей маски. ■

6.6.2. Повышение резкости цветных изображений

В этом параграфе мы рассмотрим повышение резкости изображения при помощи лапласиана (см. раздел 3.6.2). Из векторного анализа нам известно, что лапласиан вектора определяется как вектор, компоненты которого равны лапласиану от каждой отдельной скалярной компоненты входного вектора. В цветовом пространстве RGB лапласиан вектора **c** вида (6.4-2) равен

$$\nabla^2[\mathbf{c}(x,y)] = \begin{bmatrix} \nabla^2 R(x,y) \\ \nabla^2 G(x,y) \\ \nabla^2 B(x,y) \end{bmatrix}. \qquad (6.6\text{-}3)$$

Это означает, что вычисление лапласиана цветного изображения может быть осуществлено путем вычисления лапласиана отдельно для каждой компоненты изображения.

Пример 6.13. Повышение резкости с помощью лапласиана.

■ Изображение на рис. 6.41(*а*) было получено применением формулы (3.6-7) и маски на рис. 3.37(*в*) к отдельным RGB-компонентам изображения на рис. 6.38. Результаты были объединены в цветное изображение повышенной резкости. На рис. 6.41(*б*) представлен похожий результат повышения резкости, основанный на использовании изображенных на рис. 6.39 HSI-компонент. Для получения этого результата вычислялся лапласиан только

а б в

Рис. 6.41. Повышение резкости изображения с помощью лапласиана. (*а*) Результат независимой обработки каждой RGB-компоненты. (*б*) Результат обработки компоненты интенсивности и последующего перехода в RGB-пространство. (*в*) Разность двух результатов

от компоненты интенсивности, а значения цветового тона и насыщенности не изменялись. Разность между результатами повышения резкости RGB и HSI представлена на рис. 6.41(*в*). Причина различия результатов та же, что и в примере 6.12. ■

6.7. Сегментация изображения, основанная на цвете

Под сегментацией вообще понимается процесс разбиения изображения на отдельные области. Хотя сегментация является предметом рассмотрения главы 10, мы кратко обсудим цветовую сегментацию здесь для целостности изложения. У читателя не должно возникнуть трудностей в понимании нижеследующего материала.

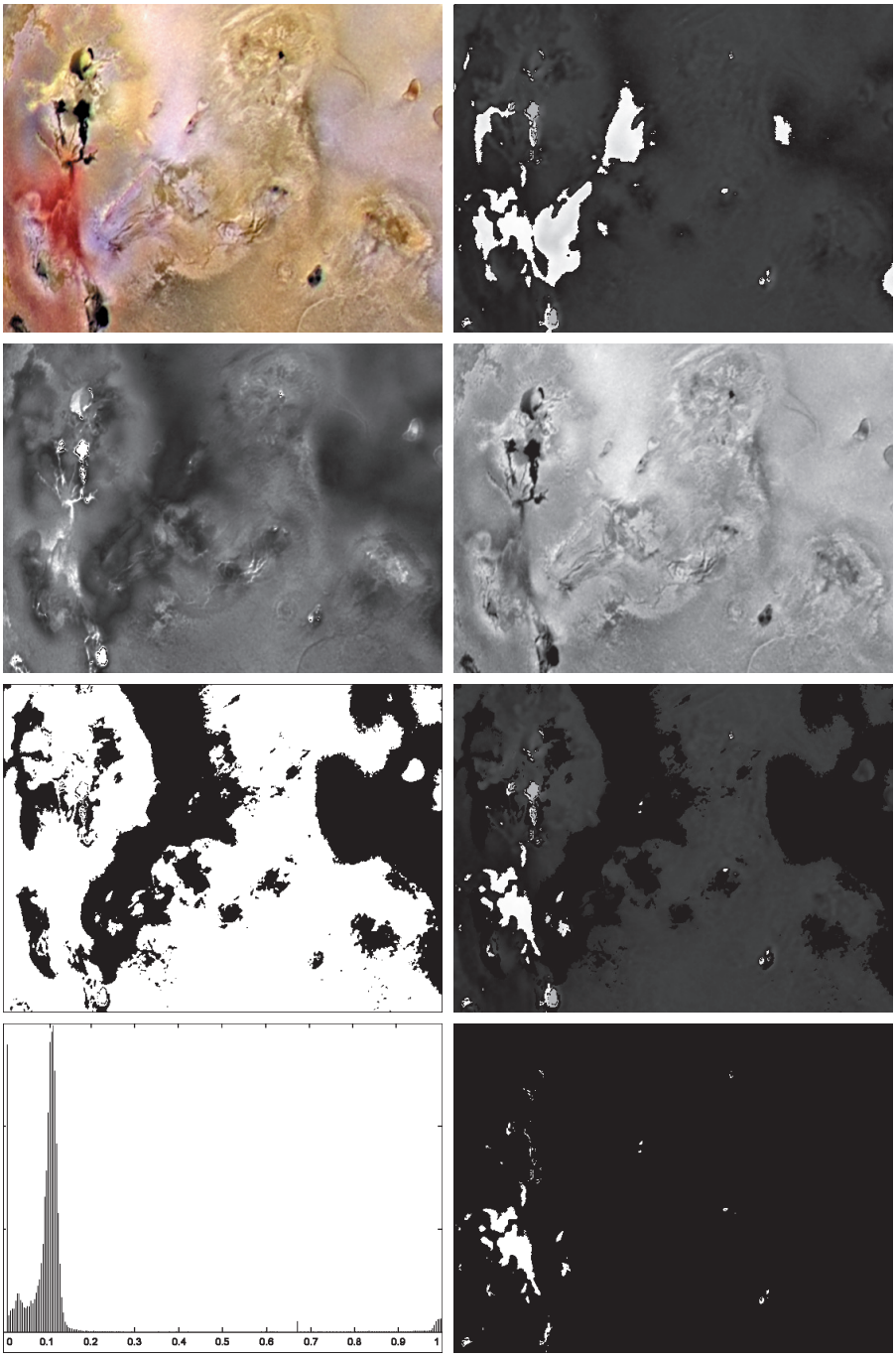
6.7.1. Сегментация в цветовом пространстве HSI

Если мы хотим отсегментировать изображение на основе цвета и, кроме того, хотим иметь возможность выполнять эту операцию отдельно по компонентам, то первое, что естественно приходит в голову, — это использовать пространство HSI, поскольку в этом пространстве основная информация о цвете содержится в компоненте цветового тона. Компонента насыщенности обычно используется для формирования маски, позволяющей в дальнейшем выделить области интереса в компоненте цветового тона. Компонента интенсивности при сегментации цветных изображений используется реже, так как она не несет цветовой информации. Ниже приводится типичный пример того, как осуществляется сегментация в цветовом пространстве HSI.

Пример 6.14. Сегментация в пространстве HSI.

■ Предположим, что перед нами стоит задача выделения области красноватого оттенка в левой нижней части изображения на рис. 6.42(*а*). Хотя это изображение является изображением в условных цветах, его можно обрабатывать (сегментировать) как обычное цветное изображение. На рис. 6.42(*б*)—(*г*) представлены изображения его HSI-компонент. Сравнение рис. 6.42(*а*) и (*б*) показывает, что в интересующей нас области цветовой тон принимает относительно высокие значения; это указывает на то, что соответствующие цвета сдвинуты в пурпурно-синюю область красного цвета (см. рис. 6.13). На рис. 6.42(*д*) представлена двоичная маска, сформированная в результате применения процедуры порогового разделения к изображению насыщенности, причем величина порога выбиралось равной 10 % от максимального значения данного изображения. Любому пикселю, значение насыщенности которого превышало пороговое, присваивалось значение 1 (белое). Всем остальным пикселям было присвоено значение 0 (черное).

На рис. 6.42(*е*) показано произведение двоичной маски и изображения цветового тона, а на рис. 6.42(*ж*) — гистограмма этого произведения (отметим, что яркость произведения принимает значения в диапазоне [0, 1]). Из гистограммы видно, что большие по величине значения (которые представляют для нас интерес) сгруппированы в крайнем правом конце яркостного диапазона вблизи значения 1,0. Применение процедуры порогового разделения (со значением поро-



а *б*
в *г*
д *е*
ж *з*

Рис. 6.42. Сегментация изображений в пространстве HSI. (*а*) Исходное изображение. (*б*) Цветовой тон. (*в*) Насыщенность. (*г*) Интенсивность. (*д*) Двоичная маска насыщенности (черное соответствует значению 0). (*е*) Произведение изображений (*б*) и (*д*). (*ж*) Гистограмма значений яркости изображения (*е*). (*з*) Результат сегментации изображения (*а*)

га 0,9) к изображению произведения на рис. 6.42(*е*) дает двоичное изображение, которое представлено на рис. 6.42(*з*). Положение белых точек указывает те места исходного изображения, цвет которых имеет интересующий нас красноватый оттенок. Полученный результат сегментации далек от идеального: на исходном изображении остались точки, про которые без сомнения можно сказать, что они имеют красноватый оттенок, но которые не были опознаны использованным методом сегментации[9]. Можно, однако, показать экспериментально, что области, отмеченные на рис. 6.42(*з*) белым, представляют собой практически лучший результат по идентификации областей красноватого оттенка, которого можно достичь с помощью рассмотренного метода. Метод сегментации, рассматриваемый в следующем параграфе, позволяет получать значительно лучшие результаты. ■

6.7.2. Сегментация в цветовом пространстве RGB

Несмотря на то, что работа в пространстве HSI, как неоднократно указывалось в настоящей главе, лучше отвечает нашим интуитивным представлениям, тем не менее сегментация является той областью, в которой более хорошие результаты обычно достигаются при работе в цветовом пространстве RGB. Предлагаемый метод достаточно ясен. Предположим, что нашей задачей является сегментация объектов на RGB-изображении, цвет которых лежит в некотором определенном диапазоне. Для некоторой выборки векторов в цветовом пространстве, репрезентативной по отношению к интересующим нас цветам, мы получаем оценку «среднего» цвета, подлежащего выделению. Пусть вектор **a** RGB-пространства обозначает этот средний цвет. Задача сегментации заключается в том, чтобы классифицировать каждый пиксель данного изображения в соответствии с тем, попадает ли его цвет в заданный диапазон или нет. Для того чтобы производить такое сопоставление, необходимо иметь в цветовом пространстве некоторую меру сходства. Простейшей такой мерой является евклидово расстояние. Пусть **z** — произвольная точка в RGB-пространстве. Будем говорить, что точка **z** сходна по цвету с точкой **a**, если расстояние между ними не превышает некоторого заданного порогового значения D_0. Евклидово расстояние между точками **z** и **a** дается выражением

$$D(\mathbf{z},\mathbf{a}) = \|\mathbf{z}-\mathbf{a}\| = \left[(\mathbf{z}-\mathbf{a})^T(\mathbf{z}-\mathbf{a})\right]^{\frac{1}{2}} = \left[(z_R-a_R)^2+(z_G-a_G)^2+(z_B-a_B)^2\right]^{\frac{1}{2}}, \qquad (6.7\text{-}1)$$

где нижние индексы *R*, *G* и *B* используются для обозначения RGB-компонент векторов **a** и **z**. Геометрическое место точек, таких что $D(\mathbf{z}, \mathbf{a}) \le D_0$, представляет

[9] Как указано в разделе 6.2.3, значения цветового тона вблизи 1 и вблизи 0 дают весьма близкие оттенки цвета пикселей (в нашем случае красного, поскольку в качестве нулевого уровня цветового тона был выбран именно красный цвет). Из-за этого при пороговом выборе точек с большим значением цветового тона точки близкого цвета с малым значением (вблизи 0) пропали. Вообще, поскольку значения цветового тона распределены на окружности, то правильнее осуществлять не пороговое разделение, а выделение точек внутри некоторой дуги окружности. Отметим также, что в практических задачах полезно выбирать нулевую точку на участке дуги с наиболее редко встречающимися цветами. — *Прим. перев.*

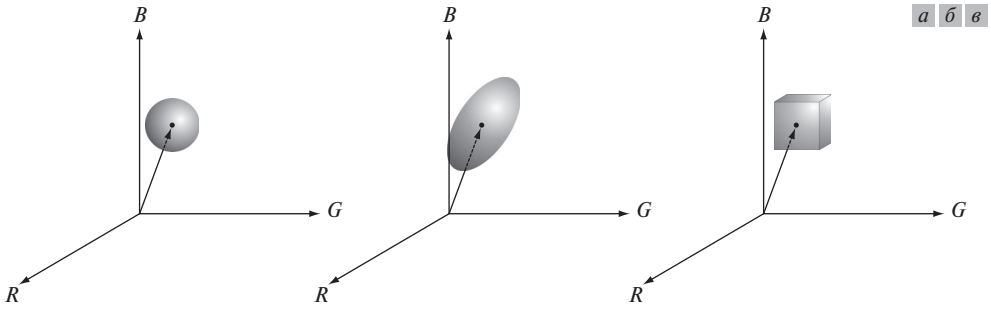


Рис. 6.43. Три способа выделения областей данных для сегментации в цветовом пространстве RGB с использованием: (*а*) шара, (*б*) эллипсоида и (*в*) параллелепипеда

собой шар радиуса D_0, показанный на рис. 6.43(*а*). Точки, лежащие внутри или на поверхности шара, удовлетворяют заданному цветовому критерию; точки вне шара — не удовлетворяют. Если присвоить двум множествам точек на изображении два различных значения, скажем, 1 (белое) и 2 (черное), то получится двоичное изображение, представляющее собой результат сегментации.

Полезным обобщением (6.7-1) является расстояние, задаваемое выражением вида

$$D(\mathbf{z},\mathbf{a}) = \left[(\mathbf{z}-\mathbf{a})^T \mathbf{C}^{-1}(\mathbf{z}-\mathbf{a})\right]^{\frac{1}{2}}, \qquad (6.7\text{-}2)$$

где **C** — ковариационная матрица[10], посчитанная по выборке векторов цветового пространства, репрезентативной по отношению к подлежащему выделению цвету. Геометрическое место точек, таких что $D(\mathbf{z}, \mathbf{a}) \le D_0$, представляет собой эллипсоид в трехмерном пространстве (см. рис. 6.43(*б*)), важное свойство которого состоит в том, что направление его главной оси совпадает с направлением наибольшего разброса данных выборки. Если $\mathbf{C} = \mathbf{I}$, т. е. ковариационная матрица равна единичной матрице размерами 3×3, то (6.7-2) сводится к (6.7-1). Процесс сегментации осуществляется так же, как описано в предыдущем абзаце.

Расстояние является положительно определенной функцией. Это дает возможность использовать не само расстояние, а его квадрат, что позволяет избежать операции извлечения квадратного корня при вычислениях. Однако даже в этом случае при изображениях обычных размеров реализация (6.7-1) или (6.7-2) требует большого объема вычислений. Компромисс заключается в том, чтобы использовать прямоугольный параллелепипед, который изображен на рис. 6.43(*в*). При таком подходе предполагается, что центр параллелепипеда находится в точке **a**, а размеры параллелепипеда в направлении каждой из осей в цветовом пространстве выбираются пропорциональными стандартному отклонению значений цветовых координат, соответствующих этой оси, в выборке. Вычисление стандартных отклонений по заданной выборке векторов в цветовом пространстве осуществляется только один раз.

Как и в формализме, основанном на использовании расстояния, сегментация для произвольной точки цветового пространства сводится к решению во-

[10] Вопрос вычисления ковариационной матрицы для совокупности векторов детально рассматривается в разделе 11.4.

проса о том, лежит ли эта точка внутри или на поверхности параллелепипеда или нет. Однако для параллелепипеда этот вопрос с вычислительной точки зрения является существенно более простым, чем для областей, имеющих сферическую или эллипсоидальную формы. Отметим, что приведенное рассмотрение представляет собой обобщение метода, который обсуждался в разделе 6.5.3 в связи с вырезанием цветового диапазона.

Пример 6.15. Цветовая сегментация изображения в пространстве RGB.
■ Прямоугольная область, показанная на рис. 6.44(*а*), содержит выборку цветов красноватого оттенка, которые мы хотим выделить на данном цветном изображении. Это та же задача, которая решалась в примере 6.14 с использованием

а
б

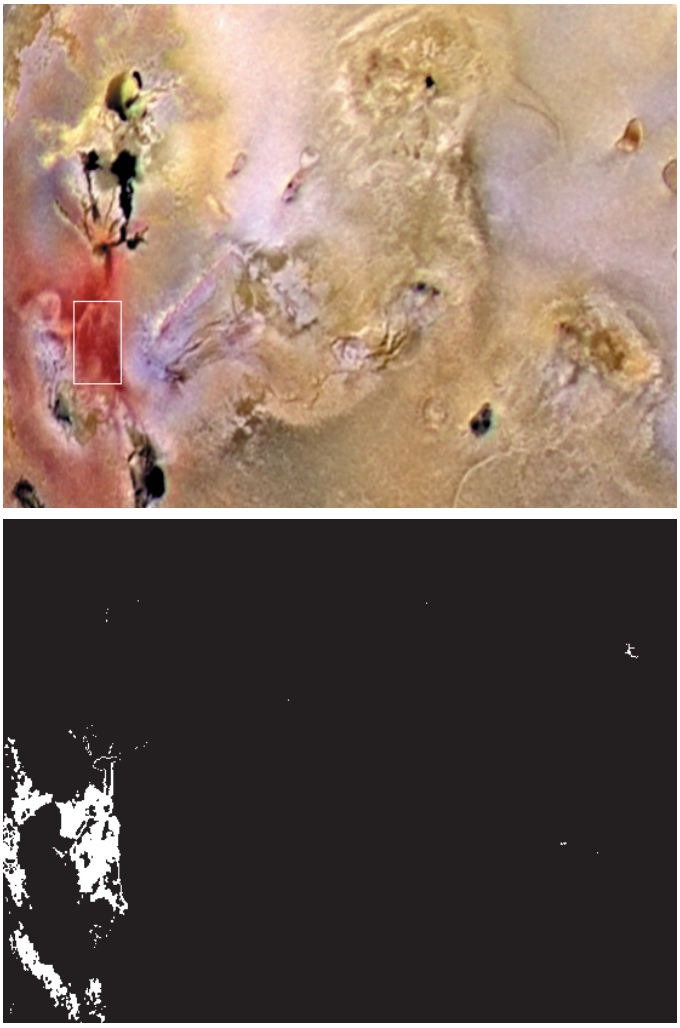


Рис. 6.44. Сегментация в пространстве RGB. (*а*) Исходное изображение; цвета, представляющие интерес, расположены внутри прямоугольника. (*б*) Результат сегментации в цветовом пространстве RGB. Сравните этот результат с результатом, представленным на рис. 6.42(*з*)

компоненты цветового тона, но теперь для ее решения применятся подход, основанный на использовании цветового пространства RGB. При таком подходе по совокупности цветовых векторов, заключенных в прямоугольной области на рис. 6.44(*а*), вычислялся средний цветовой вектор *a*, после чего вычислялись стандартные отклонения для значений красной, зеленой и синей цветовых координат векторов этой совокупности. Центр параллелепипеда располагался в точке *a*, а размеры параллелепипеда в направлении каждой из осей в RGB-пространстве выбирались так, чтобы они в 1,25 раза превосходили стандартное отклонение данных, соответствующих этой оси. Пусть, например, стандартное отклонение значений красной компоненты в выборке было равно σR. Тогда точкам параллелепипеда отвечали значения координат от $(aR - 1{,}25\sigma R)$ до $(aR + 1{,}25\sigma R)$ по оси R, где aR — значение красной компоненты среднего вектора *a*. Далее каждому пикселю цветного изображения присваивалось одно значение (белое), если значение этого пикселя лежало внутри или на поверхности параллелепипеда, и другое значение (черное) в противном случае. Полученный результат представлен на рис. 6.44(*б*). Обратим внимание на то, насколько распространилась область, выделенная на основе выборки цветовых векторов, заключенных в прямоугольнике. Сравнение рис. 6.44(*б*) и рис. 6.42(*з*) показывает, что сегментация в цветовом пространстве RGB позволила получить результаты, существенно более точные в том смысле, что они гораздо лучше отвечают тому, что мы бы определили как точки красноватого оттенка на исходном изображении. ■

6.7.3. Обнаружение контуров на цветных изображениях

Как будет обсуждаться в главе 10, важным инструментом для сегментации изображений является обнаружение контуров. Основанный на использовании контуров метод сегментации подробно рассматривается в разделе 10.2. В настоящем параграфе нас интересует вопрос о том, чем отличается прямое вычисление контуров в цветовом векторном пространстве от вычисления контуров с помощью покомпонентной обработки.

В разделе 3.6.4 в связи с задачей повышения резкости изображений мы познакомились с методом обнаружения контуров, основанным на использовании градиента. К сожалению, операция взятия градиента, в том виде как она была введена в разделе 3.6.4, не определена для векторных величин. Отсюда немедленно следует, что вычисление градиента по изображениям отдельных компонент и использование полученных результатов для формирования цветного изображения приводит к неверным окончательным результатам. Следующий простой пример поможет нам понять, почему это происходит.

Рассмотрим представленные на рис. 6.45(*г*) и (*з*) два цветных изображения размерами $M \times M$ (M — нечетное число), каждое из которых состоит из трех компонент, изображенных соответственно на рис. 6.45(*а*)—(*в*) и (*д*)—(*ж*). Если, например, мы вычислим градиент для изображений каждой отдельной компоненты (см. уравнение 3.6.11)), а затем сложим результаты и сформируем два соответствующих изображения градиента, то значения градиента в точке $[(M + 1)/2, (M + 1)/2]$ будут одинаковы в обоих случаях. Однако интуитивно следовало бы ожидать,

Рис. 6.45. (*а*)—(*в*) Изображения R-, G- и B-компонент и (*г*) соответствующее цветное RGB-изображение. (*д*)—(*ж*) Изображения R-, G- и B-компонент и (*з*) соответствующее цветное RGB-изображение

что градиент в этой точке должен быть больше для изображения на рис. 6.45(*г*), поскольку границы на изображениях всех его составляющих R, G и B проходят в одном направлении, в то время как для изображения на рис. 6.45(*з*) в одном направлении проходят только две из трех таких границ. Этот простой пример показывает, что обработка, производимая отдельно в трех цветовых плоскостях, и последующее формирование составного изображения градиента могут приводить к ошибочным результатам. Если задача состоит только в нахождении контуров, то основанный на такой покомпонентной обработке метод обычно дает приемлемые результаты. Однако для задач, в которых вопросы точности имеют первостепенное значение, очевидно требуется такое новое определение градиента, которое было бы применимо к векторным величинам. Далее мы рассмотрим метод, предложенный в этой связи Ди Зензо [Di Zenzo, 1986].

Перед нами стоит задача определить вектор градиента (его величину и направление) для векторной функции $\mathbf{c}(x, y)$ вида (6.4-2) в любой точке (x, y). Как мы уже упоминали, введенная в разделе 3.6.4 операция вычисления градиента применима к скалярной функции $f(x, y)$ и неприменима к векторным функциям. Ниже приведен один из возможных способов обобщения понятия градиента на векторные функции. Напомним, что для скалярной функции $f(x, y)$ градиент представляет собой вектор, направление которого совпадает с направлением наибольшей скорости изменения функции f в точке с координатами (x, y).

Пусть $\mathbf{r}$, $\mathbf{g}$ и $\mathbf{b}$ — единичные векторы в направлении осей R, G и B цветового пространства RGB. Определим следующие два вектора

$$\mathbf{u} = \frac{\partial R}{\partial x}\mathbf{r} + \frac{\partial G}{\partial x}\mathbf{g} + \frac{\partial B}{\partial x}\mathbf{b} \tag{6.7-3}$$

и
$$\mathbf{v} = \frac{\partial R}{\partial y}\mathbf{r} + \frac{\partial G}{\partial y}\mathbf{g} + \frac{\partial B}{\partial y}\mathbf{b}. \tag{6.7-4}$$

Определим величины g_{xx}, g_{yy} и g_{xy} через скалярные произведения этих векторов следующим образом:

$$g_{xx} = \mathbf{u}\cdot\mathbf{u} = \mathbf{u}^T\mathbf{u} = \left|\frac{\partial R}{\partial x}\right|^2 + \left|\frac{\partial G}{\partial x}\right|^2 + \left|\frac{\partial B}{\partial x}\right|^2, \tag{6.7-5}$$

$$g_{yy} = \mathbf{v}\cdot\mathbf{v} = \mathbf{v}^T\mathbf{v} = \left|\frac{\partial R}{\partial y}\right|^2 + \left|\frac{\partial G}{\partial y}\right|^2 + \left|\frac{\partial B}{\partial y}\right|^2 \tag{6.7-6}$$

и
$$g_{xy} = \mathbf{u}\cdot\mathbf{v} = \mathbf{u}^T\mathbf{v} = \frac{\partial R}{\partial x}\frac{\partial R}{\partial y} + \frac{\partial G}{\partial x}\frac{\partial G}{\partial y} + \frac{\partial B}{\partial x}\frac{\partial B}{\partial y}. \tag{6.7-7}$$

Не будем забывать, что величины R, G и B, а следовательно, и величины g_{xx}, g_{yy} и g_{xy} являются функциями переменных x и y. Можно показать [Di Zenzo, 1986], что угол θ, в направлении которого скорость изменения функции $\mathbf{c}(x, y)$ максимальна, удовлетворяет уравнению

$$\operatorname{tg}2\theta = \frac{2g_{xy}}{(g_{xx} - g_{yy})}, \tag{6.7-8}$$

а величина скорости изменения в точке (x, y) в направлении угла $\theta(x, y)$ дается выражением

$$F_\theta(x,y) = \left\{\frac{1}{2}\left[(g_{xx} + g_{yy}) + (g_{xx} - g_{yy})\cos 2\theta(x,y) + 2g_{xy}\sin 2\theta(x,y)\right]\right\}^{\frac{1}{2}}. \tag{6.7-9}$$

Поскольку $\operatorname{tg}(\alpha) = \operatorname{tg}(\alpha \pm \pi)$, то если θ_0 является решением уравнения (6.7-8), тогда и $\theta_0 \pm \pi/2$ является решением этого уравнения. Более того, так как $F(\theta) = F(\theta + \pi)$, то величину F достаточно вычислить только для значений θ из полуоткрытого интервала $[0, \pi)$. Тот факт, что уравнение (6.7-8) имеет два решения, отличающиеся на 90°, означает, что это уравнение связывает с каждой точкой (x, y) пару взаимно перпендикулярных направлений. Величина скорости изменения F максимальна вдоль одного из них и минимальна вдоль другого. Подробный вывод приведенных результатов занимает много места, и его воспроизведение здесь мало что добавило бы в плане понимания главной рассматриваемой нами задачи. Интересующие детали можно найти в работе [Di Zenzo, 1986]. Частные производные, необходимые при реализации (6.7-5)—(6.7-7), могут быть вычислены, например, с помощью рассмотренных в разделе 3.6.4 операторов Собела.

Пример 6.16. Обнаружение контуров в цветовом векторном пространстве.

■ На рис. 6.46(*а*) представлено цветное изображение, а на рис. 6.46(*б*) — его градиент, полученный с помощью только что рассмотренного векторного метода. На рис. 6.46(*в*) представлено изображение, которое было получено в результате вычисления градиента от каждой RGB-компоненты исходного изображения и последующего формирования изображения составного градиента путем сло-

а б
в г

Рис. 6.46. (*а*) RGB-изображение. (*б*) Градиент, посчитанный в цветовом векторном пространстве RGB. (*в*) Градиент, посчитанный как сумма градиентов отдельных цветовых компонент. (*г*) Разность изображений (*б*) и (*в*)

а б в

Рис. 6.47. Изображения градиентов компонент цветного изображения на рис. 6.46(*а*). Градиенты: (*а*) красной компоненты, (*б*) зеленой компоненты и (*в*) синей компоненты. Изображение на рис. 6.46(*в*) получено в результате сложения этих изображений и масштабирования

жения соответствующих значений трех полученных компонент в каждой точке (x, y). Изображение градиента на рис. 6.46(*б*), полученное векторным методом, содержит более подробные детали контуров, чем аналогичное изображение на рис. 6.46(*в*), полученное в результате покомпонентной обработки. В качестве примера обратим внимание на мелкие детали в области правого глаза модели. Изображение на рис. 6.46(*г*) представляет собой разность двух изображений градиента. Важно отметить, что оба подхода дают приемлемые результаты. Вопрос о том, сто́ят ли дополнительные детали на рис. 6.46(*б*) затраченных на их вычисление дополнительных усилий (по сравнению с реализацией операторов Собела, которые были использованы для вычисления градиента в отдельных цветовых плоскостях), может быть решен только исходя из требований конкретной задачи. На рис. 6.47 представлены изображения градиента каждой из трех компонент исходного изображения, которые после сложения и масштабирования дали изображения на рис. 6.46(*в*). ■

6.8. Шум на цветных изображениях

Рассмотренные в разделе 5.2 модели шума применимы и к цветным изображениям. Обычно шум на цветном изображении имеет одинаковые характеристики в каждом цветовом канале, но иногда шум влияет на разные цветовые каналы по-разному. Такое возможно, например, в случае неисправности электроники одного из каналов. Однако чаще различия в уровне шума вызываются различиями в относительном уровне освещенности каждого из цветовых каналов. Например, установка режекторного фильтра, ослабляющего красную часть спектра, при съемке на ПЗС-камеру уменьшит интенсивность освещения красных сенсоров. Поскольку шум ПЗС-сенсоров увеличивается при низком уровне освещенности, то уровень шума красной компоненты цветного RGB-изображения будет в этом случае превосходить уровень шума двух других компонент.

Пример 6.17. Иллюстрация эффектов, сопутствующих преобразованию RGB-изображений с шумом в систему HSI.

■ В этом примере мы кратко рассмотрим шум на цветных изображениях и остановимся на том, как преобразуется шум при переходе из одной цветовой модели в другую. На рис. 6.48(*а*)—(*в*) представлены три компоненты RGB-изображения, искаженные гауссовым шумом, а на рис. 6.48(*г*) представлено соответствующее цветное изображение. Отметим, что подобного рода мелкозернистый шум заметен на цветном изображении в меньшей степени, чем на монохромном изображении. На рис. 6.49(*а*)—(*в*) представлены результаты преобразования RGB-изображения на рис. 6.48(*г*) в цветовую систему HSI. Сравните эти результаты с изображениями HSI-компонент исходного изображения (см. рис. 6.39) и обратите внимание, как сильно искажены компоненты цветового тона и насыщенности для изображения с шумом. Это связано с нелинейностью функций арккосинуса и минимума, входящих в формулы преобразования (6.2-2) и (6.2-3). С другой стороны, изображение интенсивности на рис. 6.49(*в*) является

Рис. 6.48. (*а*)—(*в*) Красная, зеленая и синяя компоненты изображения, искаженные аддитивным гауссовым шумом с нулевым средним и дисперсией 800. (*г*) Соответствующее RGB-изображение. (Сравните изображение на рисунке (*г*) с изображением на рис. 6.46(*а*))

Рис. 6.49. HSI компоненты цветного изображения с шумом на рис. 6.48(*г*). (*а*) Цветовой тон. (*б*) Насыщенность. (*в*) Интенсивность

Рис. 6.50. (*а*) Цветное RGB-изображение, зеленая компонента которого искажена биполярным импульсным шумом. (*б*) Компонента цветового тона соответствующего HSI-изображения. (*в*) Компонента насыщенности. (*г*) Компонента интенсивности

несколько более гладким по сравнению со всеми тремя изображениями RGB компонент. Это объясняется тем, что в соответствии с формулой (6.2-4) компонента интенсивности представляет собой среднее значение RGB-компонент. (Напомним, что усреднение уменьшает случайный шум на изображении, как это обсуждалось в разделе 3.4.2.)

В том случае, когда лишь одна из RGB-компонент искажена шумом, преобразование в цветовую модель HSI приводит к распространению шума на все HSI-компоненты. Это продемонстрировано на рис. 6.50. На рис. 6.50(*а*) представлено RGB-изображение, зеленая компонента которого искажена биполярным импульсным шумом, причем вероятности импульсов обеих полярностей равны 0,05. Представленные на рис. 6.50(*б*)—(*г*) изображения HSI-компонент наглядно показывают, что шум одной зеленой компоненты RGB-изображения распространяется на все компоненты HSI-изображения. Это, разумеется, не является

неожиданностью, поскольку для вычисления каждой из HSI-компонент используются все RGB компоненты, что и показывают формулы раздела 6.2.3. ■

Подобно другим рассмотренным выше методам, фильтрация цветных изображений может осуществляться на основе покомпонентной обработки или прямо в цветовом векторном пространстве, в зависимости от выбранного метода. Например, уменьшение шума при помощи усредняющего фильтра — это метод, рассмотренный в разделе 6.6.1, который, как нам известно, дает одинаковые результаты как при непосредственном применении в цветовом векторном пространстве, так и в случае, когда каждая цветовая компонента изображения обрабатывается отдельно. Другие фильтры, однако, не обладают подобным свойством. Примером таких фильтров являются фильтры, основанные на порядковых статистиках, которые рассматривались в разделе 5.3.2. Например, для применения медианного фильтра в цветовом векторном пространстве необходимо определить способ упорядочивания векторов, так чтобы медиана имела смысл. Тогда как для скаляров дело обстоит очень просто, в случае векторов упорядочивание представляет собой значительно более сложный процесс, рассмотрение которого выходит за рамки нашего обсуждения. Хорошим руководством по вопросам упорядочивания векторов и построению фильтров, основанных на таком упорядочивании, может служить книга [Plataniotis, Venetsanopoulos, 2000].

6.9. Сжатие цветных изображений

Поскольку число битов, необходимых для представления цвета, обычно в три или четыре раза превосходит число битов, используемое для представления полутонов, то центральную роль при хранении и передаче цветных изображений играет *сжатие данных*. Что касается рассмотренных в предыдущих параграфах RGB-, CMY(K)- и HSI-изображений, то в роли данных, подлежащих сжатию, выступают значения компонент каждой точки цветного изображения (например красная, зеленая и синяя компоненты пикселей RGB-изображения); именно посредством этих значений передается цветовая информация. *Сжатие* представляет собой процесс уменьшения или устранения избыточных и/или несущественных данных. Хотя сжатие изображений является предметом рассмотрения главы 8, ниже мы кратко продемонстрируем основную идею на примере цветного изображения.

Пример 6.18. Пример сжатия цветного изображения.
■ На рис. 6.51(*а*) приведено 24-битовое цветное RGB-изображение цветков ириса; для представления каждого из значений красной, зеленой и синей компонент используется по 8 битов. Изображение на рис. 6.51(*б*) было восстановлено из сжатой копии изображения (*а*) и является по существу, приближением изображения (*а*), полученным в результате последовательного применения процедур сжатия и восстановления. Хотя сжатое изображение не может быть воспроизведено непосредственно — перед выводом на цветной монитор изображение должно быть восстановлено, оно содержит всего 1 бит данных (и, следователь-

а
б

Рис. 6.51. Сжатие цветного изображения. (*а*) Исходное RGB-изображение. (*б*) Результат последовательного сжатия и восстановления изображения (*а*)

но, 1 бит информации для хранения) на каждые 230 битов данных исходного изображения. Если предположить, что сжатое изображение может быть передано, скажем, через Интернет за одну минуту, то передача исходного изображения потребовала бы почти четырех часов. Конечно, для визуализации переданные данные необходимо восстановить, но эта операция может быть выполнена за время порядка секунды. При формировании изображения на рис. 6.51(*б*) был использован алгоритм сжатия, основанный на недавно предложенном стандарте JPEG-2000, который подробно рассматривается в разделе 8.2.10. Отметим, что восстановленное изображение является слегка размытым, что характерно для многих методов сжатия с потерями. Этот эффект может быть уменьшен или устранен при изменении степени сжатия. ■

Заключение

Материал этой главы представляет собой введение в обработку цветных изображений, и круг рассматриваемых вопросов был выбран таким образом, чтобы выработать твердое понимание основ тех методов, которые используются в этой области обработки изображений. Наша трактовка теории цвета и цветовых моделей должна рассматриваться как изложение основополагающего материала для той сферы, которая сама по себе является обширной технической областью с большим количеством приложений. В частности, мы уделили большое внимание цветовым моделям, которые, как нам кажется, не просто полезны в цифровой обработке изображений, но дают абсолютно необходимый инструмент для дальнейшего изучения вопросов, связанных с обработкой цветных изображений. Рассмотрение методов обработки изображений в псевдоцветах и в натуральных цветах, которые основаны на работе с отдельными компонентами, устанавливает связь с методами обработки полутоновых изображений, которые детально изучались в главах 3—5.

При рассмотрении материала, посвященного цветовым векторным пространствам, мы отходим от тех методов, которые были изучены нами ранее, и особо подчеркиваем некоторые важные различия между методами обработки в полутонах и в цвете. Существуют многочисленные способы прямой обработки цветовых векторов, которые включают такие операции, как медианная фильтрация, адаптивная фильтрации, морфологическая фильтрация, восстановление, сжатие и многие другие. Эти операции не эквивалентны цветовой обработке, выполняемой отдельно для каждой компоненты цветного изображения. В приведенных ниже ссылках указаны работы, которые содержат дальнейшие результаты в этой области.

Наша трактовка шума на цветных изображениях также показывает, что векторная природа задачи и то обстоятельство, что цветное изображение может быть стандартным образом преобразовано из одного цветового пространства в другое, имеют важные последствия в задаче уменьшения шума на таких изображениях. В некоторых случаях фильтрацию шума можно осуществить с помощью покомпонентной обработки. В других случаях, например при медианной фильтрации, для работы с пикселями цветного изображения требуется предпринять специальные меры, которые учитывали бы то обстоятельство, что значения пикселей представляют собой векторные величины.

Хотя сегментация изображений является предметом рассмотрения главы 10, а сжатие изображений — главы 8, мы предварительно рассмотрели их здесь в контексте обработки цветных изображений для того, чтобы обеспечить связность изложения. Как будет ясно из последующих обсуждений, многие из развиваемых в этих главах методов применимы к материалу настоящей главы.

Ссылки и литература для дальнейшего изучения

В качестве наиболее полного источника информации, касающейся науки о цвете, можно рекомендовать [Malacara, 2001]. По поводу физиологии цветовосприятия см. [Gegenfurtner, Sharpe, 1999]. Эти две работы вместе с более ранними

книгами [Walsh, 1958] и [Kiver, 1965] содержат богатый дополнительный материал к разделу 6.1. В качестве дальнейшего чтения по цветовым моделям (раздел 6.2) см. [Fortner, Meyer, 1997], [Poynton, 1996] и [Fairchild, 1998]. Подробный вывод формул преобразования для HSI-модели (раздел 6.2.3) можно найти в работе [Smith, 1978] или на веб-сайте книги. Тема псевдоцветов (раздел 6.3) тесно связана с общей проблемой визуализации данных. Основы использования псевдоцветов хорошо изложены в работе [Wolff, Yaeger, 1993]. Также представляет интерес книга [Thorell, Smith, 1990]. По поводу векторного представления цветовых данных (раздел 6.4) см. [Plataniotis, Venetsanopoulos, 2000].

Ссылками к разделу 6.5 являются [Benson, 1985], [Robertson, 1977] и [CIE, 1978]; см. также классическую работу [MacAdam, 1942]. Изложение материала, относящегося к вопросам фильтрации цветных изображений (раздел 6.6), основано на введенном в разделе 6.4 векторном формализме и нашем рассмотрении вопросов пространственной фильтрации в главе 3. Вопросы сегментации цветных изображений (раздел 6.7) были предметом пристального внимания на протяжении последних десяти лет. Статьи [Liu, Yang, 1994] и [Shafarenko et al., 1998] дают представление о работах в этой области. Представляет интерес также и специальный выпуск [*IEEE Transactions on Image Processing*, 1997]. Изложение вопросов обнаружения контуров на цветном изображении (раздел 6.7.3) взято из работы [Di Zenzo, 1986]. Книга [Plataniotis, Venetsanopoulos, 2000] хороша тем, что в ней собраны различные подходы к проблеме сегментации цветных изображений. Изложение в разделе 6.8 основывается на введенных в разделе 5.2 моделях шума. Ссылки по поводу сжатия изображений (раздел 6.9) приведены в конце главы 8. Особенности программной реализации многих методов, рассмотренных в данной главе, можно найти в книге [Gonzalez,Woods, Eddins, 2004].

Задачи

Детальное рассмотрение задач, помеченных звездочкой, можно найти на интернет-сайте книги. Также там содержатся примерные проекты, базирующиеся на материалах данной главы.

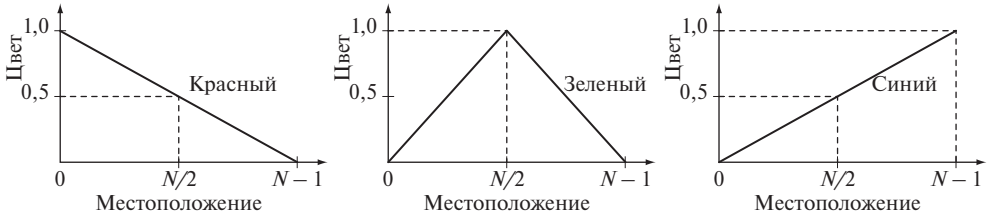
6.1 Укажите процентное содержание красного (X), зеленого (Y) и синего (Z) света, необходимое для получения цвета, который отмечен на рис. 6.5 как «дневной свет».

★**6.2** Рассмотрите два цвета c_1 и c_2 с координатами (x_1, y_1) и (x_2, y_2) на диаграмме цветностей на рис. 6.5. Выведите общее выражение (или выражения) для вычисления относительного процентного содержания цветов c_1 и c_2 в смеси, составляющей некоторый заданный цвет, про который известно, что он лежит на отрезке, соединяющем эти два цвета.

6.3 Выберите любые четыре цвета c_1, c_2, c_3 и c_4 с координатами (x_1, y_1), (x_2, y_2), (x_3, y_3) и (x_4, y_4) на диаграмме цветностей на рис. 6.5. Выведите общее выражение (или выражения) для вычисления относительного процентного содержания цветов c_1, c_2, c_3 и c_4 в смеси, составляющей некоторый заданный цвет, про который известно, что он лежит внутри четырехугольника с вершинами в точках выбранных цветов c_1, c_2, c_3 и c_4.

★**6.4** В задаче автоматизированной сборки четыре типа деталей должны различаться по цвету, чтобы их было проще идентифицировать. Однако предназначенная для регистрации изображений TV-камера является монохромной. Предложите способ, который позволил бы использовать такую камеру для обнаружения четырех различных цветов.

6.5 Простое RGB-изображение имеет такие горизонтальные профили интенсивности своих R-, G- и B-компонент, как представлено ниже на диаграммах. Какой цвет отвечает среднему столбцу этого изображения?



★**6.6** Опишите в общих чертах, как будут выглядеть на монохромном мониторе RGB-компоненты представленного ниже изображения. Все цвета на изображении имеют максимальную интенсивность и насыщенность. При решении задачи рассматривайте бордюр как часть изображения (бордюр имеет средний серый цвет).

6.7 Сколько существует различных оттенков серого цвета в таком RGB-пространстве, в котором каждая цветовая компонента представляет собой 12-битовое изображение?

6.8 Рассмотрите цветовой куб RGB, представленный на рис. 6.8, и ответьте на следующие вопросы.

★ **(а)** Опишите, как изменяется яркость на изображениях первичных цветов *R*, *G* и *B*, которые составляют верхнюю грань цветового куба.

(б) Предположим, что мы заменили каждый цвет в RGB-кубе на соответствующий CMY цвет и воспроизводим полученный новый куб на RGB-мониторе. Укажите, какие цвета будут иметь восемь вершин куба на экране.

(в) Что можно сказать о насыщенности цветов, которые расположены вдоль ребер цветового куба RGB?

6.9 **(а)** Опишите в общих чертах, как будут выглядеть на монохромном мониторе CMY-компоненты изображения из задачи 6.6.

(б) Опишите изображение, которое получится, если подать эти CMY-компоненты соответственно на красный, зеленый и синий входные каналы цветного монитора.

★**6.10** Выведите выражение (6.5-6) для функции преобразования интенсивности в CMY-пространстве из выражения (6.5-5) для соответствующей функции в RGB-пространстве.

6.11 Рассмотрите полный набор из 216 фиксированных цветов, представленный на рис. 6.10(*а*). Присвойте каждой ячейке номера (строки и столбца) согласно ее местоположению, так чтобы левой верхней ячейке отвечала пара (1,1), а нижней правой — (12,18). В какой ячейке будет находиться

(а) наиболее насыщенный красный цвет,

(б) наиболее насыщенный желтый цвет?

★**6.12** Опишите в общих чертах, как будут выглядеть на монохромном мониторе HSI-компоненты изображения из задачи 6.6.

6.13 Предложите метод получения цветного изображения, сходного с изображением видимой части электромагнитного спектра на рис. 6.2. Заметим, что диапазон начинается с темно-фиолетового цвета в левой части и заканчивается чистым красным цветам справа. (*Указание*: используйте цветовую модель HSI.)

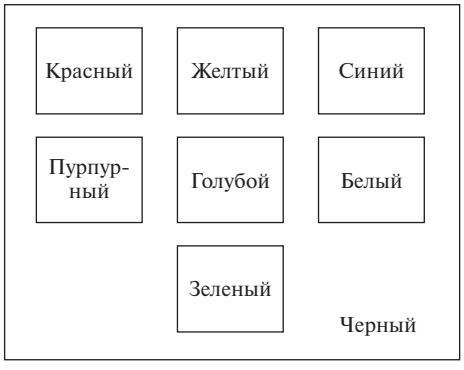
★**6.14** Предложите метод формирования цветного варианта изображения, представленного на рис. 6.13(*в*) в виде диаграммы. Ответ дайте в виде блок-схемы. Считайте значение интенсивности постоянным и заданным. (*Указание*: используйте цветовую модель HSI.)

6.15 Рассмотрите следующее изображение, составленное из сплошных цветных квадратов. Используйте при решении задачи яркостную шкалу, состоящую из восьми градаций от 0 до 7, в которой 0 соответствует черному, а 7 — белому. Пусть изображение преобразуется в цветовое пространство HSI. Отвечая на поставленные ниже вопросы, используйте при описании яркости конкретные числа оттенков серого, если использование чисел имеет смысл. В противном случае пользуйтесь выражениями типа «равный по яркости», «более светлый» или «более темный». Если невозможно описать яркость ни одним из указанных способов, то объясните почему.

(а)Опишите изображение цветового тона.

(б) Опишите изображение насыщенности.

(в) Опишите изображение интенсивности.

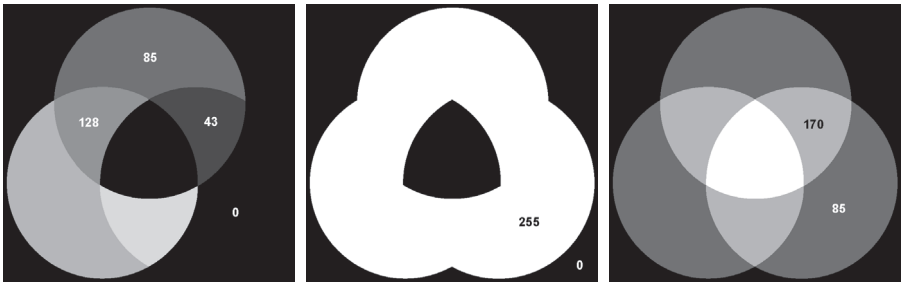


6.16 Следующие 8-битовые изображения суть (слева направо) изображения *H*-, *S*- и *I*-компонент с рис. 6.16. Значения яркости компонент указаны числами на рисунках. Ответьте на поставленные ниже вопросы и обоснуйте Ваш ответ в каждом отдельном случае. Если исходя из имеющейся информации дать ответ на вопрос невозможно, то сформулируйте точно причину этого.

★ **(а)** Найдите значения яркости для всех областей на изображении цветового тона.

(б) Найдите значения яркости для всех областей на изображении насыщенности.

(в) Найдите значения яркости для всех областей на изображении интенсивности.



(а) (б) (в)

6.17 Следующие вопросы связаны с рис. 6.27.

(а) Почему изображение на рис. 6.27(*е*) окрашено преимущественно в красные тона?

(б) Предложите способ, позволяющий в автоматическом режиме окрасить воду на рис. 6.27 в ярко-синий цвет.

(в) Предложите способ, позволяющий в автоматическом режиме окрасить на изображении объекты преимущественно искусственного происхождения в ярко-красный цвет. (*Указание*: воспользуйтесь изображением на рис. 6.27(*е*).)

★**6.18** Покажите, что компонента насыщенности для цветового дополнения некоторого цветного изображения не может быть вычислена только на основе компоненты насыщенности исходного изображения.

6.19 Объясните вид представленной на рис. 6.33(*б*) функции преобразования для компоненты цветового тона, которая используется при вычислении приближенного цветового дополнения в цветовой модели HSI.

★**6.20** Выведите функции преобразования, которые нужно использовать в модели CMY для получения цветового дополнения.

6.21 Опишите общий вид функций преобразования в цветовом пространстве RGB для устранения излишнего контраста на изображениях.

★**6.22** Предположим, что монитор и принтер некоторой системы обработки изображений имеют неправильную калибровку. Изображение, цвета которого на мониторе кажутся сбалансированными, при печати приобретает голубой оттенок. Опишите общие преобразования, которые могут устранить этот дисбаланс.

6.23 Вычислите цветовые координаты в модели $L^*a^*b^*$ для изображения из задачи 6.6, полагая, что

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0{,}588 & 0{,}179 & 0{,}183 \\ 0{,}29 & 0{,}606 & 0{,}105 \\ 0 & 0{,}068 & 1{,}021 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}.$$

Это матричное равенство определяет значение цветовых координат света, испускаемого цветными TV-люминофорами стандарта NTSC (National Television Standards Committee — Национальный комитет по телевизионным стандартам) при освещении стандартным источником D65 [Benson, 1985].

★**6.24** Как бы Вы реализовали для цветных изображений метод, аналогичный методу приведения (задания) гистограммы для полутоновых изображений из раздела 3.3.2?

6.25 Рассмотрим следующее цветное RGB-изображение размерами 1000×1000, состоящее из квадратов полностью насыщенных красного, зеленого и синего цветов, причем каждый из цветов имеет максимальную интенсивность (скажем, (1, 0, 0) для красного цвета). Данное изображение преобразовано в систему HSI.

(а) Опишите вид каждой компоненты HSI-изображения.

(б) Компонента насыщенности HSI-изображения сглажена при помощи усредняющей маски размерами 250×250. Опишите вид результата (можете игнорировать эффект границ изображения при фильтрации).

(в) Повторите (б) для компоненты цветового тона.

Зеленый	Красный
Синий	Зеленый

6.26 Покажите, что выражение (6.7-2) сводится к выражению (6.7-1), если $\mathbf{C} = \mathbf{I}$, т. е. является единичной матрицей.

6.27 ★ **(а)** В связи с рассмотрением в разделе 6.7.2 сформулируйте (в виде блок-схемы) алгоритм для определения того, находится ли цветовой вектор (точка) **z** внутри куба с центром в точке **a** и длиной стороны W. Алгоритм не должен включать вычисление расстояний.

(б) Если стороны куба направлены вдоль координатных осей, то в этом случае алгоритм может быть реализован на основе обработки отдельных компонент в цветовом пространстве. Как бы Вы это сделали?

6.28 Опишите поверхность в RGB-пространстве, которая задается уравнением

$$D(\mathbf{z},\mathbf{a}) = \left[(\mathbf{z}-\mathbf{a})^T \mathbf{C}^{-1}(\mathbf{z}-\mathbf{a})\right]^{\frac{1}{2}} = D_0,$$

где D_0 — некоторая ненулевая постоянная. Считайте, что $\mathbf{a} = 0$ и

$$\mathbf{C} = \begin{bmatrix} 8 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

6.29 Данная задача связана с материалом раздела 6.7.3. Может показаться, что при определении градиента RGB-изображения в точке (x, y) естественно было бы поступить следующим образом: вычислить *вектор* градиента (см. раздел 3.6.4) от каждой компоненты изображения и определить вектор градиента цветного изображения как векторную сумму полученных градиентов трех отдельных компонент. К сожалению, такой метод может временами давать ошибочные результаты. А именно цветное изображение с четко определенными контурами может в случае использования этого метода иметь нулевой градиент. Приведите соответствующий пример. (*Указание*: для простоты считайте, что одна из цветовых компонент изображения имеет постоянное значение.)

ГЛАВА 7

ВЕЙВЛЕТЫ И КРАТНОМАСШТАБНАЯ ОБРАБОТКА

Контролер все это время внимательно ее разглядывал — сначала в телескоп, потом в микроскоп и, наконец, в театральный бинокль.

Льюис Кэррол
«Алиса в Зазеркалье»
(пер. Н. Демуровой)

Введение

Хотя преобразование Фурье является основой трансформационных методов обработки изображений еще с конца 1950-х годов, применение более современного преобразования, называемого *вейвлет-преобразованием*, упрощает сжатие, передачу и анализ многих изображений. В отличие от преобразования Фурье, базисными функциями которого являются гармонические функции, вейвлет-преобразования основаны на разложении по малым волнам, называемым *вейвлетами*[1], изменяющейся частоты и ограниченным во времени (в пространстве). Такое разложение для изображения можно сравнить с нотной записью музыкального произведения, которая указывает музыканту не только какую ноту (частоту) взять, но и в какой момент это следует сделать. В противоположность этому преобразование Фурье содержит только частотную информацию (ноты); временна́я информация теряется в процессе преобразования.

В 1987 г. впервые было продемонстрировано [Mallat, 1987], что вейвлеты могут быть положены в основу нового мощного метода обработки и анализа сигналов, получившего название *кратномасштабный* анализ. Эта теория сводит воедино методы из разных областей, такие как субполосное кодирование из теории обработки сигналов, квадратурную зеркальную фильтрацию из теории распознавания речи и пирамидальную обработку изображений. Как следует из ее названия, кратномасштабная теория имеет дело с представлением и анализом сигналов (или изображений) в различных масштабах, т. е. при раз-

[1] Термин *вейвлет* (от английского *wavelet* — «маленькая, небольшая в смысле продолжительности волна»), утвердился в русской литературе, относящейся к областям обработки информации, сигналов и изображений. В математической литературе получил распространение предложенный К. И. Осколковым термин *всплеск*. — *Прим. перев.*

личных разрешениях[2]. Преимущество такого подхода очевидно — характерные детали, которые могут оставаться незаметными при одном разрешении, легко могут быть обнаружены при другом. Хотя до конца 1980-х годов интерес научного сообщества к кратномасштабной теории был весьма ограничен, в настоящий момент уже трудно уследить за всеми статьями, диссертациями и книгами, посвященными данному предмету.

Настоящая глава посвящена изучению преобразований, основанных на вейвлетах, с точки зрения кратномасштабного анализа. Хотя такие преобразования могут быть введены и другими способами, такой подход упрощает как математическую, так и физическую трактовку. Мы начнем с краткого рассмотрения тех методов обработки изображений, которые в числе других предопределили появление кратномасштабной теории. Наша цель состоит в том, чтобы привести основные концепции теории в контексте обработки изображений, а заодно дать небольшой исторический обзор методов и их приложений. Бóльшая часть главы посвящена разработке и использованию дискретного вейвлет-преобразования. Полезность данного преобразования демонстрируется на разнообразных примерах, от кодирования изображений до устранения шума и обнаружения контуров. В следующей главе будет рассматриваться применение вейвлетов для сжатия изображений — приложение, в котором они нашли широкое применение.

7.1. Предпосылки

Когда мы смотрим на изображение, мы обычно видим связанные области одинаковой структуры и яркости, которые, объединяясь, образуют объекты на изображении. Если эти объекты имеют маленький размер или низкий контраст, то мы, как правило, изучаем их при высоком разрешении; если же они имеют большой размер или высокий контраст, то вполне достаточно и поверхностного осмотра. Когда на изображении одновременно присутствуют как большие, так и маленькие объекты (или как низкоконтрастные, так и высококонтрастные объекты), то полезным может оказаться анализ такого изображения в разных масштабах (при различных разрешениях). Это и есть основная мотивировка для использования кратномасштабной обработки.

С математической точки зрения изображение представляет собой двумерный массив значений яркости с локально зависимыми статистиками, что обусловлено различным сочетанием характерных резких деталей, таких как контуры и контрастные однородные области. Как демонстрирует изображение на рис. 7.1, которое будет еще неоднократно рассматриваться в этом разделе, локальные гистограммы могут значительно изменяться при переходе от одной

[2] Следуя установившейся традиции, для перевода английского *multiresolution* («множественное разрешение») мы везде используем прилагательное *кратномасштабный*. Следует иметь в виду, что в действительности разные масштабы (разрешения), о которых идет речь, являются кратными по отношению друг к другу, т. е. связаны целой степенью некоторого масштабного фактора, который в большинстве случаев равен 2. — *Прим. перев.*

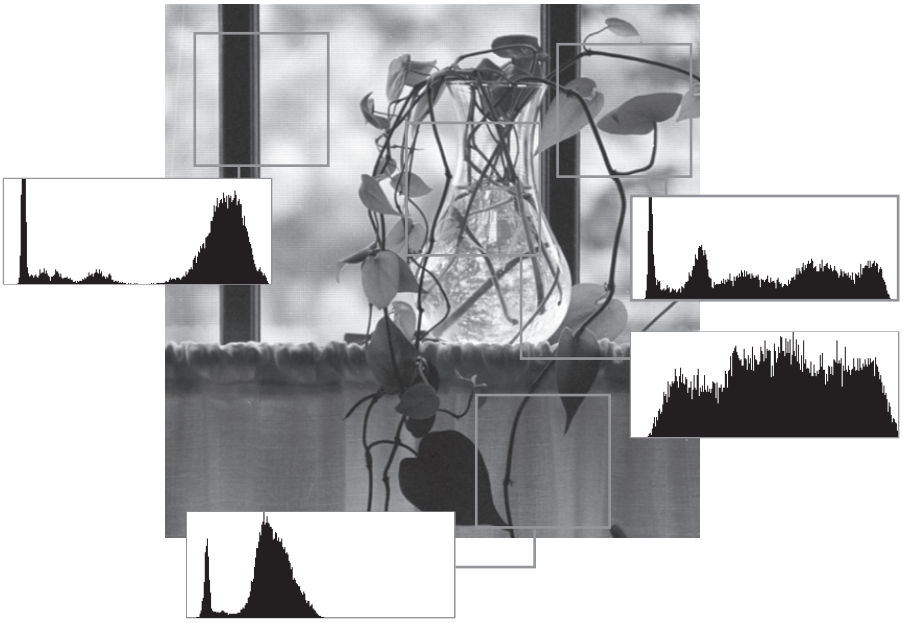


Рис. 7.1. Обычное изображение и локальные гистограммы отдельных его фрагментов

части изображения к другой, делая задачу разработки статистической модели по фрагменту изображения трудной или даже неразрешимой проблемой.

> Локальные гистограммы суть гистограммы значений пикселей в некоторой окрестности (см. раздел 3.3.3).

7.1.1. Пирамиды изображений

Мощной, но концептуально простой структурой для представления изображений в более чем одном масштабе является *пирамида изображений* [Burt, Adelson, 1983]. Разработанная первоначально для применения в задачах машинного зрения и сжатия изображений, пирамида изображений представляет собой набор изображений в уменьшающемся масштабе, организованный в форме пирамиды. Как видно из рис. 7.2(*a*), основу пирамиды составляет подлежащее обработке изображение высокого разрешения; вершина пирамиды состоит из приближения низкого разрешения. По мере движения вверх по пирамиде размеры и разрешение (масштаб) уменьшаются. Если нижний уровень J имеет размеры $2^J \times 2^J$ или $N \times N$, где $J = \log_2 N$, и верхний уровень 0 имеет размеры 1×1, то промежуточный уровень j имеет размеры $2^j \times 2^j$, где $0 \le j \le J$. Пирамида, показанная на рис. 7.2(*a*), составлена из $J + 1$ уровней от $2^J \times 2^J$ до $2^0 \times 2^0$, однако в большинстве случаев пирамида усекается до $P + 1$ уровней, где $j = J - P, \ldots, J - 2, J - 1, J$ и $1 \le P \le J$. Таким образом, мы обычно ограничиваемся P приближениями исходного изображения в уменьшенном масштабе; приближение размерами 1×1, т. е. одноточечное, не слишком информативно для описания исходного изображе-

а
б

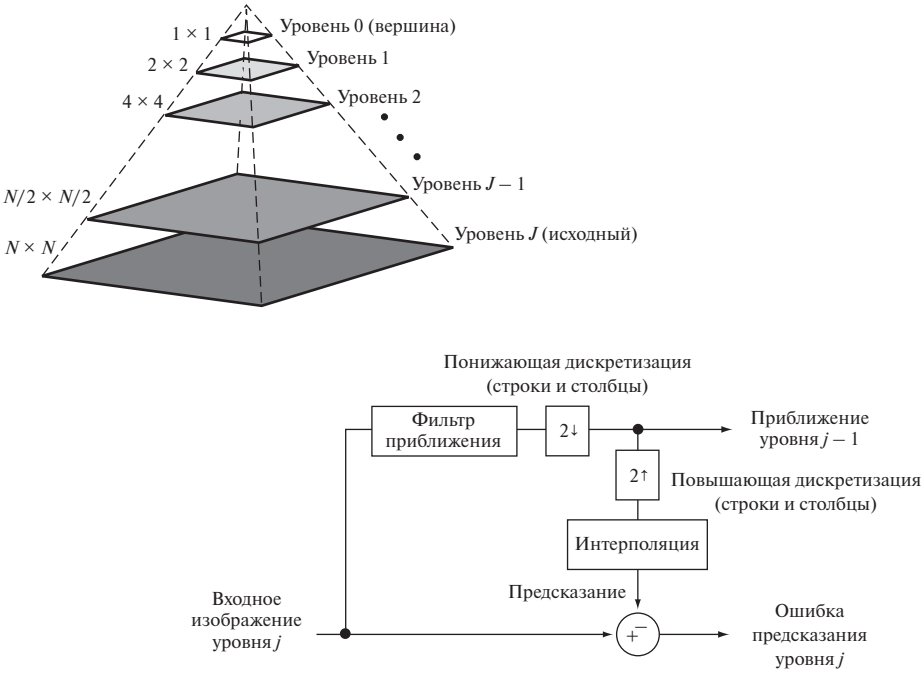


Рис. 7.2. (*а*) Пирамида изображения. (*б*) Блок-схема простой системы формирования пирамид приближения и ошибок предсказания

ния размерами, например, 512×512 элементов. Полное число пикселей на $P + 1$ уровнях пирамиды для $P > 0$ составляет

$$N^2\left(1+\frac{1}{4^1}+\frac{1}{4^2}+\ldots+\frac{1}{4^P}\right)\leq\frac{4}{3}N^2.$$

На рис. 7.2(*б*) показана простая блок-схема для построения двух тесно связанных между собой пирамид изображений. Выход *приближение* $(j - 1)$*-го уровня* формирует изображения, необходимые для построения *пирамиды приближений* (как описано в предыдущем параграфе), в то время как выход *ошибки предсказания уровня j* используется для формирования дополнительной *пирамиды ошибок предсказания* (или просто *пирамиды ошибок*). В отличие от пирамид приближений, пирамиды ошибок содержат только одно приближение входного изображения с уменьшенным масштабом (на верхнем уровне пирамиды $J - P$). Все остальные уровни содержат ошибки предсказаний, в которых уровень *j ошибки предсказаний* (для $j - P + 1 \leq j \leq J$) определяется как разность между уровнем *j* приближения (на входе в блок-схему) и оценкой приближения уровня *j*, основанной на приближении уровня $j - 1$ (выход приближения на блок-схеме).

В общем, ошибка предсказания может быть определена как разность между изображением и его предсказанной версией. Как будет показано в разделе 8.2.9, ошибки предсказания могут кодироваться более эффективно, чем двумерные массивы яркостей.

Как показано на рис. 7.2(*б*), пирамиды приближения и ошибок вычисляются итеративным образом. Перед началом первой итерации изображение, которое должно быть представлено в пирамидальной форме, помещается на уровень J пирамиды приближений. Затем следующая трехшаговая процедура выполняется P раз — для уровней $j = J - 1, J - 2,..., J - P + 1$ (именно в таком порядке):

1. Вычислить приближение уменьшенного масштаба *входного изображения уровня j* (вход показан с левой стороны блок-схемы на рис. 7.2(*б*)). Этот шаг включает фильтрацию входного изображения и понижение (частоты) дискретизации результата фильтрации с фактором 2. (Обе эти операции описаны в следующем параграфе.) Поместить результат приближения на уровень $j - 1$ пирамиды приближений.
2. Сформировать предсказание *входного изображения уровня j* на основе приближения уменьшенного масштаба, полученного на шаге 1. Это осуществляется повышением (частоты) дискретизации[3] и интерполяцией (см. следующий параграф) полученного ранее приближения. Результирующее изображение-предсказание будет иметь тот же размер, что и *входное изображение уровня j*.
3. Вычислить разность между изображением-предсказанием, полученным на шаге 2, и входным изображением шага 1. Поместить данный результат на уровень j пирамиды ошибок.

После выполнения P итераций (т. е. после итерации с номером $j = J - P + 1$) выход уровня приближения $J - P$ помещается в пирамиду ошибок на уровень $J - P$. Если пирамида ошибок не нужна, то эту операцию наряду с шагами 2 и 3, а также блоками повышения дискретизации, интерполяции и разности на рис. 7.2(*б*) можно опустить.

В систему со схемой на рис. 7.2(*б*) могут быть встроены самые разнообразные фильтры приближения и интерполяции. Обычно фильтрация осуществляется в пространственной области (см. раздел 3.4). Среди полезных методов построения фильтров приближения следует отметить среднее по окрестности (см. раздел 3.5.1), которое дает в результате *пирамиду усреднения*, низкочастотный гауссов фильтр (см. раздел 4.7.4 и 4.8.3), дающий *гауссову пирамиду*, а также отбрасывание промежуточных значений, что дает *пирамиду прореживания*. В интерполяционном фильтре могут быть использованы любые из методов, рассмотренных в разделе 2.4.4, включая интерполяцию по ближайшему соседу, билинейную или бикубическую интерполяцию. Отметим еще раз, что блоки повышения и понижения дискретизация на рис. 7.2(*б*) используются для двукратного увеличения и уменьшения пространственного разрешения при вычислении изображений приближения или предсказания. В случае одномерной последовательности отсчетов $f(n)$ процедуру *повышения дискретизации* для очередного значения n можно записать:

$$f_{2\uparrow}(n) = \begin{cases} f(n/2) & \text{если } n \text{ четное,} \\ 0 & \text{если } n \text{ нечетное,} \end{cases} \qquad (7.1\text{-}1)$$

[3] Понижение (*downsampling* или *subsampling*) и повышение (*upsampling*) частоты дискретизации также называют прореживающей и сгущающей выборками соответственно. — *Прим. перев.*

где, как видно из нижнего индекса, повышающая дискретизация имеет коэффициент 2. Дополнительная операция *понижения дискретизации* с коэффициентом 2 определяется как

$$f_{2\downarrow}(n) = f(2n). \tag{7.1-2}$$

В этой главе мы будем работать как с непрерывными, так и с дискретными функциями и переменными. За исключением двумерного изображения $f(x, y)$, а также если не указано иное, то x, y, z, ... означают непрерывные переменные, а i, j, k, l, m, n, ... — дискретные переменные.

Процедура повышения дискретизации может быть представлена как вставление нулей после каждого отсчета последовательности (сгущение); процедура понижения дискретизации — как удаление каждого второго отсчета (прореживание). Блоки повышения и понижения дискретизации на схеме рис. 7.2(*б*), которые отмечены как 2↑ и 2↓ соответственно, означают, что осуществляется повышение или понижение дискретизации как по строкам, так и по столбцам двумерного сигнала, подающегося на вход. Как и двумерное дискретное преобразование Фурье (см. раздел 4.11.1), двумерные операции повышения и понижения дискретизации могут быть реализованы как последовательные шаги одномерных операций, определенных уравнениями (7.1-1) и (7.1-2).

Пример 7.1. Пирамиды приближения и ошибок предсказания.

■ На рис. 7.3 представлены как пирамида приближения, так и пирамида ошибок для изображения с вазой на рис. 7.1. Для получения четырехуровневой пирамиды приближения использовался низкочастотный гауссов фильтр (см. раздел 4.7.4). Как можно видеть, полученная пирамида состоит из исходного изображения размерами 512×512 пикселей (в основании пирамиды) и трех приближений уменьшенных масштабов размерами 256×256, 128×128 и 64×64. Таким образом, P = 3 и пирамида была усечена до четырех уровней — с номерами 9, 8, 7 и 6 — из возможных $\log_2(512) + 1$ или 10 уровней. Обратим внимание на уменьшение детализации, которое сопровождает уменьшение масштаба изображений в пирамиде. Приближение 6-го уровня (т. е. размерами 64×64), например, позволяет обнаружить местонахождение оконного переплета (оконной рамы), но не стебли растения. Вообще уровни пирамиды мелких масштабов могут быть использованы для анализа структур большого размера или содержания изображения в целом; изображения крупных масштабов подходят для анализа особенностей отдельных объектов. Такая стратегия анализа, состоящая в постепенном переходе от грубого просмотра к точному рассмотрению, особенно полезна при распознавании образов.

Пирамиду приближений (*а*) на рис. 7.3 часто называют гауссовой пирамидой, поскольку для ее построения использовался гауссов фильтр. Пирамиду ошибок предсказания (*б*), в свою очередь, называют пирамидой лапласианов; отметим похожесть данных изображений на изображения, подвергнутые фильтрации лапласианом, в главе 3.

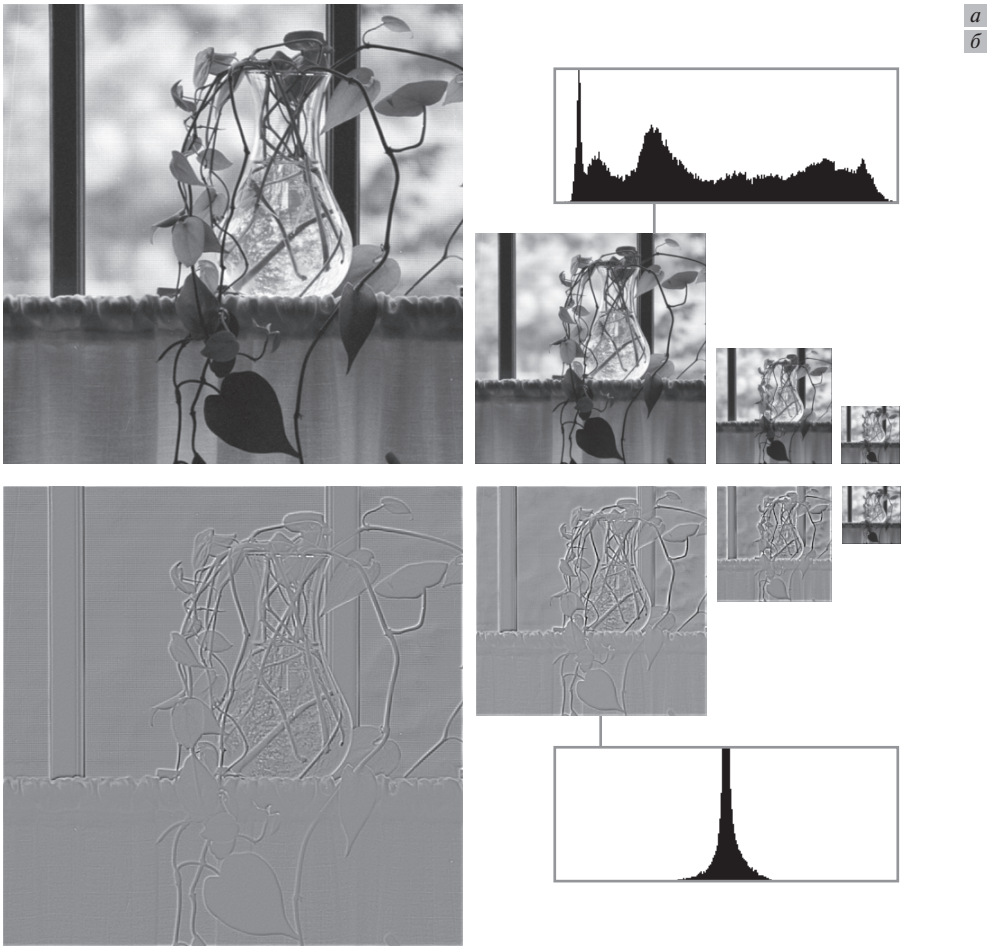


Рис. 7.3. Две пирамиды изображений и их гистограммы: (*а*) пирамида приближений; (*б*) пирамида ошибок предсказания

Для получения пирамиды ошибок на рис. 7.3(*б*) использовался фильтр билинейной интерполяции. При отсутствии ошибок квантования, используя полученную пирамиду ошибок, можно построить являющуюся к ней дополнением пирамиду приближений, показанную на рис. 7.3(*а*). Чтобы осуществить это, необходимо начать с уровня 6 изображения приближений размерами 64×64 (единственное изображение приближений в пирамиде ошибок), предсказать уровень 7 приближения размерами 128×128 (при помощи повышения дискретизации и фильтрации) и прибавить ошибку предсказания уровня 7. Эту процедуру следует итеративно повторять до достижения исходного изображения размерами 512×512 пикселей. Заметим, что значения ошибок квантования сконцентрированы вблизи нуля, и соответствующая гистограмма изображения ошибок на рис. 7.3(*б*) имеет ярко выраженный пик в этой области; гистограмма изображения приближений на рис. 7.3(*а*) выглядит совершенно иначе. В отличие от изображений приближений, изображения ошибок предсказания могут подвергаться сжатию высокой степени путем присваивания кодов меньшей длины

наиболее вероятным значениям (см. метод неравномерного кодирования в разделе 8.2.1). В заключение отметим, что разности с предсказаниями на рис. 7.3(*б*) проградуированы таким образом, чтобы сделать заметными меньшие по величине ошибки предсказания; однако соответствующая гистограмма посчитана на основе результатов до градуировки, при этом значение 128 отвечает нулевой ошибке. ■

7.1.2. Субполосное кодирование

Другим важным методом, используемым при обработке изображений и связанным с кратномасштабной теорией, является *субполосное кодирование*. В субполосном кодировании изображение разлагается на несколько составляющих с ограниченным диапазоном частот, называемых *субполосы* или *субдиапазоны*. Разложение производится таким образом, что субполосы могут быть снова собраны воедино, что позволит точно (без искажений) восстановить исходное изображение. Поскольку разложение и восстановление выполняются при помощи цифровых фильтров, мы начнем рассмотрение этого вопроса с небольшого введения в *цифровую обработку сигналов* (ЦОС) и *цифровую фильтрацию сигналов*.

Рассмотрим простой *цифровой фильтр* на рис. 7.4(*а*) и обратим внимание, что он составлен из трех основных компонентов — *блоков задержки*, *умножителей* и *сумматоров*. В верхнем ряду фильтра блоки задержки соединены последовательно и создают $K-1$ вариант задержанной (т. е. сдвинутой вправо) входной последовательности $f(n)$. Так например, задержанная на 2 последовательность $f(n-2)$ будет выглядеть

$$f(n-2)=\begin{cases}\vdots & \\ f(0) & \text{для } n=2 \\ f(1) & \text{для } n=2+1=3 \\ \vdots & \end{cases}.$$

> Термин «задержка» неявно выражает временну́ю зависимость элементов входной последовательности и отражает тот факт, что в цифровой фильтрации сигналов исходной информацией обычно является оцифрованный аналоговый сигнал.

Как показано серыми подписями на рис. 7.4(*а*), входная последовательность $f(n) = f(n-0)$ и $K-1$ задержанных последовательностей на выходах блоков задержки, обозначаемых $f(n-1)$, $f(n-2)$, ..., $f(n-K+1)$, умножаются соответственно на $h(0)$, $h(1)$, ..., $h(K-1)$ и суммируются, давая на выходе фильтрованную последовательность

$$\hat{f}(n)=\sum_{k=-\infty}^{\infty}h(k)f(n-k)=f(n)\bigstar h(n), \qquad (7.1\text{-}3)$$

где ★ означает свертку. Отметим, что за исключением замены переменных, уравнение (7.1-3) эквивалентно дискретной свертке, определяемой уравнением (4.4-10) в главе 4. K постоянных множителей на рис. 7.4(*а*) и в уравнении (7.1-3) называются *коэффициентами фильтра*. Каждый коэффициент задает *элемент*

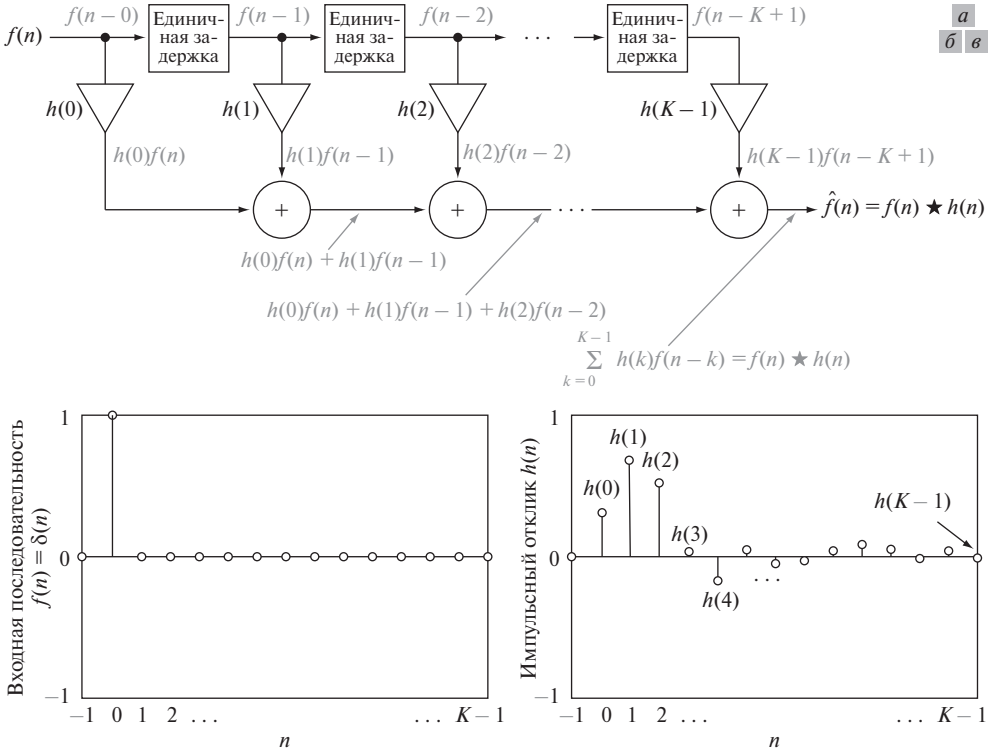


Рис. 7.4. (*а*) Блок-схема цифрового фильтра; (*б*) последовательность, содержащая единичный дискретный импульс; (*в*) импульсный отклик фильтра

фильтра, необходимый для вычисления одного члена суммы в уравнении (7.1-3). Такой фильтр называют *фильтром порядка* $K-1$.

> Если коэффициенты в фильтре на рис. 7.4(*а*) проиндексированы значениями от 0 до $K-1$ (как на рисунке), то пределы суммирования в уравнении (7.1-3) могут быть сокращены — от 0 до $K-1$ (аналогично уравнению (4.4-10)).

Если на вход фильтра рис. 7.4(*а*) подается единичный дискретный импульс (рис. 7.4(*б*) и раздел 4.2.3), уравнение (7.1-3) примет вид

$$\hat{f}(n) = \sum_{k=-\infty}^{\infty} h(k)\delta(n-k) = h(n). \qquad (7.1\text{-}4)$$

Таким образом, подставляя $\delta(n)$ в качестве входной последовательности $f(n)$ в уравнение (7.1-3) и воспользовавшись фильтрующим свойством единичного дискретного импульса, как определено в уравнении (4.2-13), мы обнаружим, что импульсным откликом фильтра на рис. 7.4(*а*) будет K-элементная последовательность коэффициентов, задающих фильтр. Физически единичный импульс сдвигается слева направо через элементы верхнего ряда фильтра (с единичной задержкой при переходе к каждому очередному элементу), давая в качестве выхода значение коэффициента в позиции, равной величине задержки импульса. Поскольку имеется всего K коэффициентов, то отклик фильтра будет иметь

длину K; такой фильтр называется фильтром с *конечной импульсной характеристикой* (КИХ-фильтром).

На рис. 7.5 показаны импульсные отклики шести функционально связанных фильтров. Фильтр $h_2(n)$ на рис. 7.5(*б*) является *знако-инвертирующим* (т. е. зеркально отражающим относительно горизонтальной оси) вариантом $h_1(n)$ на рис. 7.5(*а*), т. е.

$$h_2(n) = -h_1(n). \tag{7.1-5}$$

В оставшейся части раздела термин «фильтр $h(n)$» будет использоваться для обозначения фильтра, импульсный отклик которого есть $h(n)$.

Фильтры $h_3(n)$ и $h_4(n)$ на рис. 7.5(*в*) и (*г*) являются *последовательно-инвертирующими* вариантами $h_1(n)$:

$$h_3(n) = h_1(-n), \tag{7.1-6}$$

$$h_4(n) = h_1(K-1-n). \tag{7.1-7}$$

Последовательно-инвертирующий фильтр часто называют время-инвертирующим, когда входная последовательность зависима от времени.

Фильтр $h_3(n)$ является отражением $h_1(n)$ относительно вертикальной оси; фильтр $h_3(n)$ является отраженным и смещенным (т. е. сдвинутым) вариантом $h_1(n)$. Если пренебречь смещением, отклики обоих фильтров идентичны. Фильтр $h_5(n)$ на рис. 7.5(*д*), который определяется как

$$h_5(n) = (-1)^n h_1(n), \tag{7.1-8}$$

называется *модулирующим* вариантом $h_1(n)$. Поскольку данная модуляция меняет знаки всех коэффициентов с нечетными индексами (т. е. коэффициентов, для которых n на рис. 7.5(*а*) является нечетным), $h_5(1) = -h_1(1)$ и $h_5(3) = -h_1(3)$, но $h_5(0) = h_1(0)$ и $h_5(2) = h_1(2)$. Наконец, последовательность, представленная на рис. 7.5(*е*), является инвертированным по времени, смещенным и модулированным вариантом $h_1(1)$:

$$h_6(n) = (-1)^n h_1(K-1-n). \tag{7.1-9}$$

Данная последовательность включена в рассмотрение, чтобы проиллюстрировать, что инвертирование по знаку, по времени и модуляция иногда комбинируются при задании взаимосвязи между фильтрами.

После такого краткого введения в цифровую фильтрацию сигналов рассмотрим двухканальную систему субполосного кодирования и декодирования, блок-схема которой показана на рис. 7.6(*а*). Как видно из рисунка, система состоит из двух *банков фильтров*[4], каждый содержащий два КИХ-фильтра показанного на рис. 7.4(*а*) типа. Эти КИХ-фильтры изображены на рис. 7.6(*а*) в виде отдельных блоков, внутри каждого из которых указан импульсный отклик данного фильтра и символ свертки ★. Банк фильтров *анализа*, который содержит фильтры

[4] *Банк фильтров* — совокупность двух или более фильтров.

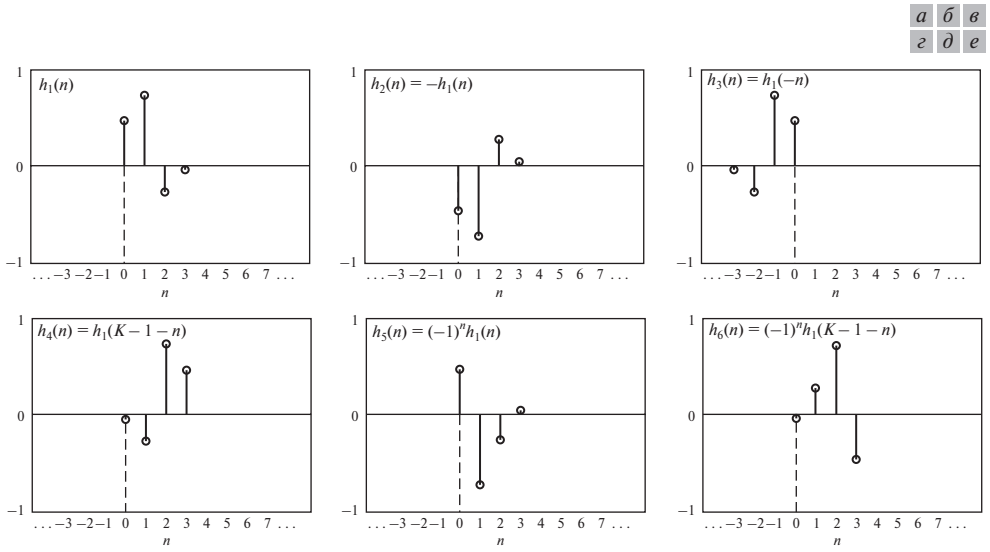


Рис. 7.5. Импульсные отклики шести функционально взаимосвязанных фильтров: (*а*) опорный отклик; (*б*) знако-инвертированный отклик; (*в*) и (*г*) последовательно-инвертированные отклики (различающиеся вносимой задержкой); (*д*) модулированный отклик; (*е*) инвертированный по времени, смещенный и модулированный отклик

$h_0(n)$ и $h_1(n)$, используется для разбиения входной последовательности $f(n)$ на две последовательности половинной длины — $f_{lp}(n)$ и $f_{hp}(n)$, т. е. *поддиапазоны*, вместе представляющие вход. Отметим, что фильтры $h_0(n)$ и $h_1(n)$ являются полосовыми фильтрами (на половину диапазона каждый), идеализированные передаточные характеристики H_0 и H_1 которых показаны на рис. 7.6(*б*). Фильтр $h_0(n)$ представляет собой низкочастотный фильтр поддиапазона $f_{lp}(n)$, сигнал на выходе которого является (низкочастотным) *приближением* исходного сигнала $f(n)$; фильтр $h_1(n)$ представляет собой высокочастотный фильтр поддиапазона $f_{hp}(n)$, сигнал на выходе которого является высокочастотной или *детальной* частью исходного сигнала $f(n)$. Фильтры банка *синтеза* $g_0(n)$ и $g_1(n)$ объединяют данные в поддиапазонах $f_{lp}(n)$ и $f_{hp}(n)$, формируя $\hat{f}(n)$. Задача при построении субполосного кодирования состоит в выборе таких $h_0(n)$, $h_1(n)$, $g_0(n)$ и $g_1(n)$, чтобы $\hat{f}(n) = f(n)$. Это означает, что вход системы субполосного кодирования и выход системы субполосного декодирования должны быть идентичными. Когда это достигается, то говорят, что в результирующей системе используются *фильтры точного восстановления*.

Имеется много двухканальных банков с КИХ-фильтрами точного восстановления с действительными коэффициентами, описанных в литературе по банкам фильтров. Во всех из них фильтры синтеза являются модулированными вариантами фильтров анализа, причем знак одного (и только одного) из них меняется на противоположный. Для точного восстановления импульсные отклики фильтров синтеза и анализа должны быть связаны одной из следующих зависимостей:

$$\begin{aligned} g_0(n) &= (-1)^n h_1(n), \\ g_1(n) &= (-1)^{n+1} h_0(n) \end{aligned} \qquad (7.1\text{-}10)$$

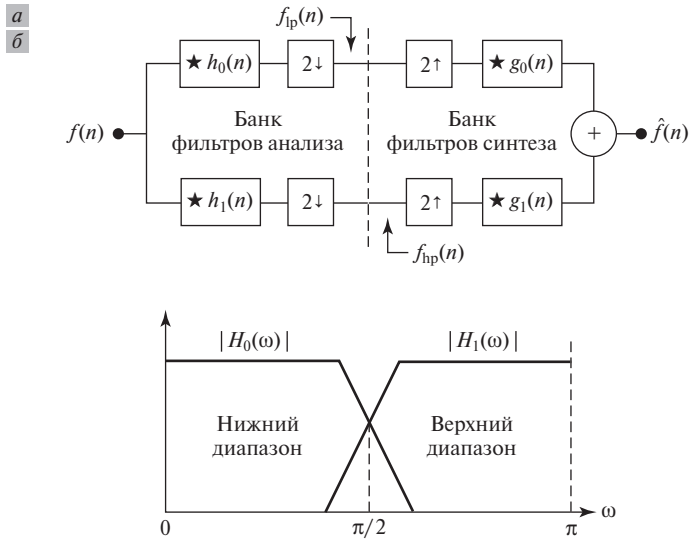


Рис. 7.6. (*а*) Двухканальный банк фильтров системы субполосного кодирования и декодирования и (*б*) его характеристики разделения диапазона

Под действительными коэффициентами понимается, что коэффициенты фильтра являются действительными (не комплексными) значениями.

или

$$g_0(n) = (-1)^{n+1} h_1(n), \quad g_1(n) = (-1)^n h_0(n). \tag{7.1-11}$$

Уравнения (7.1-10)—(7.1-14) подробно рассматриваются в литературе по банкам фильтров (см., например, [Vetterli, Kovacevic, 1995]).

Фильтры $h_0(n)$, $h_1(n)$, $g_0(n)$ и $g_1(n)$ в уравнениях (7.1-10) и (7.1-11) называют *перекрестно модулированными* (*кросс-модулированными*) фильтрами, поскольку диагонально противоположные фильтры в блок-схеме на рис. 7.6(*а*) связаны модуляцией (и инвертированием знака, когда коэффициент модуляции есть $-(-1)^n$ или $(-1)^{n+1}$). Более того, можно показать, что они удовлетворяют следующему условию *биортогональности*:

$$\left\langle h_i(2n-k), g_j(k) \right\rangle = \delta(i-j)\delta(n), \quad i,j = \{0,1\}. \tag{7.1-12}$$

Здесь $\langle h_i(2n-k), g_i(k)\rangle$ означает скалярное произведение $h_i(2n-k)$ и $g_i(k)$[5]. Когда $i = j$, скалярное произведение равно единичной дискретной импульсной функции $\delta(n)$; когда $i \neq j$, то произведение равно 0. Биортогональность будет еще раз рассмотрена в разделе 7.2.1.

[5] Скалярное произведение последовательностей $f_1(n)$ и $f_2(n)$ определяется выражением $\langle f_1, f_2\rangle = \sum_n f_1^*(n) f_2(n)$, где знак * означает комплексное сопряжение. Если последовательности $f_1(n)$ и $f_2(n)$ вещественные, то $\langle f_1, f_2\rangle = \langle f_2, f_1\rangle$.

Особый интерес в субполосном кодировании (и в разработке быстрого вейвлет-преобразования в разделе 7.4) представляют фильтры, которые продвигаются дальше биортогональности и требуют выполнения условия

$$\langle g_i(n), g_j(n+2m)\rangle = \delta(i-j)\delta(m), \quad i,j=\{0,1\}, \tag{7.1-13}$$

которое означает *ортонормированность* для банков фильтров точного восстановления. В дополнение к уравнению (7.1-13) можно показать, что ортонормированные фильтры должны удовлетворять следующим двум условиям:

$$\begin{aligned} g_1(n) &= (-1)^n g_0(K_{\text{четное}} - 1 - n), \\ h_i(n) &= g_i(K_{\text{четное}} - 1 - n), \qquad i=\{0,1\}, \end{aligned} \tag{7.1-14}$$

где нижний индекс в $K_{\text{четное}}$ использован для обозначения, что число коэффициентов фильтра должно быть кратно 2 (т. е. четным числом). Как видно из уравнения (7.1-14), фильтр синтеза g_1 связан с g_0 инвертированием времени и модуляцией. Кроме того, как h_0, так и h_1 являются соответственно время-инвертирующими вариантами g_0 и g_1. Тем самым, ортонормированные банки фильтров могут разрабатываться на основе импульсного отклика единственного фильтра, называемого *прототипом*; остальные фильтры могут быть вычислены исходя из импульсного отклика выбранного прототипа. Для биортогонального банка фильтров требуются два прототипа; остальные фильтры вычисляются при помощи уравнений (7.1-10) или (7.1-11). Формирование полезных прототипов фильтров, как ортонормированных, так и биортогональных, выходит за рамки данной главы. Мы просто используем фильтры, представленные в литературе, и даем ссылки для дальнейшего изучения.

Прежде чем завершить раздел примером двумерного субполосного кодирования, отметим, что одномерные ортонормированные и биортогональные фильтры могут использоваться для обработки изображений в качестве двумерных разделимых фильтров. Как можно увидеть из схемы на рис. 7.7, разделимые фильтры сначала используются в одном направлении (например вертикаль-

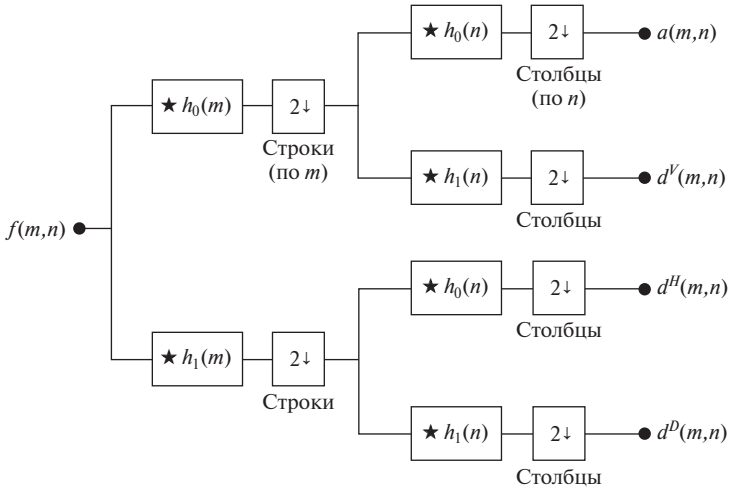


Рис. 7.7. Двумерный четырехканальный банк фильтров для субполосного кодирования изображений

ном), а затем в другом (горизонтальном) способом, изложенным в разделе 2.6.7. Более того, понижение дискретизации осуществляется в два этапа — после каждой из операций фильтрации — для сокращения общего количества вычислений. Получаемые на выходе результаты фильтрации называются субдиапазонами изображения. Субдиапазон $a(m,n)$ на рис. 7.7 называется субдиапазоном *приближения*, а субдиапазоны $d^V(m,n)$, $d^H(m,n)$ и $d^D(m,n)$ — субдиапазонами *вертикальных*, *горизонтальных* и *диагональных* деталей соответственно. Каждый из этих субдиапазонов в свою очередь может быть разделен на четыре меньших субдиапазона, которые могут быть разделены еще раз и т. д. — данное свойство будет рассмотрено подробно в разделе 7.4.

Пример 7.2. Четырехканальное субполосное кодирование изображения вазы на рис. 7.1.

■ На рис. 7.8 изображены импульсные характеристики четырех ортонормированных 8-элементных фильтров Добеши. Коэффициенты прототипа фильтра синтеза $g_0(n)$ для $0 \le n \le 7$ (на рис. 7.8(*в*)) представлены в табл. 7.1. [Daubechies, 1992]. Коэффициенты остальных ортонормированных фильтров могут быть вычислены с помощью (7.1-14). Заметим, что перекрестная модуляция фильтров анализа и синтеза видна прямо из рис. 7.8. С помощью вычислений нетрудно проверить, что фильтры являются и биортогональными (удовлетворяют (7.1-12)), и ортонормированными (удовлетворяют (7.1-13)). В результате 8-элементные фильтры Добеши на рис. 7.8 обеспечивают точное восстановление после разложения исходного сигнала.

Таблица 7.1. Коэффициенты прототипа фильтра синтеза $g_0(n)$ ортонормированных 8-элементных фильтров Добеши [Daubechies, 1992]

n	$g_0(n)$
0	0,23037781
1	0,71484657
2	0,63088076
3	0,02798376
4	−0,18703481
5	0,03084138
6	0,03288301
7	−0,01059740

На рис. 7.9 представлено четырехканальное разложение изображения с вазой на рис. 7.1 размерами 512×512 с использованием представленных на рис. 7.8 фильтров. Каждое из четырех изображений на этом рисунке представляет субдиапазон размерами 256×256. Расположение субдиапазонов, начиная от верхнего левого по часовой стрелке, следующее: субдиапазон приближения a, субдиапазон горизонтальных деталей d^H, субдиапазон диагональных деталей d^D и субдиапазон вертикальных деталей d^V. Значения яркости всех субдиапазонов, за исключением субдиапазона приближения на рис. 7.9(*а*), были подвергнуты градационному преобразованию для того, чтобы сделать их внутреннюю струк-

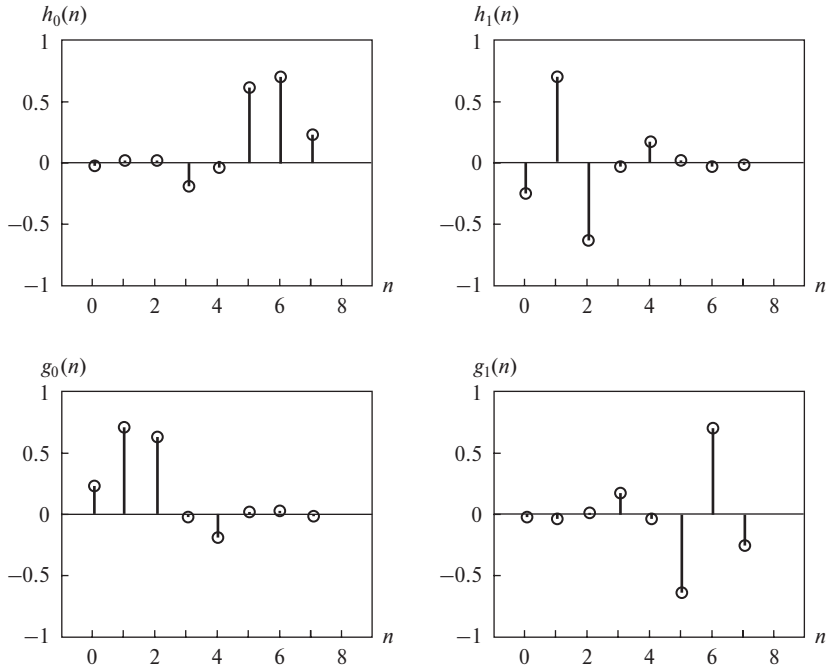


Рис. 7.8. Импульсные характеристики четырех 8-элементных ортонормированных фильтров Добеши. См. табл. 7.1 значений $g_0(n)$ для $0 \le n \le 7$

туру более заметной. Обратим внимание на артефакты характерной геометрической структуры, присутствующие на рис. 7.9(*б*) и (*в*) в субдиапазонах d^H и d^V. Появление волнообразных структур в области окна вызвано наличием едва заметных полос на исходном изображении на рис. 7.1 и наложением спектров в результате снижения разрешения. Несмотря на это, оригинальное изображение может быть восстановлено по субдиапазонам без искажений. Требуемые для этого фильтры синтеза $g_0(n)$ и $g_1(n)$ определяются исходя из табл. 7.1 и уравнения (7.1-14) и встраиваются в банк фильтров, который, грубо говоря, является зеркальным отражением системы на рис. 7.7. В этом новом банке фильтры $h_i(n)$, $i = \{0,1\}$, заменяются на отвечающие им фильтры $g_i(n)$, снижение разрешения заменяется на повышение разрешения и добавляется процедура суммирования. ■

По поводу наложения спектров см. раздел 4.5.4.

7.1.3. Преобразование Хаара

Третьим и последним используемым при обработке изображений методом, который связан с кратномасштабной теорией, является преобразование Хаара [Haar, 1910]. В контексте настоящей главы значимость этого преобразования обусловлена тем, что его базисные функции (определенные ниже) образуют первую и простейшую из известных систему ортонормированных вейвлетов. Эти вейвлеты будут использованы в ряде примеров далее.

а *б*
в *г*

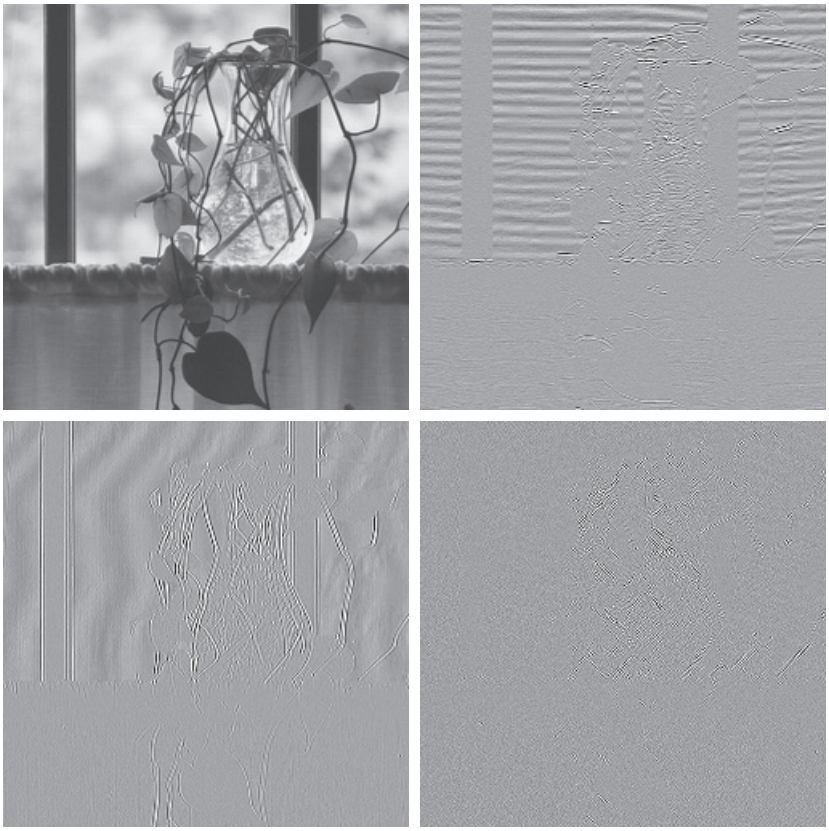


Рис. 7.9. Разложение изображения с вазой на рис. 7.1 при помощи системы субполосного кодирования (на рис. 7.7) на четыре субдиапазона: (*а*) приближения, (*б*) горизонтальных деталей, (*в*) вертикальных деталей, (*г*) диагональных деталей

Преобразование Хаара является разделимым и симметричным и в соответствии с обсуждением в разделе 2.6.7 может быть записано в матричном виде следующим образом:

$$\mathbf{T} = \mathbf{HFH}^T, \tag{7.1-15}$$

где **F** — матрица изображения, **H** — матрица преобразования, **T** — результат преобразования (каждая размерами $N \times N$). Верхний индекс T означает операцию транспонирования матрицы; транспонирование необходимо, поскольку матрица **H** не является симметричной (в уравнении (2.6-38) раздела 2.6.7 матрица преобразования предполагалась симметричной). Матрица преобразования Хаара **H** состоит из базисных функций Хаара $h_k(z)$. Эти функции определены на непрерывном замкнутом интервале $z \in [0,1]$ при $k = 0, 1, 2, ..., N-1$, где $N = 2^n$. Чтобы получить **H**, зададим целое k такое, что $k = 2^p + q - 1$, где $0 \le p \le n-1$, $q = 0$ или 1 при $p = 0$ и $1 \le q \le 2^p$, когда $p \ne 0$. Тогда *базисные функции Хаара* будут

$$h_0(z) = h_{00}(z) = \frac{1}{\sqrt{N}}, \quad z \in [0,1] \tag{7.1-16}$$

и
$$h_k(z) = h_{pq}(z) = \frac{2^{p/2}}{\sqrt{N}} \begin{cases} 1 & \text{при } (q-1)/2^p \le z < (q-0{,}5)/2^p; \\ -1 & \text{при } (q-0{,}5)/2^p \le z < q/2^p; \\ 0 & \text{в остальных случаях, } z \in [0,1]. \end{cases} \quad (7.1\text{-}17)$$

Строка с номером i матрицы преобразования Хаара $\mathbf{H}$ размерами $N \times N$ состоит из значений функции $h_i(z)$, взятых в точках $z = 0/N, 1/N, 2/N, \ldots, (N-1)/N$. При $N = 2$, например, первая строка матрицы 2×2 Хаара вычисляется как $h_0(z)$ при $z = 0/2, 1/2$. Согласно уравнению (7.1-16), $h_0(z)$ равно $1/\sqrt{2}$ независимо от z, так что первая строка $\mathbf{H}_2$ содержит два одинаковых элемента $1/\sqrt{2}$. Вторая строка получается вычислением $h_1(z)$ для $z = 0/2, 1/2$. Поскольку $k = 2^p + q - 1$, то для $k = 1$ имеем: $p = 0$ и $q = 1$. Таким образом, из уравнения (7.1-17) получаем: $h_1(0) = 2^0/\sqrt{2} = 1/\sqrt{2}$ и $h_1(1/2) = -2^0/\sqrt{2} = -1/\sqrt{2}$. Таким образом, матрица 2×2 преобразования Хаара равна

$$\mathbf{H}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \quad (7.1\text{-}18)$$

Если $N = 4$, индексы k, q, и p принимают значения

k	p	q
0	0	0
1	0	1
2	1	1
3	1	2

и 4×4 матрица преобразования $\mathbf{H}_4$ имеет вид

$$\mathbf{H}_4 = \frac{1}{\sqrt{4}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{bmatrix}. \quad (7.1\text{-}19)$$

Преобразование Хаара представляет интерес главным образом тем, что строки матрицы $\mathbf{H}_2$ могут быть использованы для формирования фильтров анализа $h_0(n)$ и $h_1(n)$ двухэлементного банка фильтров точного восстановления (см. предыдущий раздел), а также уточняющих последовательностей для масштабирующей функции и вейвлетов (определенных в разделах 7.2.2 и 7.2.3 соответственно) простейшего и старейшего вейвлет-преобразования (см. пример 7.10 в разделе 7.4). Вместо того, чтобы заканчивать раздел вычислением преобразования Хаара, мы завершим его примером, который иллюстрирует влияние методов разложения, разбиравшихся до настоящего момента, на методы, которые будут рассмотрены в оставшейся части главы.

Пример 7.3. Дискретное вейвлет-преобразование на основе функций Хаара.

■ На рис. 7.10(*a*) показано разложение изображения, представленного на рис. 7.1 и имеющего размеры 512×512 элементов, которое объединяет ключевые особен-

ности пирамидального кодирования, субполосного кодирования и преобразования Хаара (трех методов, рассмотренных к настоящему моменту). Данное представление, называемое дискретным вейвлет-преобразованием (и разрабатываемое ниже в настоящей главе), характеризуется следующими важными свойствами:

1. За исключением уменьшенного субизображения в левом верхнем углу рис. 7.10(*а*), гистограммы остальных субизображений весьма похожи и содержат много значений элементов вблизи нуля. Для наглядности все субизображения (за исключением левого верхнего) были подвергнуты градационному преобразованию, в результате чего нуль яркости был сдвинут в точку со значением 128 и все изображения выглядят серыми. Большое число элементов разложения со значениями вблизи нуля делает такие субизображения отличными кандидатами для сжатия (см. главу 8).
2. Субизображения на рис. 7.10(*а*) могут быть использованы для получения как грубых, так и точных приближений оригинального изображения на рис. 7.1. Это может осуществляться способом, аналогичным тому, при помощи которого уровни пирамиды ошибок предсказания на рис. 7.3(*б*) использовались для создания изображений приближений разных масштабов. Изображения на рис. 7.10(*б*)—(*г*) размерами 64×64, 128×128 и 256×256 соответственно были получены из субизображений на рис. 7.10(*а*). Возможно также и точное восстановление исходного изображения.
3. Подобно разложению при помощи субполосного кодирования на рис. 7.9, для получения субизображений на рис. 7.10(*а*) использовался простой банк КИХ-фильтров с вещественными коэффициентами, имеющий структуру блок-схемы на рис. 7.7. После получения четырех изображений субполос, подобным представленным на рис. 7.9, изображение приближения субполосы 256×256 элементов подверглось разложению (используя тот же самый банк фильтров) и было заменено четырьмя субполосами 128×128, а последнее полученное приближение в свою очередь подверглось разложению и было заменено четырьмя субполосами 64×64. Этот процесс дает в результате однозначное упорядочивание субизображений, которые характеризуют дискретное вейвлет-преобразование. При переходе из нижнего правого угла рис. 7.10(*а*) в левый верхний размеры субизображений уменьшаются.
4. Рис. 7.10(*а*) не является преобразованием Хаара изображения на рис. 7.1. Хотя в случае данного разложения использовались коэффициенты банка фильтров, которые были взяты из матрицы преобразования Хаара $\mathbf{H}_2$, в дискретном вейвлет-преобразовании могут быть использованы самые различные ортонормированные и биортогональные банки фильтров.
5. Как будет показано в разделе 7.4, каждое субизображение на рис. 7.10(*а*) представляет свою собственную полосу пространственных частот в исходном изображении. Кроме того, многие из субизображений являются чувствительными к направлению. Так, субизображение в левом верх-

нем углу рис. 7.10(*a*) содержит основную информацию о горизонтальных контурах, имеющихся на исходном изображениии.

Можно только поразиться, рассматривая этот впечатляющий список свойств, что дискретное вейвлет-преобразование, показанное на рис. 7.10(*a*), было получено при использовании двухэлементных цифровых фильтров, имеющих всего четыре коэффициента. ■

7.2. Кратномасштабное разложение

В предыдущем разделе мы рассмотрели три хорошо известных метода обработки изображений, которые сыграли важную роль в развитии математической теории, называемой *кратномасштабным анализом* (КМА). В КМА использование *масштабирующей функции* позволяет построить последовательность приближений для некоторой функции или изображения, причем каждое приближение отличается от соседнего масштабным коэффициентом 2. Для кодирования информации, описывающей разность между соседними приближениями, используются дополнительные функции, называемые *вейвлетами*.

7.2.1. Разложения в ряды

Функцию (сигнал) $f(x)$) часто проще анализировать, если представить ее в виде линейной комбинации функций из некоторой *системы функций разложения* $\{\varphi_k(x)\}$

$$f(x) = \sum_k \alpha_k \varphi_k(x), \tag{7.2-1}$$

где индекс суммирования k принимает конечное или бесконечное множество целых значений[6], вещественные числа α_k называются *коэффициентами разложения*, а сами функции φ_k принимают вещественные значения и называются *функциями разложения*. Если разложение единственно, т. е. для любой заданной функции $f(x)$ существует единственный набор коэффициентов α_k такой, что выполнено (7.2-1), то функции $\varphi_k(x)$ называются *базисными функциями*, а все множество *функций разложения* $\{\varphi_k(x)\}$ называется *базисом* в том классе функций, которые могут быть представлены таким образом. Представимые в виде (7.2-1) функции образуют *пространство функций*, которое называется *замыканием линейной оболочки* функций $\{\varphi_k(x)\}$ или *пространством*, *натянутым на систему функций* $\{\varphi_k(x)\}$, и обозначается

$$V = \overline{\operatorname{Span}_k \{\varphi_k(x)\}}. \tag{7.2-2}$$

Запись $f(x) \in V$ означает, что функция $f(x)$ принадлежит замыканию линейной оболочки функций $\{\varphi_k(x)\}$ и может быть записана в виде (7.2-1).

---

[6] Обычно термин «линейная комбинация» употребляется лишь в том случае, если сумма в правой части (7.2-1) конечная, а в случае бесконечного числа слагаемых принято говорить о разложении в ряд. — *Прим. перев.*

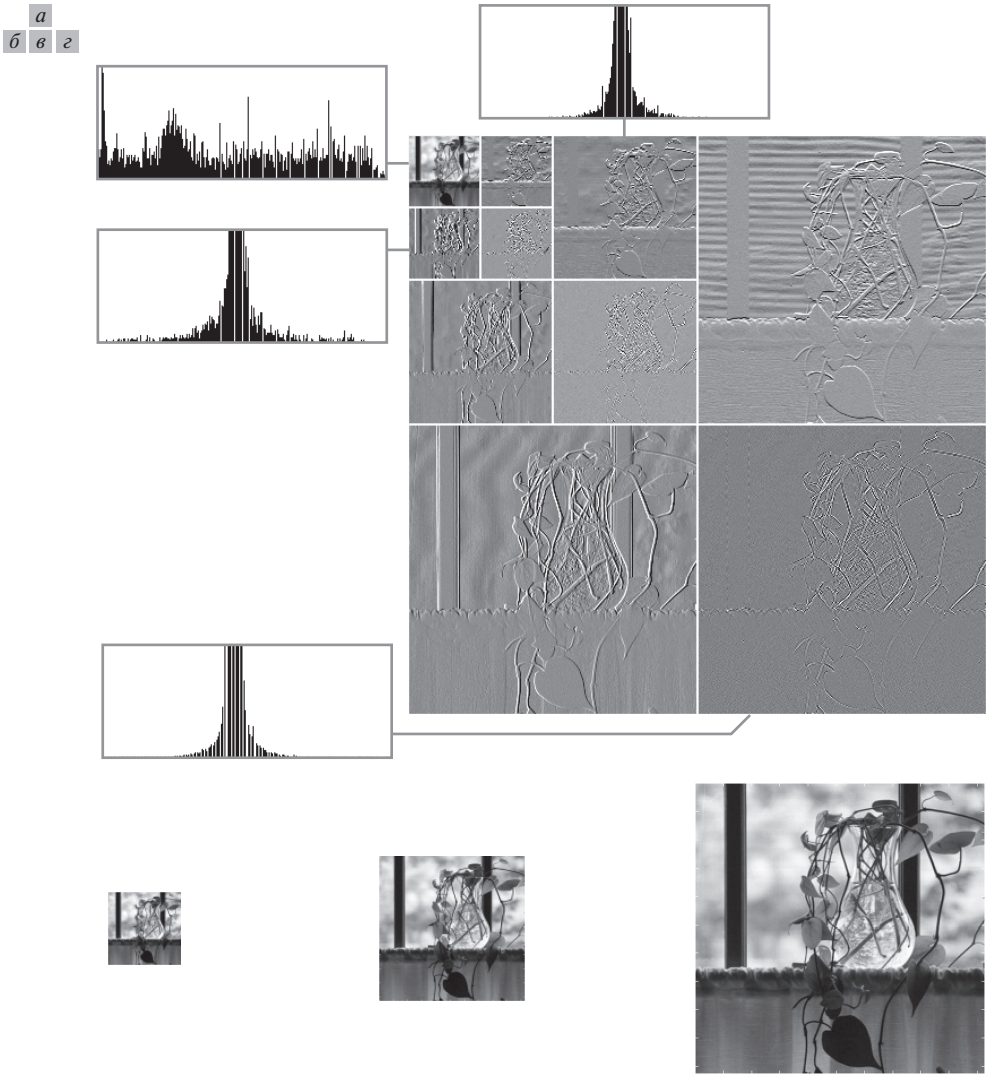


Рис. 7.10. (*а*) Дискретное вейвлет-преобразование с использованием базисных функций Хаара $\mathbf{H}_2$. Приведены гистограммы полученных компонент. (*б*)—(*г*) Несколько различных приближений размерами 64×64, 128×128 и 256×256, которые могут быть получены на основе изображения (*а*)

Для любого пространства функций V и соответствующей системы функций разложения $\{\varphi_k(x)\}$ существует *двойственная* система функций $\{\tilde{\varphi}_k(x)\}$, которая может быть использована для вычисления коэффициентов разложения α_k в (7.2-1) любой функции $f(x)$. Эти коэффициенты вычисляются как *скалярные произведения*[7] двойственных функций $\tilde{\varphi}_k(x)$ и функции $f(x)$, а именно

[7] Скалярное произведение двух вещественно- или комплекснозначных функций $f(x)$ и $g(x)$ задается выражением $\langle f(x), g(x)\rangle = \int f(x)g^*(x)dx$. Если функция $f(x)$ вещественная, то $g^*(x) = g(x)$ и $\langle f(x), g(x)\rangle = \int f(x)g(x)dx$.

$$\alpha_k = \langle f(x), \tilde{\varphi}_k(x) \rangle = \int f(x)\tilde{\varphi}_k^*(x)dx, \tag{7.2-3}$$

где символ * означает операцию комплексного сопряжения[8]. В зависимости от свойств системы функций разложения, можно выделить три различных случая, которые проиллюстрированы в задаче 7.10 в конце главы на примере векторов в двумерном евклидовом пространстве.

Случай 1. Система функций $\{\varphi_k(x)\}$ образует в пространстве V ортонормированный базис, т. е.

$$\langle \varphi_j(x), \varphi_k(x) \rangle = \delta_{jk} = \begin{cases} 0 & j \neq k; \\ 1 & j = k. \end{cases} \tag{7.2-4}$$

В этом случае двойственная система функций совпадает с исходным ортонормированным базисом, $\varphi_k(x) = \tilde{\varphi}_k(x)$, и формула (7.2-3) принимает вид

$$\alpha_k = \langle f(x), \varphi_k(x) \rangle. \tag{7.2-5}$$

Коэффициенты α_k вычисляются как скалярные произведения базисных функций и функции $f(x)$.

Случай 2. Система функций $\{\varphi_k(x)\}$ не является ортогональной, т. е. некоторые из условий

$$\langle \varphi_j(x), \varphi_k(x) \rangle = 0, \quad j \neq k, \tag{7.2-6}$$

нарушаются, но образует в пространстве V базис Рисса[9]. Тогда двойственная система функций $\{\tilde{\varphi}_k(x)\}$ также образует базис Рисса, и вместе они образуют так называемую *биортогональную* систему, что означает выполнение условий

$$\langle \varphi_j(x), \tilde{\varphi}_k(x) \rangle = \delta_{jk} = \begin{cases} 0 & j \neq k; \\ 1 & j = k. \end{cases} \tag{7.2-7}$$

Коэффициенты α_k вычисляются по формуле (7.2-3).

Случай 3. Система функций $\{\varphi_k(x)\}$, участвующих в разложении (7.2-1), не является базисом, т. е. существует более одного набора коэффициентов разложения α_k для функций $f(x) \in V$. В этом случае говорят, что система функций разложения является *переполненной* или избыточной. Если существуют константы $A > 0$, $B < \infty$, такие, что для всех функций $f(x) \in V$ выполнено условие[10]

---

[8] В любом из трех рассматриваемых ниже случаев (т. е. когда система функций разложения образует ортонормированный базис, базис Рисса или фрейм) приведенное утверждение справедливо. — *Прим. перев.*

[9] Базис в гильбертовом пространстве называется базисом Рисса, если он является безусловным, т. е. порядок, в котором производится суммирование в (7.2-1), не имеет значения. Базис Рисса может быть охарактеризован следующим требованием: существуют $A > 0$, $B < \infty$ такие, что для любой функции $f(x)$ выполнено приводимое ниже в тексте условие (7.2-8). Базисы Рисса — следующие после ортонормированных «хорошие» базисы. Отметим, что (1) в конечномерном пространстве любой базис является базисом Рисса и (2) если $A = B = 1$, то базис Рисса является ортонормированным базисом. — *Прим. перев.*

[10] Норма функции $f(x)$ определяется как квадратный корень из скалярного произведения функции на себя и обозначается символом $\|f(x)\|$.

$$A\|f(x)\|^2 \le \sum_k \left|\langle f(x), \varphi_k(x)\rangle\right|^2 \le B\|f(x)\|^2, \qquad (7.2\text{-}8)$$

то функции разложения образуют *фрейм*[11]. В этом случае система двойственных функций также переполнена и образует *двойственный фрейм*[12]. Разделив неравенство (7.2-8) на квадрат нормы функции $f(x)$, мы видим, что константы A и B ограничивают сумму квадратов модулей коэффициентов разложения нормированной функции. Для вычисления коэффициентов разложения можно использовать формулы[13], аналогичные (7.2-3) и (7.2-5). Если границы фрейма равны, $A = B$, то фрейм называется *жестким фреймом* и можно показать [Daubechies, 1992], что в этом случае

$$f(x) = \frac{1}{A}\sum_k \langle f(x), \varphi_k(x)\rangle \varphi_k(x). \qquad (7.2\text{-}9)$$

С точностью до множителя A^{-1}, который является «мерой избыточности» фрейма, последнее выражение совпадает по форме с выражением, которое можно получить, подставив выражение (7.2-5) (для ортонормированных базисов) в разложение (7.2-1)[14].

7.2.2. Масштабирующие функции

Рассмотрим систему функций разложения, которая состоит из целочисленных сдвигов и двоичных изменений масштаба (т. е. двоичных сжатий и растяжений с сохранением нормы) некоторой заданной вещественной квадратично-интегрируемой функции $\varphi(x)$. Таким образом, мы рассматриваем систему функций $\{\varphi_{j,k}(x)\}$ вида

$$\varphi_{j,k} = 2^{j/2}\varphi(2^j x - k), \qquad (7.2\text{-}10)$$

где $j, k \in \mathbb{Z}$ и $\varphi(x) \in L^2(\mathbb{R})$.[15] Здесь индекс k определяет положение функции $\varphi_{j,k}(x)$ на оси x, индекс j — ширину функции $\varphi_{j,k}(x)$ вдоль оси x, а множитель $2^{j/2}$ регулирует высоту (амплитуду) функции. Функция $\varphi(x)$ называется *масштабирующей*

---

[11] Константы A и B называются границами фрейма. Заметим также, что при определении фрейма прямо не требуется, чтобы множество $\{\varphi_k\}$ было переполненным, требуется только выполнение приведенного условия. Однако если множество не переполнено, т. е. функции образуют базис, то фрейм превращается в базис Рисса (ср. прим. 9) и мы попадаем в ситуацию, рассмотренную в *случае* 2. — *Прим. перев.*

[12] Границами двойственного фрейма $\{\tilde{\varphi}_k\}$ являются константы B^{-1} и A^{-1}. — *Прим. перев.*

[13] Формулы разложения имеют вид $\sum_k \langle f, \tilde{\varphi}_k\rangle \varphi_k = f = \sum_k \langle f, \varphi_k\rangle \tilde{\varphi}_k$. — *Прим. перев.*

[14] То обстоятельство, что выражение (7.2 9) весьма похоже на разложение функции по ортонормированному базису, не должно вводить в заблуждение. Даже жесткие фреймы *не* являются ортонормированными базисами, что показывает пример (**в**) из задачи 7.10. Однако если жесткий фрейм состоит из нормированных функций ($\{\varphi_k\}$, $\|\varphi_k\| = 1$) и его границы $A = B = 1$, то такой фрейм *является* ортонормированным базисом. — *Прим. перев.*

[15] Здесь $\mathbb{R}$ — множество вещественных чисел, $\mathbb{Z}$ — множество целых чисел, а $L^2(\mathbb{R})$ обозначает пространство измеримых интегрируемых с квадратом функций на действительной оси.

функцией. Надлежащий выбор функции $\varphi(x)$ позволяет добиться того, что пространство измеримых квадратично-интегрируемых функций $L^2(\mathbb{R})$ оказывается натянутым на систему функций $\{\varphi_{j,k}(x)\}$ (т. е. замыкание линейной оболочки системы функций $\{\varphi_{j,k}(x)\}$ совпадает с $L^2(\mathbb{R})$).

Если мы ограничимся рассмотрением какого-нибудь одного фиксированного значения j в (7.2-10), скажем, $j = j_0$, то получаемая в результате система функций разложения $\{\varphi_{j_0,k}(x)\}$ будет подмножеством всей системы функций $\{\varphi_{j,k}(x)\}$. Замыкание линейной оболочки такой системы не совпадает со всем пространством $L^2(\mathbb{R})$, а является некоторым его подпространством. Используя систему обозначений предыдущего параграфа, это подпространство можно определить выражением

$$V_{j_0} = \overline{\underset{k}{\text{Span}}\{\varphi_{j_0,k}(x)\}}. \tag{7.2-11}$$

Таким образом, подпространство V_{j_0} — это пространство, натянутое на функции $\varphi_{j_0,k}$ при различных значениях k. Если функция $f(x) \in V_{j_0}$, то она может быть записана в виде

$$f(x) = \sum_k \alpha_k \varphi_{j_0,k}(x). \tag{7.2-12}$$

В общем случае обозначим через V_j подпространство, натянутое на систему функций[16] $\{\varphi_{j,k}(x)\}$ при любом фиксированном j и всех k:

$$V_j = \overline{\underset{k}{\text{Span}}\{\varphi_{j,k}(x)\}}. \tag{7.2-13}$$

Как будет видно из следующего примера, при увеличении j возрастает размер подпространства V_j, так что в него попадают более быстро меняющиеся функции (описывающие мелкие детали). Это является следствием того, что при увеличении значения j функции $\varphi_{j,k}(x)$, используемые для представления функций из соответствующего подпространства, становятся у́же и разделяются на меньших расстояниях по оси x.

Пример 7.4. Масштабирующая функция Хаара.

■ Рассмотрим масштабирующую функцию Хаара, которая представляет собой характеристическую функцию полуинтервала [0,1) [Haar, 1910]:

$$\varphi(x) = \begin{cases} 1 & 0 \le x < 1; \\ 0 & \text{в остальных случаях.} \end{cases} \tag{7.2-14}$$

На рис. 7.11(*а*)—(*г*) представлены четыре функции из системы функций разложения, которая может быть получена подстановкой в выражение (7.2-10) масштабирующей функции, имеющей вид прямоугольного импульса. Заметим, что при $j = 1$ функции разложения на рис. 7.11(*в*) и (*г*) имеют половину ширины функций для $j = 0$ на рис. 7.11(*а*) и (*б*). Для данного полуинтервала по x можно задать вдвое больше масштабирующих функций V_1, чем V_0 (например $\varphi_{1,0}$ и $\varphi_{1,1}$ из подпространства V_1 по сравнению с $\varphi_{0,0}$ из подпространства V_0 для полуинтервала $0 \le x < 1$).

---

[16] Ниже эту систему функций автор часто называет масштабирующими функциями j-го масштаба (или подпространства V_j), что не должно вызывать недоразумений. — *Прим. перев.*

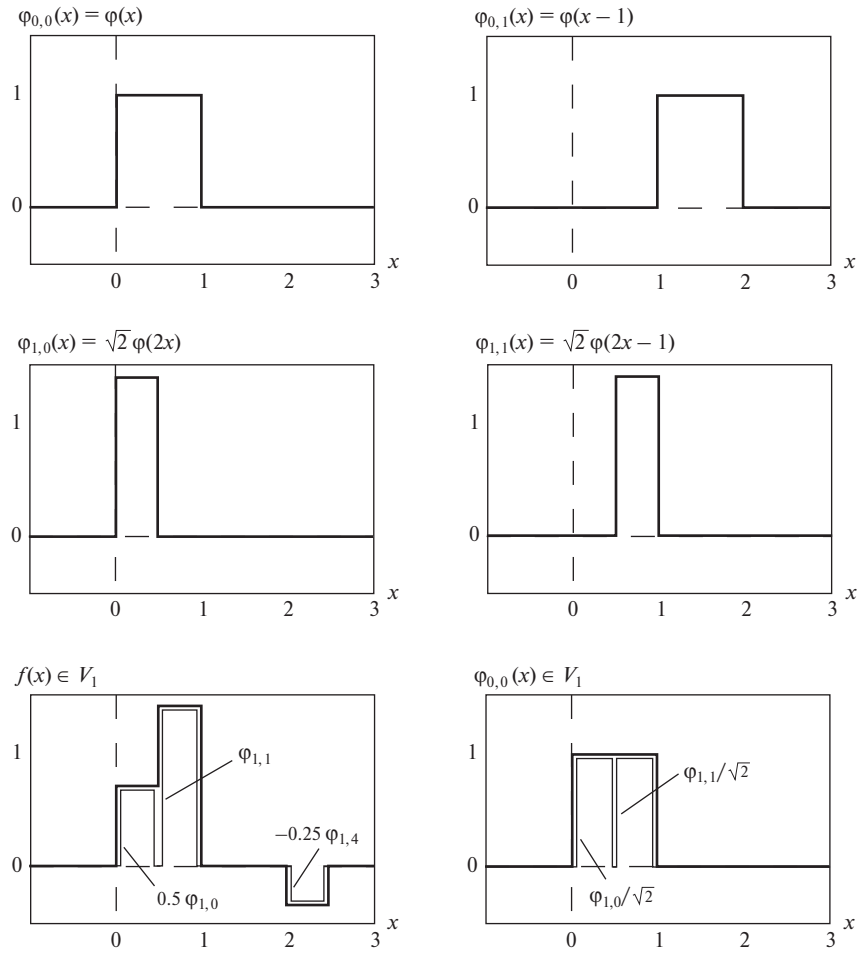


Рис. 7.11. Некоторые масштабирующие функции системы Хаара

На рис. 7.11(*д*) представлена функция из подпространства V_1. Эта функция не принадлежит подпространству V_0, поскольку функции разложения для подпространства V_0 (см. рис. 7.11(*а*) и (*б*)) слишком грубы для представления данной функции с их помощью. Для этого необходимо использовать функции разложения более крупного масштаба (высокого разрешения), такие как функции, представленные на рис. 7.11(*в*) и (*г*). Как показано на рис. 7.11(*д*), использование последних позволяет получить представление интересующей функции в виде разложения, состоящего из трех слагаемых:

$$f(x) = 0{,}5\varphi_{1,0}(x) + \varphi_{1,1}(x) - 0{,}25\varphi_{1,4}(x).$$

Завершая обсуждение примера, заметим, что на рис. 7.11(*е*) представлено разложение функции $\varphi_{0,0}(x)$ по функциям разложения подпространства V_1. Аналогично можно поступить с любой функцией разложения подпространства V_0, если воспользоваться формулой

$$\varphi_{0,k}(x)=\frac{1}{\sqrt{2}}\varphi_{1,2k}(x)+\frac{1}{\sqrt{2}}\varphi_{1,2k+1}(x).$$

Таким образом, если функция $f(x)$ является элементом подпространства V_0, то она является также и элементом подпространства V_1. Причина этого в том, что все функции разложения подпространства V_0 являются элементами подпространства V_1. На математическом языке указанный факт означает, что подпространство V_0 вложено в подпространство V_1 (эквивалентные формулировки: V_0 само является подпространством V_1, V_1 содержит (включает) V_0) и записывается в виде $V_0 \subset V_1$. ■

Простая масштабирующая функция из приведенного примера удовлетворяет четырем основным условиям кратномасштабного анализа [Mallat, 1989a].

КМА-условие 1. Масштабирующая функция и ее целые сдвиги ортогональны.

В случае функции Хаара это условие очевидно выполнено. Действительно, во всех точках, где значение функции Хаара равно единице, значение любой функции, полученной из нее сдвигом на целую величину, равно нулю. Поэтому равно нулю и их скалярное произведение. Говорят, что масштабирующая функция Хаара есть функция с *компактным носителем*. Это означает, что функция обращается в нуль вне некоторого конечного интервала, являющегося носителем[17]. Размер носителя функции Хаара равен единице; функция обращается в нуль вне полуоткрытого интервала [0,1). Следует отметить, что добиться выполнения рассматриваемого условия ортогональности труднее в случае, когда размер носителя масштабирующей функции становится больше единицы.

КМА-условие 2. Подпространства, натянутые на систему масштабирующих функций при низком разрешении (в мелком масштабе), содержатся в подпространствах, натянутых на систему масштабирующих функций при более высоком разрешении (в более крупном масштабе).

Подпространства, содержащие функции высокого разрешения, с необходимостью содержат также все функции более низкого разрешения, как показано на рис. 7.10. Таким образом,

$$V_{-\infty} \subset \cdots \subset V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \cdots \subset V_{\infty}. \qquad (7.2\text{-}15)$$

Более того, подпространства удовлетворяют тому естественному условию, что если $f(x) \in V_j$, то $f(2x) \in V_{j+1}$. То обстоятельство, что масштабирующая функция Хаара удовлетворяет рассматриваемому условию, не следует воспринимать как признак того, что любая функция с размером носителя, не превосходящим единицы, также автоматически удовлетворяет этому условию. В качестве упражнения мы предлагаем читателю убедиться в том, что похожая на функцию Хаара простая функция вида

$$\varphi(x)=\begin{cases}1 & 0{,}25 \le x < 0{,}75,\\ 0 & \text{в остальных случаях}\end{cases}$$

[17] Напомним, что носителем функции f называется замыкание множества тех точек, где $f(x) \neq 0$. Если носитель f — ограниченное множество, то функция f называется финитной. — *Прим. перев.*

не является корректной масштабирующей функцией кратномасштабного анализа (см. задачу 7.11).

КМА-условие 3. Единственной функцией, принадлежащей одновременно всем подпространствам V_j, является функция $f(x) = 0$.

Если мы рассмотрим самую грубую из систем функций разложения (т. е. при $j = -\infty$), то единственной представимой функцией будет функция, не содержащая никакой информации. Итак,

$$V_{-\infty} = \{0\}. \tag{7.2-16}$$

КМА-условие 4. Любая функция может быть представлена с произвольной точностью.

Хотя разложение (подобное тому, которое имело место для функции на рис. 7.11(*д*)) для некоторой конкретной функции $f(x)$ при любом заданном масштабе j может оказаться невозможным, все измеримые квадратично-интегрируемые функции оказываются представимы масштабирующими функциями в пределе $j \to +\infty$. Таким образом[18],

$$V_{+\infty} = \{L^2(\mathbb{R})\}. \tag{7.2-17}$$

При выполнении указанных условий функции разложения подпространства V_j могут быть выражены в виде суммы с весами функций разложения подпространства V_{j+1}. Используя (7.2-12), мы можем записать

$$\varphi_{j,k}(x) = \sum_n \alpha_n \varphi_{j+1,n}(x),$$

где мы для ясности изменили индекс суммирования. Подставив $\varphi_{j+1,k}(x)$ в виде (7.2-10) и заменив обозначение коэффициентов α_n на $h_\varphi(n)$, преобразуем последнее равенство к виду

$$\varphi_{j,k}(x) = \sum_n h_\varphi(n) 2^{(j+1)/2} \varphi(2^{j+1}x - n).$$

> Коэффициенты α_n изменены на $h_\varphi(n)$, поскольку последние используются ниже как коэффициенты банка фильтров (см. раздел 7.4).

Поскольку $\varphi(x) = \varphi_{0,0}(x)$, то положив индексы j и k равными нулю, получим простейшее выражение

$$\varphi(x) = \sum_n h_\varphi(n) \sqrt{2} \varphi(2x - n). \tag{7.2-18}$$

Коэффициенты $h_\varphi(n)$ в этом рекурсивном равенстве называются *масштабными коэффициентами для масштабирующей функции*; при этом вся последовательность h_φ называется *масштабной* (*двухмасштабной*) или *уточняющей последовательностью*[19] *для масштабирующей функции*. Равенство (7.2-18) играет фундаментальную роль в кратномасштабном анализе и называется *уравнением КМА*,

---

[18] Обычно условие 3 записывают в виде $\bigcap_{j\in\mathbb{Z}} V_j = \{0\}$, а условие 4 — в виде $\overline{\bigcup_{j\in\mathbb{Z}} V_j} = L^2(\mathbb{R})$. — *Прим. перев.*

[19] При этом функцию $H_\varphi(\omega) = \frac{1}{\sqrt{2}} \sum_{n\in\mathbb{Z}} h_\varphi(n) e^{-in\omega}$, являющуюся спектром последовательности $h_\varphi(n)$, называют масштабирующим фильтром или уточняющей маской. — *Прим. перев.*

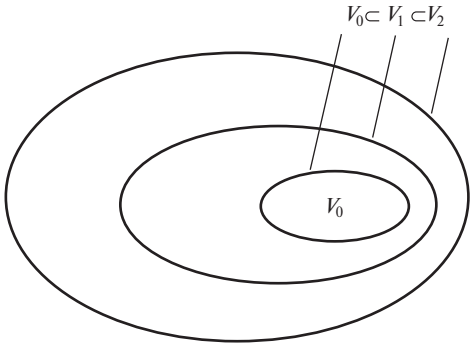


Рис. 7.12. Вложенные функциональные пространства, порождаемые масштабирующей функцией

масштабным равенством или *уточняющим равенством*[20]. Утверждается, что функции разложения любого подпространства могут быть построены из своих копий удвоенного разрешения, т. е. из функций разложения соседнего подпространства более крупного масштаба. Центральное пространство V_0 может быть выбрано произвольно.

Пример 7.5. Масштабные коэффициенты для функции Хаара.

■ Масштабные коэффициенты для заданной (7.2-14) функции Хаара равны $h_\varphi(0) = h_\varphi(1) = 1/\sqrt{2}$ и представляют собой первую строку матрицы $\mathbf{H}_2$ в (7.1-18). Таким образом, уравнение (7.2-28) примет вид

$$\varphi(x) = \frac{1}{\sqrt{2}}\left[\sqrt{2}\varphi(2x)\right] + \frac{1}{\sqrt{2}}\left[\sqrt{2}\varphi(2x-1)\right].$$

Это разложение было проиллюстрировано графически для функции $\varphi_{0,0}(x)$ на рис. 7.11(*e*), где члены в квадратных скобках являются, как нетрудно видеть, функциями $\varphi_{1,0}(x)$ и $\varphi_{1,1}(x)$. После дополнительного упрощения мы получаем $\varphi(x) = \varphi(2x) + \varphi(2x-1)$. ■

7.2.3. Вейвлет-функции

Если задана масштабирующая функция, удовлетворяющая КМА-условиям предыдущего параграфа, мы можем определить такую *вейвлет-функцию*[21] $\psi(x)$, что система функций, состоящих из целых сдвигов и двоичных изменений масштаба функции $\psi(x)$, порождает разность между двумя смежными КМА-подпространствами V_j и V_{j+1}. Данная ситуация проиллюстрирована графически на рис. 7.13. Мы определяем систему вейвлетов

$$\psi_{j,k}(x) = 2^{j/2}\psi(2^j x - k), \tag{7.2-19}$$

[20] Обычно масштабным называют само равенство (7.2-18), а равенство, которое получается из него переходом к образам Фурье, $\Phi(\omega) = H_\varphi(\omega/2)\Phi(\omega/2)$, называют уточняющим. — *Прим. перев.*

[21] Эту функцию называют также базовым или материнским вейвлетом. — *Прим. перев.*

и каждое из пространств W_j на рисунке оказывается натянутым на подсистему $\{\psi_{j,k}(x)\}$ при k $\in \mathbb{Z}$. Как и ранее, мы записываем

$$W_j = \overline{\underset{k}{\text{Span}}\{\psi_{j,k}(x)\}}, \tag{7.2-20}$$

и если $f(x) \in W_j$, то

$$f(x) = \sum_k \alpha_k \psi_{j,k}(x). \tag{7.2-21}$$

Подпространства на рис. 7.13, порождаемые масштабирующей функцией и вейвлет-функцией, связаны между собой соотношением

$$V_{j+1} = V_j \oplus W_j, \tag{7.2-22}$$

где символ $\oplus$ обозначает прямую сумму пространств (аналог объединения множеств). Подпространство W_j является ортогональным дополнением подпространства V_j до подпространства V_{j+1}, причем все элементы подпространства V_j ортогональны к элементам подпространства W_j. Таким образом,

$$\langle \varphi_{j,k}(x), \psi_{j,l}(x) \rangle = 0 \tag{7.2-23}$$

для всех значений j, k, $l \in \mathbb{Z}$.

Теперь мы можем представить пространство измеримых квадратично-интегрируемых функций в виде

$$L^2(\mathbb{R}) = V_0 \oplus W_0 \oplus W_1 \oplus \ldots \tag{7.2-24}$$

или

$$L^2(\mathbb{R}) = V_1 \oplus W_1 \oplus W_2 \oplus \ldots, \tag{7.2-25}$$

или даже

$$L^2(\mathbb{R}) = \ldots \oplus W_{-2} \oplus W_{-1} \oplus W_0 \oplus W_1 \oplus W_2 \oplus \ldots, \tag{7.2-26}$$

причем зависимость от масштабирующей функции исключена из последнего представления, которое, таким образом, определяется только в терминах вейвлетов (т. е. в соотношении (7.2-26) присутствуют только пространства, порождаемые вейвлет-функциями). Заметим, что если функция $f(x)$ принадлежит подпространству V_1, но не принадлежит подпространству V_0, то разложение, основанное на (7.2-24), содержит две части: первая часть есть *приближение* $f(x)$ масштабирующими функциями подпространства V_0; вторая часть из подпространства W_0 при этом связана с вейвлетами и содержит *разность* этого приближения и самой функции. Выражения (7.2-24)—(7.2-26) можно обобщить следующим образом:

$$L^2(\mathbb{R}) = V_{j_0} \oplus W_{j_0} \oplus W_{j_0+1} \oplus \ldots, \tag{7.2-27}$$

где j_0 — произвольный начальный масштаб.

Поскольку вейвлет-пространство W_j принадлежит пространству V_{j+1}, натянутому на масштабирующие функции следующего более крупного масштаба (см. рис. 7.13), любая вейвлет-функция, подобно масштабирующей функции,

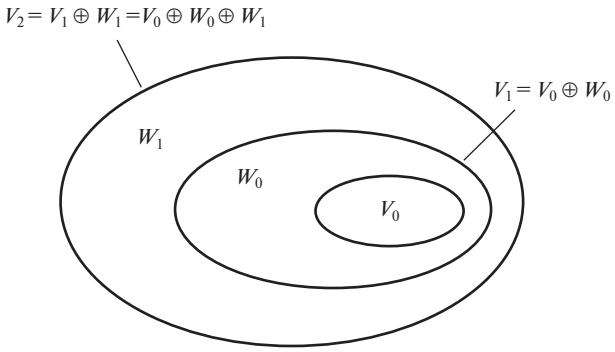


Рис. 7.13. Взаимосвязь функциональных пространств, порождаемых масштабирующей функцией и вейвлет-функцией

может быть выражена в виде суммы с весами масштабирующих функций этого увеличенного масштаба. Значит, можно записать

$$\psi(x) = \sum_n h_\psi(n)\sqrt{2}\varphi(2x-n), \tag{7.2-28}$$

где коэффициенты $h_\psi(n)$ называются *масштабными коэффициентами для вейвлет-функции*, а h_ψ — *масштабной* или *уточняющей последовательностью для вейвлетов*. Используя ортогональность вейвлетов, а также то обстоятельство, что вейвлеты порождают ортогональное дополнение соответствующих пространств, которые изображены на рис. 7.13, можно показать (см., например, [Burrus, Gopinath, Guo, 1998]), что масштабные коэффициенты $h_\psi(n)$ и $h_\varphi(n)$ связаны соотношением

$$h_\psi(n) = (-1)^n h_\varphi(1-n). \tag{7.2-29}$$

Обратим внимание на сходство этого выражения и выражения (7.1-14), которое определяет взаимосвязь импульсных характеристик ортонормированных фильтров анализа и синтеза в схеме субполосного кодирования.

Пример 7.6. Масштабные коэффициенты для вейвлет-функции Хаара.

■ В предыдущем примере было установлено, что ненулевые члены масштабной последовательности для функции Хаара равны $h_\varphi(0) = h_\varphi(1) = 1/\sqrt{2}$. Используя (7.2-29), мы получаем, что ненулевые члены масштабной последовательности для вейвлет-функции Хаара равны $h_\psi(0) = (-1)^0 h_\varphi(1-0) = 1/\sqrt{2}$ и $h_\psi(1) = (-1)^1 h_\varphi(1-1) = -1/\sqrt{2}$. Заметим, что эти коэффициенты соответствуют второй строке матрицы $\mathbf{H}_2$ в (7.1-18). Подставив эти значения в (7.2-28), мы получаем функцию $\psi(x) = \varphi(2x) - \varphi(2x-1)$, график которой представлен на рис. 7.14(*а*). Таким образом, вейвлет-функция Хаара имеет вид

$$\psi(x) = \begin{cases} 1 & 0 \le x < 0{,}5; \\ -1 & 0{,}5 \le x < 1; \\ 0 & \text{в остальных случаях.} \end{cases} \tag{7.2-30}$$

Используя (7.2-19), мы теперь можем построить всю систему вейвлетов Хаара, состоящую из целых сдвигов и двоичных изменений масштаба функции $\psi(x)$. Графики двух вейвлетов из этой системы, $\psi_{0,2}(x)$ и $\psi_{1,0}(x)$, представлены на рис. 7.14(*б*)

и (*в*) соответственно. Отметим, что вейвлет $\psi_{1,0}(x)$ из пространства W_1 ýже, чем вейвлет $\psi_{0,2}(x)$ из пространства W_0; он может быть использован для представления более тонких деталей.

На рис. 7.14(*г*) представлена функция из подпространства V_1, которая не принадлежит подпространству V_0. Эта функция уже рассматривалась в примере ранее (см. рис. 7.11(*д*)). Несмотря на то, что эта функция не может быть точно представлена в подпространстве V_0, формула (7.2-22) показывает, что она может быть представлена с использованием функций разложения подпространств V_0 и W_0. Результат имеет вид

$$f(x) = f_a(x) + f_d(x),$$

где

$$f_a(x) = \frac{3\sqrt{2}}{4}\varphi_{0,0}(x) - \frac{\sqrt{2}}{8}\varphi_{0,2}(x)$$

а *б*
в *г*
д *е*

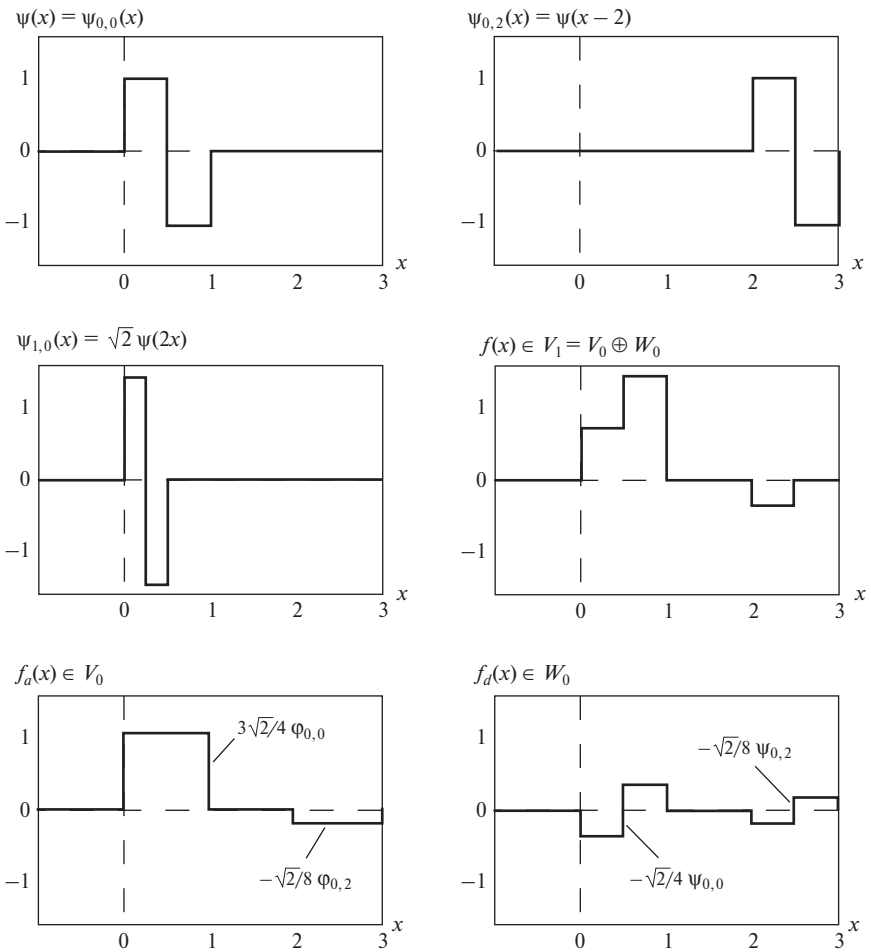


Рис. 7.14. Вейвлеты Хаара в подпространствах W_0 и W_1

и $$f_d(x) = \frac{-\sqrt{2}}{4}\psi_{0,0}(x) - \frac{\sqrt{2}}{8}\psi_{0,2}(x).\text{ф}$$

Здесь слагаемое $f_a(x)$ представляет собой приближение функции $f(x)$ масштабирующими функциями подпространства V_0, в то время как слагаемое $f_d(x)$ представляет собой разность $f(x) - f_a(x)$, выраженную в виде суммы вейвлетов подпространства W_0. Полученное разложение, части которого представлены на рис. 7.14(*д*) и (*е*), аналогично разложению функции $f(x)$ с помощью низкочастотного и высокочастотного фильтров, как это обсуждалось в связи с рис. 7.6. Приближение $f_a(x)$ содержит низкие частоты функции $f(x)$ (оно состоит из средних значений функции $f(x)$ в целых интервалах), а высокочастотная (детальная) составляющая содержится в части $f_d(x)$. ■

7.3. Одномерные вейвлет–преобразования

Теперь мы в состоянии рассмотреть несколько тесно связанных между собой вейвлет-преобразований: *разложение в вейвлет-ряды*, *дискретное вейвлет-преобразование* и *интегральное* (*непрерывное*) *вейвлет-преобразование*. В Фурье-анализе аналогами этих преобразований являются соответственно разложение в ряд Фурье, дискретное Фурье-преобразование и интегральное (непрерывное) Фурье-преобразование. В разделе 7.4 мы рассмотрим эффективный с вычислительной точки зрения способ реализации дискретного вейвлет-преобразования, который называется *быстрым вейвлет-преобразованием*.

7.3.1. Разложение в вейвлет-ряды

Мы начнем с рассмотрения разложения функции $f(x) \in L^2(\mathbb{Z})$ в *вейвлет-ряд*, отвечающий вейвлет-функции $\psi(x)$ и масштабирующей функции $\varphi(x)$. В соответствии с (7.2-27) $f(x)$ может быть представлена разложением масштабирующей функции в подпространстве V_{j_0} (такое разложение определяется уравнением (7.2-12)) и некоторыми из разложений вейвлет-функций в подпространствах $W_{j_0}, W_{j_0+1}, \ldots$ (как это определено в (7.2-21)). Таким образом,

$$f(x) = \sum_k c_{j_0}(k)\varphi_{j_0,k}(x) + \sum_{j=j_0}^{+\infty}\sum_k d_j(k)\psi_{j,k}(x), \qquad (7.3\text{-}1)$$

где значение j_0 определяет произвольный начальный масштаб, а коэффициенты $c_{j_0}(k)$ и $d_j(k)$ суть переобозначенные коэффициенты разложения из (7.2-12) и (7.2-21). Коэффициенты $c_{j_0}(k)$ обычно называются *коэффициентами приближения* или *масштабными коэффициентами*, а коэффициенты $d_j(k)$ — *коэффициентами деталей* или *вейвлет-коэффициентами*. Использование таких названий оправдано следующими обстоятельствами. Первое слагаемое в правой части (7.3-1) дает приближение функции $f(x)$ в масштабе j_0, выраженное посредством соответствующих масштабирующих функций (за исключением того случая, когда $f(x) \in V_{j_0}$ и сумма масштабирующих функций равна

$f(x)$). Второе слагаемое в правой части (7.3-1) соответствует более крупным масштабам $j \geq j_0$ и добавляет к приближению «высокоразрешающую часть» — сумму вейвлетов, что обеспечивает возрастающую по мере роста j детализацию. Если система функций разложения образует ортонормированный базис или жесткий фрейм (что является наиболее частым случаем), коэффициенты разложения вычисляются на основе формул (7.2-5) или (7.2-9) следующим образом:

$$c_{j_0}(k) = \left\langle f(x), \varphi_{j_0,k}(x) \right\rangle = \int f(x)\varphi_{j_0,k}(x)dx \qquad (7.3\text{-}2)$$

и

$$d_j(k) = \left\langle f(x), \psi_{j,k}(x) \right\rangle = \int f(x)\psi_{j,k}(x)dx. \qquad (7.3\text{-}3)$$

Поскольку f является вещественной, комплексного сопряжения в скалярных произведениях (7.3-2) и (7.3-3) не требуется.

В уравнениях (7.2-5) и (7.2-9) коэффициенты разложения (т. е. α_k) определяются как скалярные произведения преобразуемой функции и используемых функций разложения. В (7.3-2) и (7.3-3) функциями разложения являются $\varphi_{j_0,k}$ и $\psi_{j,k}$; коэффициентами разложения являются c_{j_0} и d_j. Если функции разложения являются частью биортогонального базиса, то φ- и ψ-функции в соответствующих членах этих выражений должны быть заменены на двойственные $\tilde{\varphi}$- и $\tilde{\psi}$-функции.

Пример 7.7. Разложение функции $y = x^2$ в вейвлет-ряд по системе Хаара.

■ Рассмотрим простую функцию

$$y = \begin{cases} x^2 & 0 \leq x < 1; \\ 0 & \text{в остальных случаях,} \end{cases}$$

график которой представлен на рис. 7.15(*а*). Используя (7.3-2) и (7.3-3), мы можем получить для коэффициентов разложения этой функции по вейвлетам Хаара с начальным масштабом $j_0 = 0$ (см. (7.2-14) и (7.2-30)) следующие значения:

$$c_0(0) = \int_0^1 x^2\varphi_{0,0}(x)dx = \int_0^1 x^2 dx = \frac{x^3}{3}\Big|_0^1 = \frac{1}{3};$$

$$d_0(0) = \int_0^1 x^2\psi_{0,0}(x)dx = \int_0^{0,5} x^2 dx - \int_{0,5}^1 x^2 dx = -\frac{1}{4};$$

$$d_1(0) = \int_0^1 x^2\psi_{1,0}(x)dx = \int_0^{0,25} x^2\sqrt{2}\,dx - \int_{0,25}^{0,5} x^2\sqrt{2}\,dx = -\frac{\sqrt{2}}{32};$$

$$d_1(1) = \int_0^1 x^2\psi_{1,1}(x)dx = \int_{0,5}^{0,75} x^2\sqrt{2}\,dx - \int_{0,75}^1 x^2\sqrt{2}\,dx = -\frac{3\sqrt{2}}{32}.$$

Подставляя полученные значения в (7.3-1), мы получаем начальную часть разложения в вейвлет-ряд

$$y=\underbrace{\frac{1}{3}\varphi_{0,0}(x)}_{V_0}+\underbrace{\left[-\frac{1}{4}\psi_{0,0}(x)\right]}_{W_0}+\underbrace{\left[-\frac{\sqrt{2}}{32}\psi_{1,0}(x)-\frac{3\sqrt{2}}{32}\psi_{1,1}(x)\right]}_{W_1}+\ldots\ .$$

$$V_1=V_0\oplus W_0$$

$$V_2=V_1\oplus W_1=V_0\oplus W_0\oplus W_1$$

В первом члене этого ряда коэффициент $c_0(0)$ используется для формирования приближения подлежащей разложению функции в подпространстве V_0. Это приближение показано на рис. 7.15(*б*) и представляет собой среднее значение исходной функции. Во втором члене коэффициент $d_0(0)$ используется для уточнения этого приближения с помощью добавления деталей из подпространства W_0. Добавляемые детали и результирующее приближение в пространстве V_1 представлены на рис. 7.15(*в*) и (*г*) соответственно. Для добавления деталей из подпространства W_1 на следующем уровне используются коэффициенты $d_1(0)$ и $d_1(1)$. Эти дополнительные детали представлены на рис. 7.15(*д*), а на рис. 7.15(*е*) представлено результирующее приближение в пространстве V_2. Заметим, что последнее приближение начинает напоминать исходную функцию. По мере того как масштаб увеличивается (добавляются детали все более

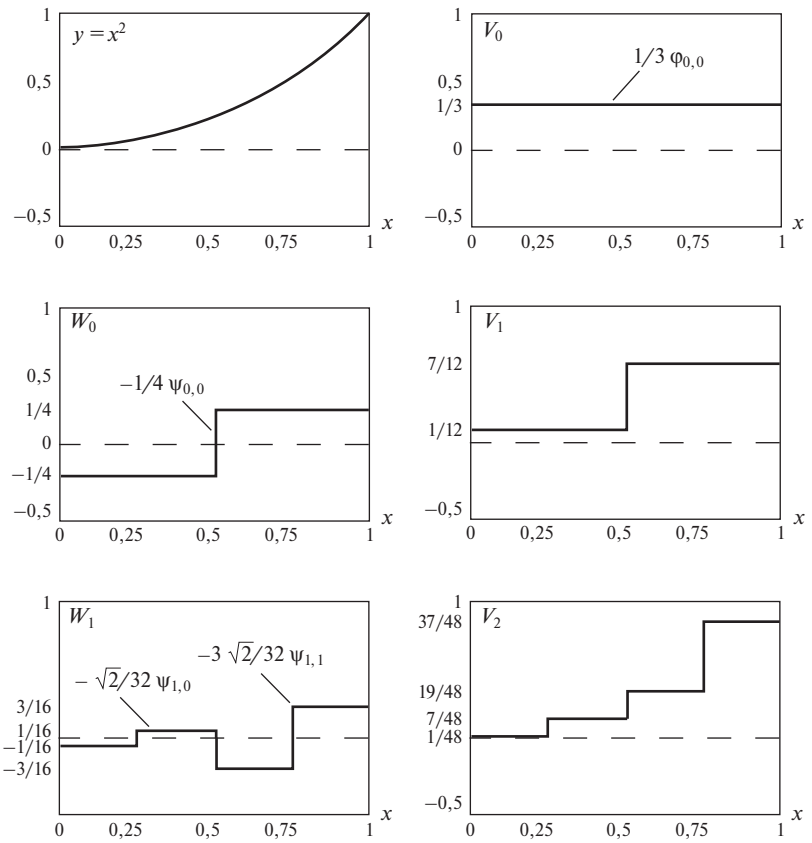


Рис. 7.15. Разложение функции $y = x^2$ в вейвлет-ряд по системе вейвлетов Хаара

высокого уровня), приближение становится все более точным и совпадает с исходной функцией в пределе j → ∞. ■

7.3.2. Дискретное вейвлет-преобразование

Подобно разложению в ряд Фурье, рассмотренное в предыдущем параграфе разложение в вейвлет-ряд ставит в соответствие функции непрерывного аргумента некоторую последовательность коэффициентов. В том случае, когда подлежащая разложению функция является дискретной (т.е. последовательностью чисел), получаемая последовательность коэффициентов называется *дискретным вейвлет-преобразованием* (ДВП) функции $f(x)$. Например, если $f(n) = f(x_0 + n\Delta x)$ для некоторых значений x_0, Δx и $n = 0, 1, 2, ..., M - 1$, коэффициенты разложения $f(x)$ в вейвлет-ряд (задаваемые уравнениями (7.3-2) и (7.3-3)) становятся коэффициентами *прямого* ДВП последовательности $f(n)$:

$$W_\varphi(j_0,k) = \frac{1}{\sqrt{M}} \sum_n f(n) \varphi_{j_0,k}(n), \tag{7.3-5}$$

$$W_\psi(j,k) = \frac{1}{\sqrt{M}} \sum_n f(n) \psi_{j,k}(n) \quad \text{для} \quad j \ge j_0. \tag{7.3-6}$$

В этих формулах функции $\varphi_{j_0,k}(n)$ и $\psi_{j,k}(n)$ являются дискретными версиями базисных функций $\varphi_{j_0,k}(x)$ и $\psi_{j,k}(x)$. Например, $\varphi_{j_0,k}(n) = \varphi_{j_0,k}(x_s + n\Delta x_s)$ для некоторых значений x_s, Δx и $n = 0, 1, 2, ..., M - 1$. Тем самым используются M равноотстоящих отсчетов в области носителя базисных функций (см. ниже пример 7.8). В соответствии с уравнением (7.3-1) дополнением к прямому будет *обратное* ДВП вида

$$f(n) = \frac{1}{\sqrt{M}} \sum_k W_\varphi(j_0,k) \varphi_{j_0,k}(n) + \frac{1}{\sqrt{M}} \sum_{j=j_0}^{+\infty} \sum_k W_\psi(j,k) \psi_{j,k}(n). \tag{7.3-7}$$

Обычно полагают $j_0 = 0$ и выбирают число M так, чтобы оно было степенью двойки (т.е. $M = 2^J$); при этом суммирование в уравнениях (7.3-5)—(7.3-7) производится по значениям $n = 0, 1, 2, ..., M - 1$, $j = 0, 1, 2, ..., J - 1$ и $k = 0, 1, 2, ..., 2^j - 1$. Для системы Хаара дискретные аналоги масштабирующих функций и вейвлет-функций (т.е. базисных функций), участвующих в преобразовании, соответствуют строкам $M \times M$ матрицы преобразования Хаара из раздела 7.1.3. Само преобразование состоит из M коэффициентов, минимальный масштаб равен нулю, а максимальный равен $J - 1$. По причинам, указанным в разделе 7.3.1 и проиллюстрированным на примере 7.6, определяемые формулами (7.3-5) и (7.3-6) коэффициенты обычно называют *коэффициентами приближения* и *коэффициентами деталей* соответственно.

Коэффициенты $W_\varphi(j_0,k)$ и $W_\psi(j,k)$ в формулах (7.3-5)—(7.3-6) соответствуют коэффициентам разложения $c_{j_0}(k)$ и $d_j(k)$ функции $f(x)$ в вейвлет-ряд, которое было рассмотрено в предыдущем параграфе. (Такая смена обозначений не является обязательной, но приводит к стандартной системе записи, используемой при рассмотрении интегрального вейвлет-преобразования, которое еще предстоит нам в следующем параграфе.) Заметим, что интегрирование, исполь-

зуемое при разложении в вейвлет-ряд, заменилось суммированием, и нормировочный множитель $1/\sqrt{M}$, напоминающий аналогичный множитель ДПФ в разделе 4.2.1, добавился в выражения как для прямого, так и обратного преобразования. Этот множитель может быть также отнесен в прямое или обратное преобразование целиком (как $1/M$). В заключение необходимо напомнить, что формулы (7.3-5)—(7.3-7) справедливы только для ортонормированных базисов и жестких фреймов. Для биортогональных базисов функции φ и ψ в формулах (7.3-5) и (7.3-6) должны быть заменены на двойственные функции $\tilde{\varphi}$ и $\tilde{\psi}$ соответственно.

Пример 7.8. Вычисление одномерного дискретного вейвлет-преобразования.

■ Проиллюстрируем использование формул (7.3-5)—(7.3-7) на примере дискретной функции, состоящей из четырех элементов: $f(0) = 1$, $f(1) = 4$, $f(2) = -3$ и $f(3) = 0$. Поскольку $M = 4$, $J = 2$, то при $j_0 = 0$ суммирование производится по $x = 0, 1, 2, 3$ и $k = 0$ при $j = 0$ или $k = 0,1$ при $j = 1$. Мы будем использовать систему вейвлетов Хаара и предполагать, что четыре отсчета функции $f(x)$ распределены по носителю масштабирующей функции, который представляет собой отрезок [0,1]. Подставляя отсчеты в (7.3-5), находим

$$W_{\varphi}(0,0) = \frac{1}{2}\sum_{n=0}^{3} f(n)\varphi_{0,0}(n) = \frac{1}{2}[1\cdot 1 + 4\cdot 1 - 3\cdot 1 + 0\cdot 1] = 1,$$

поскольку $\varphi_{0,0}(x) = 1$ при $x = 0, 1, 2, 3$. Заметим, что мы использовали равномерно распределенные в пространстве отсчеты масштабирующей функции Хаара при $j = 0$ и $k = 0$. Значения отсчетов соответствуют первой строке матрицы преобразования Хаара $\mathbf{H}_4$ из раздела 7.1.3. Далее, используя (7.3-6) и значения аналогично расположенных в пространстве отсчетов функций $\psi_{j,k}(x)$, которые соответствуют строкам 2, 3 и 4 матрицы $\mathbf{H}_4$, мы получаем

$$W_{\psi}(0,0) = \frac{1}{2}[1\cdot 1 + 4\cdot 1 - 3\cdot(-1) + 0\cdot(-1)] = 4,$$

$$W_{\psi}(1,0) = \frac{1}{2}\left[1\cdot\sqrt{2} + 4\cdot\left(-\sqrt{2}\right) - 3\cdot 0 + 0\cdot 0\right] = -1{,}5\sqrt{2},$$

$$W_{\psi}(1,1) = \frac{1}{2}\left[1\cdot 0 + 4\cdot 0 - 3\cdot\sqrt{2} + 0\cdot\left(-\sqrt{2}\right)\right] = -1{,}5\sqrt{2}.$$

Таким образом, дискретное вейвлет-преобразование по системе вейвлетов Хаара нашей простой функции, состоящей из четырех отсчетов, есть $\{1, 4, -1{,}5\sqrt{2}, -1{,}5\sqrt{2}\}$, причем коэффициенты приведены в порядке их вычисления.

Выражение (7.3-7) позволяет нам восстановить исходную функцию по ее вейвлет-преобразованию. В развернутом виде мы имеем для $n = 0, 1, 2, 3$

$$f(n) = \frac{1}{2}\left[W_{\varphi}(0,0)\varphi_{0,0}(n) + W_{\psi}(0,0)\psi_{0,0}(n) + W_{\psi}(1,0)\psi_{1,0}(n) + W_{\psi}(1,1)\psi_{1,1}(n)\right].$$

При $n = 0$, например,

$$f(0)=\frac{1}{2}\left[1\cdot 1+4\cdot 1-1{,}5\cdot\left(\sqrt{2}\right)-1{,}5\sqrt{2}\cdot 0\right]=1.$$

При вычислении обратного преобразования, так же как и при вычислении прямого, мы используем равномерно распределенные в пространстве отсчеты масштабирующей функции и вейвлетов. ■

Четырехточечное ДВП предыдущего примера служит иллюстрацией двухмасштабного (j = {0,1}) разложения функции $f(x)$. Исходное предположение состояло в том, что начальный масштаб равен нулю. Возможно, однако, использование и других начальных масштабов. В качестве упражнения (задача 7.16) мы предлагаем читателю получить одномасштабное преобразование $\{2{,}5\sqrt{2}, -1{,}5\sqrt{2}, -1{,}5\sqrt{2}, -1{,}5\sqrt{2}\}$, которое соответствует начальному масштабу, равному 1. Таким образом, выражения (7.3-5) и (7.3-6) определяют семейство преобразований, различающихся начальным масштабом j_0.

7.3.3. Интегральное вейвлет-преобразование

Естественным обобщением дискретного вейвлет-преобразования является *интегральное* (*непрерывное*) *вейвлет-преобразование* (ИВП). Это преобразование отображает непрерывную функцию одной переменной в функцию двух непрерывных переменных — сдвига и сжатия (растяжения), содержащую сильно избыточную информацию. Получаемая функция допускает удобную интерпретацию и полезна при частотно-временном анализе. Хотя наши интересы лежат в области дискретных функций (изображений), мы для полноты рассмотрим здесь ИВП.

Интегральное вейвлет-преобразование непрерывной квадратично-интегрируемой функции $f(x)$ относительно вещественнозначного базового вейвлета $\psi(x)$ задается формулой

$$W_{\psi}(s,\tau)=\int_{-\infty}^{+\infty} f(x)\psi_{s,\tau}(x)\,dx, \tag{7.3-8}$$

где

$$\psi_{s,\tau}(x)=\frac{1}{\sqrt{s}}\psi\left(\frac{x-\tau}{s}\right), \tag{7.3-9}$$

и параметры $s > 0$, τ называются соответственно *параметрами масштаба* и *сдвига*. Исходная функция $f(x)$ может быть восстановлена по заданной функции $W_{\psi}(s,\tau)$ с помощью формулы *обратного интегрального вейвлет-преобразования*:

$$f(x)=\frac{2}{C_{\psi}}\int_{0}^{+\infty}\int_{-\infty}^{+\infty} W_{\psi}(s,\tau)\frac{\psi_{s,\tau}(x)}{s^2}\,d\tau\,ds, \tag{7.3-10}$$

где

$$C_{\psi}=\int_{-\infty}^{+\infty}\frac{|\Psi(u)|^2}{|u|}\,du \tag{7.3-11}$$

и $\Psi(u)$ — Фурье-преобразование функции $\psi(x)$. Выражения (7.3-8)–(7.3-11) определяют обратимое преобразование, если выполнено так называемое *условие до-*

пустимости $C_\psi < \infty$ [Grossman, Morlet, 1984]. В большинстве случаев это условие означает просто, что $\Psi(0) = 0$ и $\Psi(u) \to 0$ при $u \to \infty$ достаточно быстро, для того чтобы обеспечить $C_\psi < \infty$.

Написанные выше выражения напоминают свои дискретные аналоги — выражения (7.2-19), (7.3-1), (7.3-3), (7.3-6) и (7.3-7). Следует отметить их следующие сходные черты.

1. Непрерывный параметр сдвига τ играет ту же роль, что и целый параметр сдвига k.
2. Непрерывный параметр масштаба s связан с двоичным параметром масштаба 2^j обратной зависимостью, поскольку в выражении (7.3-9) он входит в знаменатель: $\psi((x - \tau)/s)$. Таким образом, вейвлеты, используемые в интегральном преобразовании, сжимаются при $0 < s < 1$ и растягиваются при $s > 1$. В этом смысле понятие масштаба для вейвлета и наше обычное понятие частоты находятся в «обратном отношении».
3. Интегральное преобразование аналогично разложению в вейвлет-ряд (см. (7.3-1)) или дискретному вейвлет-преобразованию (см. (7.3-6)), в которых начальный масштаб $j_0 = -\infty$. В соответствии с (7.2-26) это исключает явную зависимость от масштабирующей функции; таким образом, функция определяется только в терминах вейвлетов.
4. Подобно дискретному преобразованию, интегральное преобразование может рассматриваться как множество коэффициентов $\{W_\psi(s,\tau)\}$, величины которых дают количественную «меру сходства» функции $f(x)$ с функциями из множества базисных функций $\{\psi_{s,\tau}(x)\}$. В непрерывном случае, однако, оба множества бесконечны. Поскольку функции $\psi_{s,\tau}(x)$ являются вещественнозначными, т. е. $\psi_{s,\tau}(x) = \psi^*_{s,\tau}(x)$, то каждый из коэффициентов в (7.3-8) представляет собой скалярное произведение $\langle f(x), \psi_{s,\tau}(x)\rangle$ функций $f(x)$ и $\psi_{s,\tau}(x)$.

Пример 7.9. Одномерное интегральное вейвлет-преобразование.

■ Вейвлет «мексиканская шляпа» вида

$$\psi(x) = \left(\frac{2}{\sqrt{3}}\pi^{-1/4}\right)(1 - x^2)e^{-x^2/2} \tag{7.3-12}$$

получил такое название из-за своей характерной формы (см. рис. 7.16(*а*)). Он пропорционален второй производной от гауссовой функции плотности распределения, имеет нулевое среднее и очень быстро убывает при $|x| \to \infty$. Хотя этот вейвлет удовлетворяет условию допустимости, тем не менее не существует связанной с ним масштабирующей функции, и при вычислении преобразования нельзя использовать ортогональность. Наиболее примечательными свойствами этого вейвлета является его симметрия, а также то, что он задается явной формулой (7.3-12).

Непрерывная одномерная функция на рис. 7.16(*а*) представляет собой сумму двух вейвлетов «мексиканская шляпа»:

$$f(x) = \psi_{1,10}(x) + \psi_{6,80}(x).$$

Фурье-спектр этой функции, представленный на рис. 7.16(*б*), обнаруживает тесную связь между масштабом вейвлета и его частотным диапазоном. Спектр

состоит из двух частотных диапазонов, которые соответствуют двум всплескам гауссова типа, составляющим функцию.

На рис. 7.16(*в*) представлена часть ($1 \le s \le 10$ и $\tau \le 100$) ИВП функции на рис. 7.16(*а*), построенной на основе вейвлета «мексиканская шляпа». В отличие от Фурье-спектра, показанного на рис. 7.16(*б*), ИВП содержит одновременно как пространственную, так и частотную информацию. Заметим, например, что при $s = 1$ преобразование достигает максимума в точке $\tau = 10$, которая отвечает положению в пространстве составляющей части $\psi_{1,10}(x)$ функции $f(x)$. Поскольку значение преобразования определяет объективную «меру сходства» функции $f(x)$ и того вейвлета, для которого оно посчитано, легко понять, как преобразование может быть использовано для нахождения характерных деталей. Нужно просто подобрать вейвлет так, чтобы он соответствовал по форме интересующей детали. Аналогичные наблюдения могут быть сделаны на основе полутонового изображения ИВП на рис. 7.16(*г*), где абсолютная величина преобразования $|W\psi(s,\tau)|$ представлена в виде яркости соответствующей точки. Отметим еще раз, что интегральное вейвлет-преобразование превращает одномерную функцию в двумерную. ■

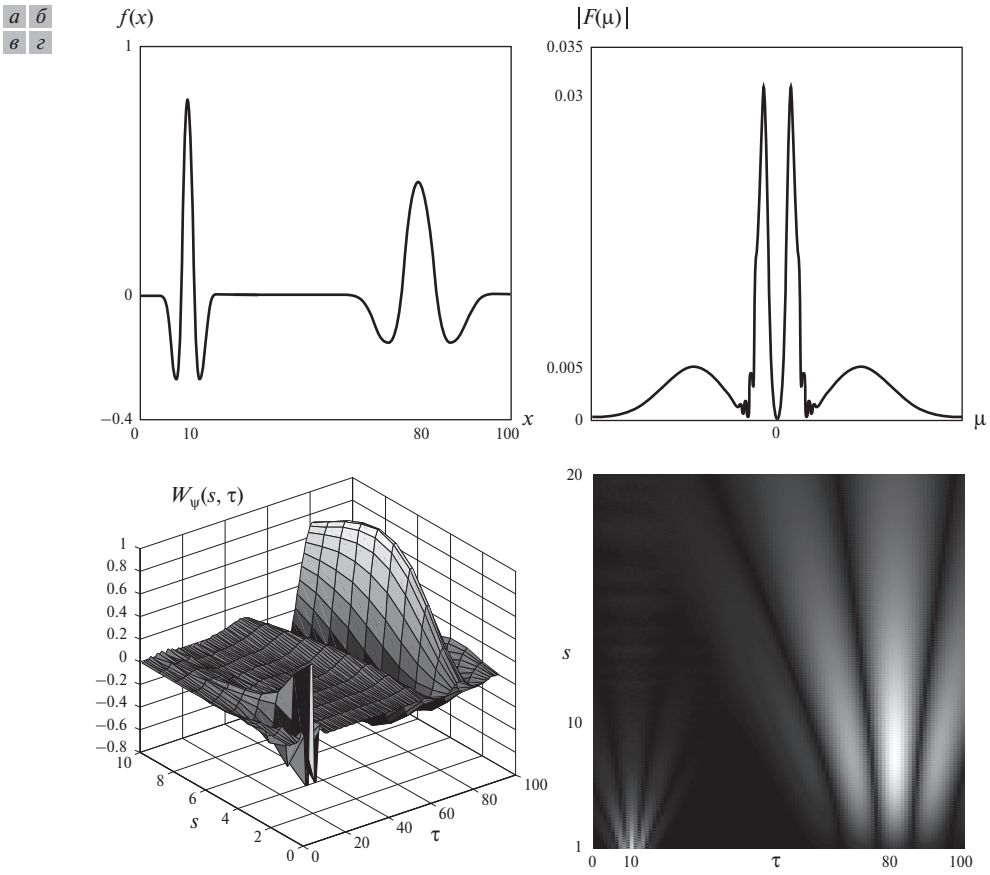


Рис. 7.16. (*а*) Одномерная непрерывная функция. (*б*) Ее Фурье-спектр и интегральное вейвлет-преобразование (*в*) и (*г*)

7.4. Быстрое вейвлет–преобразование

Быстрое вейвлет-преобразование (БВП) представляет собой эффективный метод реализации вычислений дискретного вейвлет-преобразования (ДВП), который использует взаимосвязь между коэффициентами ДВП соседних масштабов. Метод БВП, называемый также *иерархическим алгоритмом*[22] *Малла* [Mallat, 1989a, 1989б], напоминает рассмотренную в разделе 7.1.2 схему двухканального субполосного кодирования.

Рассмотрим снова масштабное равенство кратномасштабного анализа

$$\varphi(x)=\sum_n h_\varphi(n)\sqrt{2}\varphi(2x-n). \tag{7.4-1}$$

Уравнение (7.4-1) повторяет уравнение (7.2-1) раздела 7.2.2.

Изменим масштаб по переменной x в 2^j раз, осуществим сдвиг на величину k и сделаем замену переменной суммирования $m = 2k + n$. Это дает

$$\begin{aligned}\varphi(2^j x-k) &= \sum_n h_\varphi(n)\sqrt{2}\,\varphi(2(2^j x-k)-n)= \\ &= \sum_m h_\varphi(n)\sqrt{2}\,\varphi(2^{j+1}x-2k-n)= \\ &= \sum_m h_\varphi(m-2k)\sqrt{2}\,\varphi(2^{j+1}x-m).\end{aligned} \tag{7.4-2}$$

Заметим, что члены уточняющей последовательности h_φ могут рассматриваться как весовые коэффициенты в разложении функции $\varphi(2^j x - k)$ в сумму масштабирующих функций $(j + 1)$-го масштаба. Та же последовательность действий — начиная с уравнения (7.2-28) — дает аналогичный результат для вейвлета $\psi(2^j x - k)$:

$$\psi(2^j x-k)=\sum_m h_\psi(m-2k)\sqrt{2}\,\varphi(2^{j+1}x-m), \tag{7.4-3}$$

где уточняющая последовательность h_φ для масштабирующей функции в (7.4-2) соответствует уточняющей последовательности для вейвлетов h_ψ в (7.4-3).

Теперь рассмотрим выражения (7.3-5) и (7.3-6) из раздела 7.3.3, которые определяют коэффициенты разложения в вейвлет-ряд непрерывной функции $f(x)$. Подставляя определение вейвлетов (7.2-19) в уравнение (7.3-3), получаем выражение

$$d_j(k)=\int f(x)2^{j/2}\psi(2^j x-k)dx, \tag{7.4-4}$$

которое после замены функций $\psi(2^j x - k)$ на их выражения в правой части (7.4-3) принимает вид

$$d_j(k)=\int f(x)2^{j/2}\left[\sum_m h_\psi(m-2k)\sqrt{2}\,\varphi(2^{j+1}x-m)\right]dx. \tag{7.4-5}$$

---

[22] В оригинале *herringbone algorithm*, что означает дословно «елочный алгоритм». В русскоязычной литературе иногда используется название «графически-изобразительный алгоритм». — *Прим. перев.*

Коэффициенты разложения в вейвлет-ряд становятся коэффициентами ДВП, когда f является дискретной. Мы начинаем с коэффициентов разложения в ряд для упрощения вывода; свободно можно было бы вместо этого воспользоваться более ранними результатами (такими, как определения масштабирующей функции и вейвлетов).

Изменение порядка суммирования и простые подобные преобразования теперь дают

$$d_j(k) = \sum_m h_\psi(m-2k)\left[\int f(x)2^{(j+1)/2}\varphi(2^{j+1}x-m)dx\right], \tag{7.4-6}$$

причем величина в квадратных скобках есть $c_{j_0}(k)$ из уравнения (7.3-2) при $j_0 = j+1$ и $k = m$. Чтобы убедиться в этом, достаточно подставить (7.2-10) в (7.3-2) и положить j_0 равным $j+1$, а $k = m$. Таким образом, мы можем записать

$$d_j(k) = \sum_m h_\psi(m-2k)c_{j+1}(m) \tag{7.4-7}$$

и констатировать, что коэффициенты деталей ДВП масштаба j выражаются через коэффициенты приближения масштаба $(j+1)$. Начиная с (7.4-2) и (7.3-2), мы выводим таким же образом аналогичное выражение, связывающее коэффициенты приближения разложения в вейвлет-ряд (и ДВП) двух соседних масштабов:

$$c_j(k) = \sum_m h_\varphi(m-2k)c_{j+1}(m). \tag{7.4-8}$$

Поскольку коэффициенты разложения в вейвлет-ряд $c_j(k)$ и $d_j(k)$ в случае дискретной $f(x)$ становятся коэффициентами ДВП $W_\varphi(j,k)$ и $W_\psi(j,k)$ (см. раздел 7.3.2), можно записать:

$$W_\psi(j,k) = \sum_m h_\psi(m-2k)W_\varphi(j+1,m), \tag{7.4-9}$$

$$W_\varphi(j,k) = \sum_m h_\varphi(m-2k)W_\varphi(j+1,m). \tag{7.4-10}$$

Равенства (7.4-9) и (7.4-10) обнаруживают замечательную взаимосвязь между коэффициентами ДВП соседних масштабов. Сопоставляя эти равенства с формулой (7.1-7), мы видим, что и коэффициенты приближения $W_\varphi(j,k)$, и коэффициенты деталей $W_\psi(j,k)$ масштаба j могут быть вычислены исходя из коэффициентов приближения $W_\varphi(j+1,k)$ масштаба $(j+1)$ с помощью операций свертки и прореживающей выборки. А именно достаточно вычислить свертки коэффициентов $W_\varphi(j+1,k)$ с обращенными во времени уточняющими последовательностями для масштабирующей функции $h_\varphi(-n)$ и для вейвлетов $h_\psi(-n)$, и оставить лишь члены с четными номерами. На рис. 7.17 эта процедура вычисления коэффициентов представлена в виде блок-схемы. Заметим, что она совпадает с блоком анализа представленной на рис. 7.6 системы двухканального субполосного кодирования и декодирования при $h_0(n) = h_\varphi(-n)$ и $h_1(n) = h_\psi(-n)$. Поэтому можно записать:

$$W_\psi(j,k) = h_\psi(-n) \bigstar W_\varphi(j+1,n)\Big|_{n=2k,k\geq 0} \tag{7.4-11}$$

и

$$W_\varphi(j,k) = h_\varphi(-n) \bigstar W_\varphi(j+1,n)\Big|_{n=2k,k\geq 0}, \tag{7.4-12}$$

где свертки вычисляются в моменты времени $n = 2k$ при $k \geq 0$. Как будет показано в примере 7.10, вычисление сверток для неотрицательных четных значений индексов эквивалентно фильтрации и прореживающей выборке с фактором 2.

> Если $h_\varphi(m - 2k)$ в (7.4-9) переписать в виде $h_\varphi(-(2k - m)$, можно увидеть, что первый знак минуса отвечает за инверсию порядка (см. уравнение (7.1-6)), $2k$ отвечает за понижение дискретизации (см. уравнение (7.1-2)), а m становится формальной переменной свертки (см. уравнение (7.1-7)).

Уравнения (7.4-11) и (7.4-12) являются определяющими для вычисления быстрого вейвлет-преобразования. Число требуемых операций для вычисления последовательности длиной $M = 2^J$ составляет порядка $O(M)$, т. е. пропорционально ее длине, поскольку число операций умножения и деления, необходимых для вычисления сверток, выполняемых в банке анализа БВП на рис. 7.17, пропорционально числу элементов последовательности. Таким образом, БВП имеет преимущество по сравнению с алгоритмом БПФ, который требует порядка $O(M \log_2 M)$ операций.

В заключение нашего обсуждения БВП отметим, что банк фильтров на рис. 7.17 может использоваться как базовый элемент итеративной многоступенчатой структуры для вычисления БВП-коэффициентов в двух или более последовательных масштабах. На рис. 7.18(*а*), например, представлен двухступенчатый банк фильтров для вычисления коэффициентов в двух наиболее крупных масштабах преобразования. Предполагается, что коэффициенты самого крупного масштабы суть отсчеты исходной функции. Таким образом, $W_\varphi(J, n) = f(n)$, где J — самый крупный масштаб. (Это означает, в соответствии с разделом 7.2.2, что $f(x) \in V_J$, т. е. V_J — то масштабирующее пространство, которому принадлежит функция $f(x)$.) Первый банк фильтров на рис. 7.18(*а*) раскладывает исходную функцию на низкочастотную составляющую (приближение), которая соответствует коэффициентам $W_\varphi(J - 1, n)$, и высокочастотную (детальную) составляющую, которая соответствует коэффициентам $W_\psi(J - 1, n)$. Это изображено графически на рис. 7.18(*б*), где масштабирующее пространство V_J разложено в подпространство вейвлетов W_{J-1} и масштабирующее пространство V_{J-1}. Спектр исходной функции оказывается разложенным на два частотных полудиапазона. Второй банк фильтров на рис. 7.18(*а*) раскладывает подпространство V_{J-1} на подпространства W_{J-2} и V_{J-2} с коэффициентами ДВП, равными $W_\psi(J - 2, n)$ и $W_\varphi(J - 2, n)$ соответственно. При этом нижний частотный полудиапазон в свою очередь раскладывается на два частотных полудиапазона.

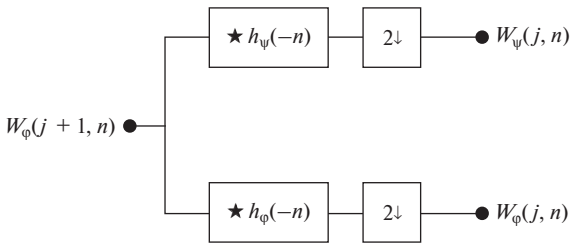


Рис. 7.17. Банк анализа БВП

а
б

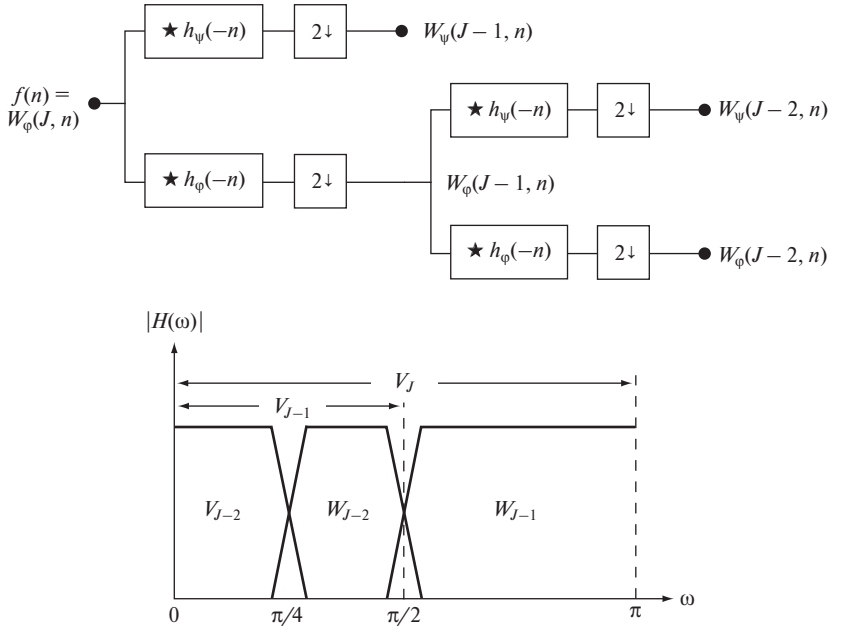


Рис. 7.18. (*а*) Двухступенчатый или двухмасштабный БВП-банк анализа и (*б*) его свойства в отношении разделения частот

Двухступенчатый банк фильтров на рис. 7.18(*а*) легко продолжить так, чтобы получить разложение на любое число масштабов. Банк фильтров третьей ступени, например, будет оперировать с коэффициентами $W_\varphi(J-2, n)$ и разлагать подпространство V_{J-2} на подпространства W_{J-3} и V_{J-3}, что отвечает разложению нижней четверти спектра исходного сигнала на частотные полудиапазоны. В обычной ситуации мы берем 2^J отсчетов функции $f(x)$ и используем P банков фильтров (таких как на рис. 7.17), чтобы получить P-масштабное БВП в масштабах $J-1$, $J-2$, ..., $J-P$. Коэффициенты для наиболее крупного масштаба $J-1$ вычисляются вначале, для наиболее мелкого масштаба $J-P$ — в конце. Если частота дискретизации функции $f(x)$ порядка частоты Найквиста, что обычно и имеет место, то отсчеты функции являются хорошим приближением для коэффициентов того масштаба, разрешение которого соответствует шагу дискретизации. Эти отсчеты могут быть использованы на входе алгоритма в качестве начальных масштабных коэффициентов (коэффициентов приближения) высокого разрешения. Другими словами, вейвлет-коэффициенты (коэффициенты деталей) в масштабе, который определяется шагом дискретизации, не нужны. Масштабирующие функции самого крупного масштаба ведут себя в формулах (7.3-5) и (7.3-6) подобно единичным дискретным функциям, что позволяет использовать отсчеты $f(n)$ в качестве приближения J-го масштаба или масштабных коэффициентов на входе первого двухканального банка фильтров [Odegard, Gopinath, Burrus, 1992].

Пример 7.10. Вычисление одномерного быстрого вейвлет-преобразования.
■ Для иллюстрации представленной концепции рассмотрим дискретную функцию $f(n) = \{1, 4, -3, 0\}$ из примера 7.8. Как и в том примере, мы снова вычислим

преобразование относительно масштабирующих функций и вейвлетов Хаара. Теперь, однако, мы не будем непосредственно использовать функции разложения, как это было сделано в примере 7.8, а воспользуемся уточняющими последовательностями из примеров 7.5 и 7.6:

$$h_{\varphi}(n)=\begin{cases}1/\sqrt{2} & n=0,1,\\ 0 & \text{в остальных случаях}\end{cases} \tag{7.4-13}$$

и

$$h_{\psi}(n)=\begin{cases}1/\sqrt{2} & n=0,\\ -1/\sqrt{2} & n=1,\\ 0 & \text{в остальных случаях.}\end{cases} \tag{7.4-14}$$

Эти функции используются при построении банка фильтров БВП; они определяют коэффициенты фильтров. Заметим, что, поскольку масштабирующие функции и вейвлеты Хаара являются ортонормированными, уравнение (7.1-14) может быть использовано для получения коэффициентов фильтров БВП из единственного фильтра-прототипа — аналогично значениям $h_{\varphi}(n)$ в табл. 7.2, которые соответствуют $g_0(n)$ в уравнении (7.1-14).

Таблица 7.2. Коэффициенты ортонормированного фильтра Хаара для $h_{\varphi}(n)$

n	$h_{\varphi}(n)$
0	$1/\sqrt{2}$
1	$1/\sqrt{2}$

Поскольку посчитанное в примере 7.8 ДВП состояло из элементов $\{W_{\varphi}(0,0),\ W_{\psi}(0,0),\ W_{\psi}(1,0),\ W_{\psi}(1,1)\}$, мы вычислим соответствующее БВП для масштабов $j=\{0,1\}$. Итак, $J=2$ (всего имеется $2^J=2^2$ отсчетов) и $P=2$ (мы работаем с масштабами $J-1=2-1=1$ и $J-P=2-2=0$). Для вычисления преобразования будет использован двухступенчатый банк фильтров, показанный на рис. 7.18(*а*). На рис. 7.19 представлены последовательности, получаемые в результате применения операций свертки и прореживающей выборки, которые составляют БВП. На вход самого левого банка фильтров подается функция $f(n)$. Чтобы вычислить коэффициенты $W_{\psi}(1,k)$, которые появляются в конце верхней ветви на рис. 7.17, сначала вычисляется свертка функции $f(n)$ с функцией $h_{\psi}(-n)$. Как было объяснено в разделе 3.4.2, для этого нужно отразить одну из функций относительно начала координат, сдвинуть ее относительно другой и вычислить сумму значений произведения двух этих функций во всех точках. Для последовательностей $\{1, 4, -3, 0\}$ и $\{-1/\sqrt{2}, 1/\sqrt{2}\}$ это дает результат

$$\{-1/\sqrt{2}, -3/\sqrt{2}, 7/\sqrt{2}, -3/\sqrt{2}, 0\},$$

причем второй член соответствует значению индекса $k=2n=0$. (На рис. 7.19 подчеркнутые элементы отвечают отрицательным значениям индекса, т. е. $n<0$.) После прореживающей выборки (т. е. после удержания только членов, отвечающих четным значениям индекса) мы имеем $W_{\psi}(1,k)=\{-3/\sqrt{2}, -3/\sqrt{2}\}$ при $k=\{0,1\}$. Альтернативный способ вычисления состоит в использовании выражения (7.4-9):

$$W_\psi(1,k) = h_\psi(-n) \star W_\varphi(2,n)\Big|_{n=2k,k\geq 0} = h_\psi(-n) \star f(n)\Big|_{n=2k,k\geq 0} =$$

$$= \sum_l h_\psi(l-2k) f(l)\Big|_{k=0,1} = \frac{1}{\sqrt{2}}\left(f(2k) - f(2k+1)\right)\Big|_{k=0,1}.$$

Здесь мы подставили индекс $2k$ вместо индекса n в свертку и использовали индекс l как индекс суммирования. Из всей суммы остается лишь два слагаемых, поскольку лишь два члена последовательно-инвертированной масштабной последовательности для вейвлетов $h(-n)$ отличны от нуля. При $k = 0$ мы получаем $W_\psi(1,0) = -3/\sqrt{2}$, а при $k = 1$ получаем $W_\psi(1,1) = -3/\sqrt{2}$. Таким образом, после фильтрации и прореживающей выборки мы имеем последовательность $\{-3/\sqrt{2}, -3/\sqrt{2}\}$, что совпадает с полученным ранее результатом. Остальные операции свертки и прореживающей выборки осуществляются аналогично. ■

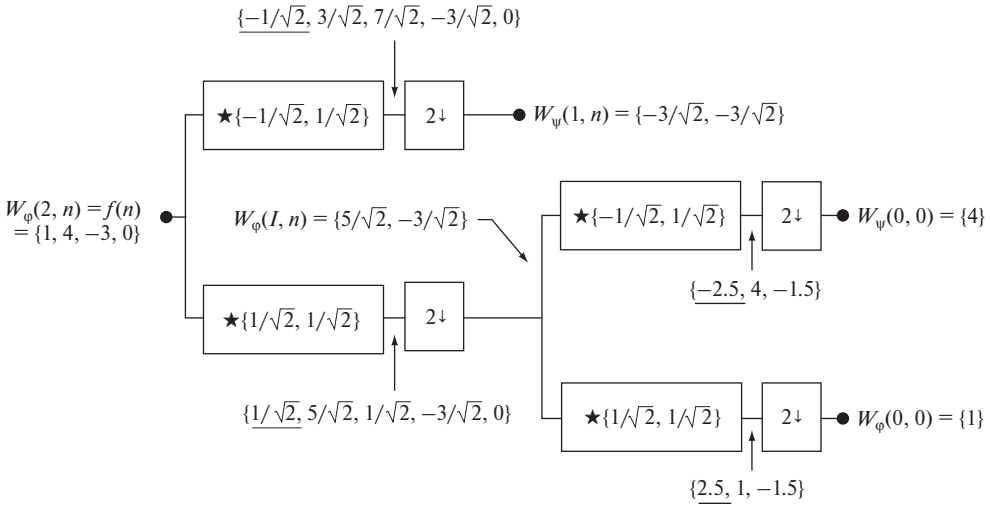


Рис. 7.19. Вычисление двухмасштабного быстрого вейвлет-преобразования последовательности {1, 4, –3, 0} с использованием уточняющих последовательностей для масштабирующей функции и вейвлетов Хаара

Как и можно было ожидать, возможно построить столь же эффективную обратную процедуру, восстанавливающую функцию $f(x)$ из результатов, полученных при прямом преобразовании. Эта процедура, называемая *обратное быстрое вейвлет-преобразование* (*обратное* БВП), использует уточняющие последовательности прямого преобразования и коэффициенты приближения и деталей масштаба j для получения коэффициентов приближения масштаба $j + 1$. Па-

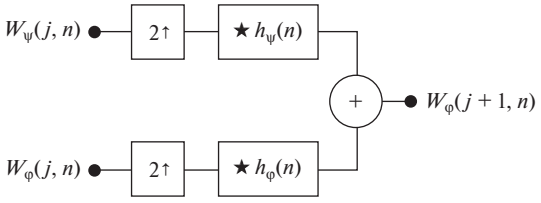


Рис. 7.20. Банк фильтров синтеза обратного БВП

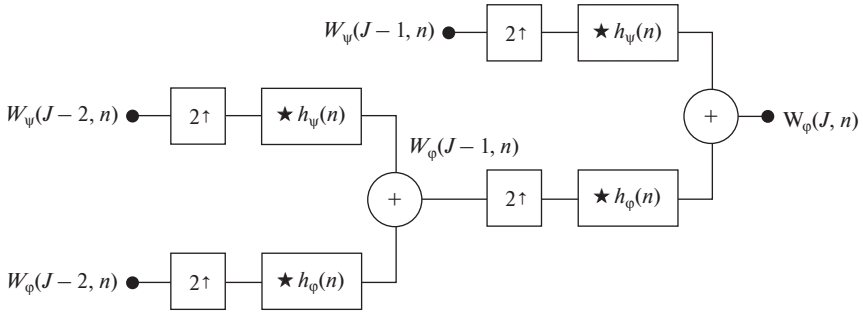


Рис. 7.21. Двухступенчатый или двухмасштабный банк синтеза обратного БВП

мятуя о сходстве банка анализа для БВП на рис. 7.17 и банка анализа в схеме двухканального субполосного кодирования и декодирования на рис. 7.6(*a*), мы легко можем предположить, какой вид имеет *банк фильтров синтеза* для обратного БВП. На рис. 7.20 представлена структура такого банка, идентичного банку синтеза на рис. 7.6(*a*). Условия (7.1-14) раздела 7.1.2 определяют подходящие фильтры синтеза. Как отмечено в указанном разделе, для точного восстановления (в случае ортонормированных квадратурных фильтров) требуется выполнение условия $g_i(n) = h_i(-n)$ при $i = \{0,1\}$. Таким образом, фильтры синтеза должны быть последовательно-инвертированными копиями фильтров анализа и наоборот. Поскольку фильтрами анализа для БВП являются последовательности $h_0(n) = h_\varphi(-n)$ и $h_1(n) = h_\psi(-n)$ (см. рис. 7.17), то фильтрами синтеза для обратного БВП должны быть последовательности $g_0(n) = h_0(-n) = h_\varphi(n)$ и $g_1(n) = h_1(-n) = h_\psi(n)$. Следует, однако, напомнить, что возможно также использование биортогональных фильтров анализа и синтеза, которые не являются последовательно-инвертированными копиями друг друга. Биортогональные фильтры анализа и синтеза являются перекрестно-модулированными (см. (7.1-10) и (7.1-11)).

Банк фильтров синтеза обратного БВП реализует вычисления

$$W_\varphi(j+1,k) = h_\varphi(k) \star W_\varphi^{2\uparrow}(j,k) + h_\psi(k) \star W_\psi^{2\uparrow}(j,k)\Big|_{k\geq 0}, \qquad (7.4\text{-}15)$$

где $W^{2\uparrow}$ обозначает применение сгущающей выборки с фактором 2 (т. е. добавление нулей между коэффициентами W, как определялось в (7.1-1), в результате чего полное число коэффициентов увеличивается вдвое). Для получения коэффициентов приближения более крупного масштаба предварительно «сгущенные» коэффициенты сворачиваются с последовательностями $h_\varphi(n)$ и $h_\psi(n)$ и складываются. При этом, в сущности, формируется более точное приближение последовательности $f(n)$, более подробное и с более высоким разрешением. Как и в случае прямого БВП, описанная операция обратного БВП может применяться повторно. На рис. 7.21 изображена структура двухступенчатого (двухмасштабного) банка фильтров обратного БВП. Такая процедура вычисления коэффициентов может быть распространена на любое число масштабов и гарантирует точное восстановление последовательности $f(n)$.

Следует помнить, что, как и в пирамидальном кодировании (см. раздел 7.1.1), вейвлет-преобразования могут вычисляться для задаваемого числа масштабов. Например, для изображения размерами $2^J \times 2^J$ существуют $1 + \log_2 J$ возможных масштабов.

Пример 7.11. Вычисление одномерного обратного быстрого вейвлет-преобразования.

■ Процесс вычисления обратного быстрого вейвлет-преобразования является «зеркальным отражением» процесса вычисления прямого преобразования. Для рассмотренной в примере 7.10 последовательности этот процесс изображен на рис. 7.22. Прежде всего к коэффициентам приближения и деталей уровня (масштаба) 0 применяется сгущающая выборка, что дает соответственно {1,0} и {4,0}. Свертка с фильтрами $g_0(n)=h_\varphi(n)=\{1/\sqrt{2},1/\sqrt{2}\}$ и $g_1(n)=h_\psi(n)=\{1/\sqrt{2},-1/\sqrt{2}\}$ дает $\{1/\sqrt{2},1/\sqrt{2},0\}$ и $\{4/\sqrt{2},-4/\sqrt{2},0\}$; при последующем сложении получаем $W_\varphi(1,n)=\{5/\sqrt{2},-3/\sqrt{2}\}$. Тем самым получено приближение уровня 1 на рис. 7.22, совпадающее с приближением того же уровня на рис. 7.19. Аналогично с помощью второго банка фильтров синтеза (в правой части на рис. 7.22) строится вся последовательность $f(n)$. ■

В заключение обсуждения быстрого вейвлет-преобразования заметим, что в то время как базисные функции преобразования Фурье (т. е. синусоидальные функции) гарантируют существование БПФ-алгоритма, существование БВП-алгоритма зависит как от наличия масштабирующей функции для используемой системы вейвлетов, так и от ортогональности (или биортогональности) системы масштабирующих функций и соответствующих вейвлетов. Так, для заданного формулой (7.3-12) вейвлета «мексиканская шляпа» не существует подходящей масштабирующей функции, и вычисления для этой системы не могут быть осуществлены с помощью БВП. Другими словами, для вейвлета «мексиканская шляпа» невозможно построить банк фильтров типа представленного на рис. 7.17, который бы удовлетворял основополагающему требованию БВП-метода.

Наконец отметим, что хотя временна́я и частотная области часто рассматриваются как различные пространства, тем не менее они между собой неразрывно связаны. При попытке анализировать функцию одновременно во временно́й и частотной областях мы сталкиваемся со следующей проблемой. Если нам не-

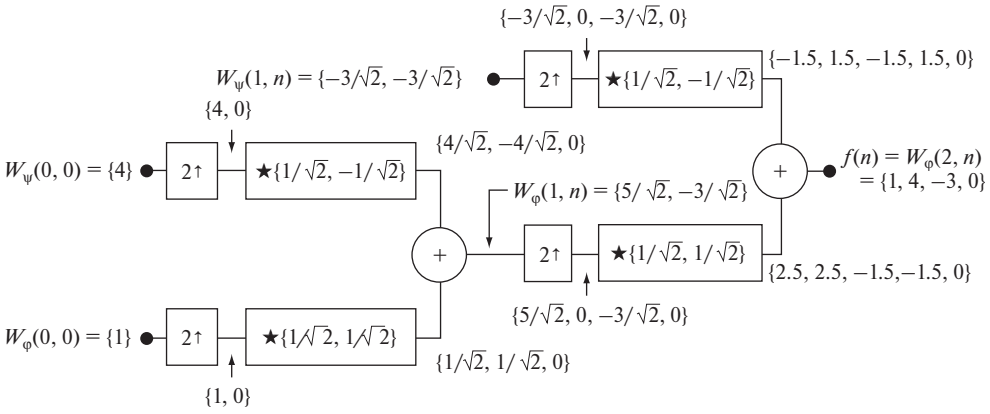


Рис. 7.22. Вычисление двухмасштабного обратного быстрого вейвлет-преобразования последовательности $\{1,4,-1.5\sqrt{2},-1.5\sqrt{2}\}$ с использованием уточняющих последовательностей для масштабирующей функции и вейвлетов Хаара

обходима точная временна́я информация, то мы обязаны допустить некоторую неопределенность в частотной информации, и наоборот. Это есть принцип *неопределенности Гейзенберга* в приложении к задачам обработки информации. Для того чтобы графически проиллюстрировать этот принцип, нужно схематически рассматривать каждую базисную функцию, используемую для представления всех остальных функций, как *элемент покрытия* в *частотно-временно́й плоскости*. Элемент покрытия, называемый также *ячейкой Гейзенберга*, показывает частотное наполнение базисной функции, которое она представляет, а также положение базисной функции во времени. Ортонормированные базисные функции характеризуются тем, что их ячейки не пересекаются.

На рис. 7.23 изображены частотно-временные покрытия для следующих базисов: (*а*) базис импульсной функции (во временно́й области), (*б*) базис синусоидальных функций (БПФ-базис) и (*в*) БВП-базис. Каждый элемент покрытия представлен прямоугольной областью на рис. 7.23(*а*)—(*в*); высота и ширина каждой из областей определяют частотные и временные характеристики функций, которые могут быть представлены с использованием указанных базисных функций. Заметим, что обычный временно́й базис на рис. 7.23(*а*) точно указывает момент времени, когда произошло событие, но не дает никакой частотной информации (ширина каждого из прямоугольников на рис. 7.23(*а*) должна считаться как один момент времени). Таким образом, чтобы представить синусоиду одной частоты как разложение при помощи базиса импульсных функций, требуется использование всех базисных функций. С другой стороны, БПФ-базис на рис. 7.23(*б*) точно указывает частотные составляющие события, которое происходит внутри длительного интервала времени, но не обеспечивает вовсе разрешения по времени (высота каждого из прямоугольников на рис. 7.23(*б*) должна считаться как одна частота). Таким образом, синусоида одной частоты, которая представлялась бесконечным числом импульсных базисных функций, может быть представлена как разложение, требующее лишь одну базисную функцию БПФ. Временно́е и частотное разрешение БВП-ячеек (прямоугольников) на рис. 7.23(*в*) различны, но их площади одинаковы. Это значит, что каждая из них представляет одинаковую часть частотно-временно́й плоскости.

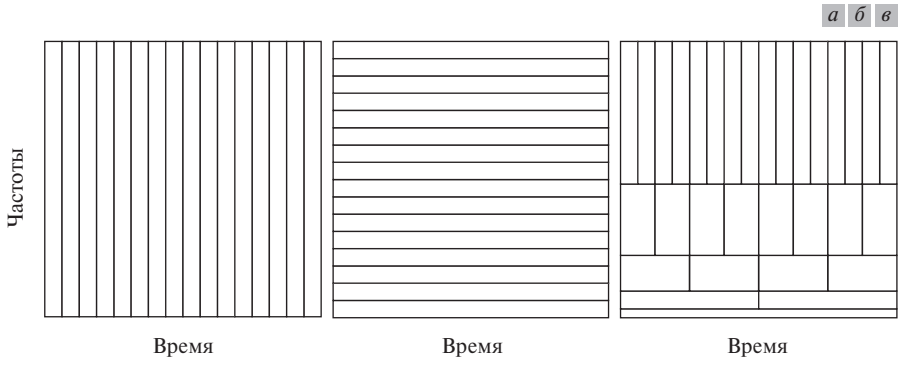


Рис. 7.23. Частотно-временные покрытия для (*а*) временно́го, (*б*) БПФ- и (*в*) БВП-базисов функций. Обратите внимание, что на рисунке (*в*) каждая горизонтальная полоса прямоугольников равной высоты представляет свой масштаб БВП

При низких частотах ячейки имеют малую высоту (т. е. имеют высокое частотное разрешение или меньшую неопределенность по частоте), но большую ширину (что соответствует низкому временно́му разрешению или бóльшей неопределенности во времени). При высоких частотах ширина ячеек меньше (что отвечает более высокому временно́му разрешению), а высота больше (что означает меньшее частотное разрешение). Тем самым базисные функции БВП создают компромисс между двумя пограничными случаями на рис. 7.23(*а*) и (*б*). Рассмотренное фундаментальное различие между БПФ и БВП отмечалось во введении к настоящей главе и играет важную роль при анализе нестационарных процессов, когда частоты изменяются во времени.

7.5. Двумерные вейвлет–преобразования

Рассмотренные в предыдущем разделе одномерные вейвлет-преобразования нетрудно обобщить на двумерные функции (изображения). В двумерном случае необходимо иметь двумерную масштабирующую функцию $\varphi(x,y)$ и три двумерных вейвлет-функции $\psi^H(x,y)$, $\psi^V(x,y)$ и $\psi^D(x,y)$. Каждая функция представляет собой произведение одномерной масштабирующей функции φ и соответствующей вейвлет-функции ψ. Если исключить те произведения, которые приводят к «одномерным результатам» типа $\varphi(x)\psi(x)$, то оставшиеся образуют *разделимую* масштабирующую функцию

$$\varphi(x,y)=\varphi(x)\varphi(y) \tag{7.5-1}$$

и разделимые «направленные» вейвлет-функции

$$\psi^H(x,y)=\psi(x)\varphi(y), \tag{7.5-2}$$

$$\psi^V(x,y)=\varphi(x)\psi(y), \tag{7.5-3}$$

$$\psi^D(x,y)=\psi(x)\psi(y). \tag{7.5-4}$$

Эти вейвлеты измеряют вариации значений функции — изменения яркости для изображений — по разным направлениям: ψ^H измеряет вариации вдоль столбцов (связанные, например, с горизонтальными краями объектов), ψ^V — вдоль строк (вертикальные края) и ψ^V — вдоль диагоналей. Диагональная чувствительность является естественным следствием свойства разделимости, выраженного формулами (7.5-2)—(7.5-4); она не приводит к увеличению вычислительной сложности рассматриваемого в этом разделе двумерного преобразования.

Если заданы двумерные разделимые масштабирующая функция и вейвлет-функции, то дальнейшее обобщение одномерного БВП на двумерный случай не представляет труда. Прежде всего определим семейство базисных функций с помощью операций сдвигов и изменений масштаба:

$$\varphi_{j,m,n}(x,y)=2^{j/2}\varphi(2^j x-m,2^j y-n), \tag{7.5-5}$$

$$\psi^i_{j,m,n}(x,y)=2^{j/2}\psi^i(2^j x-m,2^j y-n),\qquad i=\{H,V,D\}, \tag{7.5-6}$$

причем верхний индекс i не является показателем степени, а служит для идентификации направленных вейвлетов, заданных формулами (7.5-2)—(7.5-4), и принимает значения H, V или D. После этого определим дискретное вейвлет-преобразование изображения $f(x,y)$ размерами $M \times N$ следующим образом:

$$W_\varphi(j_0,m,n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \varphi_{j_0,m,n}(x,y), \tag{7.5-7}$$

$$W_\psi^i(j,m,n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \psi_{j,m,n}^i(x,y), \qquad i = \{H,V,D\}. \tag{7.5-8}$$

Поскольку теперь мы имеем дело с двумерными изображениями, $f(x,y)$ является дискретной функцией или последовательностью значений, а x и y — дискретными переменными. Масштабирующие функции и вейвлеты в (7.5-7) и (7.5-8) являются дискретизованными на протяжении своего носителя (как это было в одномерном случае в разделе 7.3.2).

Как и в одномерном случае, j_0 — произвольный начальный масштаб, и коэффициенты $W_\varphi(j_0,m,n)$ определяют приближение функции $f(x,y)$ в масштабе j_0. Коэффициенты $W_\psi^i(j,m,n)$ определяют горизонтальные, вертикальные и диагональные детали для масштабов $j \geq j_0$. Обычно мы полагаем $j_0 = 0$ и выбираем $N = M = 2^J$, так что $j = 0, 1, 2, \ldots, J-1$ и $m, n = 0, 1, 2, \ldots, 2^J - 1$. Исходная функция $f(x,y)$ может быть восстановлена по заданным коэффициентам W_φ и W^i_ψ в (7.5-7) и (7.5-8) при помощи обратного дискретного вейвлет-преобразования

$$\begin{aligned} f(x,y) = {} & \frac{1}{\sqrt{MN}} \sum_m \sum_n W_\varphi(j_0,m,n) \varphi_{j_0,m,n}(x,y) + \\ & + \frac{1}{\sqrt{MN}} \sum_{i=H,V,D} \sum_{j=j_0}^{+\infty} \sum_m \sum_n W_\psi^i(j,m,n) \psi_{j,m,n}^i(x,y). \end{aligned} \tag{7.5-9}$$

Как и одномерное дискретное вейвлет-преобразование, двумерное ДВП может быть реализовано с помощью операций фильтрации и прореживающей выборки. Поскольку используемые масштабирующая функция и вейвлет-функции являются разделимыми, то сначала вычисляется одномерное БВП по строкам от функции $f(x,y)$, а затем вычисляется одномерное БВП по столбцам от полученного результата. На рис. 7.24(*а*) представлена блок-схема такой процедуры. Заметим, что, как и в случае своего одномерного аналога на рис. 7.17, для получения коэффициентов приближения и деталей в масштабе j двумерное БВП оперирует с коэффициентами приближения масштаба $j + 1$. В двумерном случае, однако, мы имеем дело с тремя наборами коэффициентов деталей — горизонтальным, вертикальным и диагональным.

Одномасштабный банк фильтров на рис. 7.24(*а*) может применяться повторно (для чего коэффициенты приближения на выходе этого банка фильтров нужно подать на вход такого же следующего банка фильтров), результатом чего будет P-масштабное преобразование с масштабами $j = J-1, J-2, \ldots, J-P$. Как и в одномерном случае, изображение $f(x,y)$ используется в качестве коэффициентов $W_\varphi(J,m,n)$ на входе. Сверткой строк изображения с последовательностями $h_\varphi(-n)$ и $h_\psi(-n)$ и прореживанием столбцов полученного результата мы

а
б
в

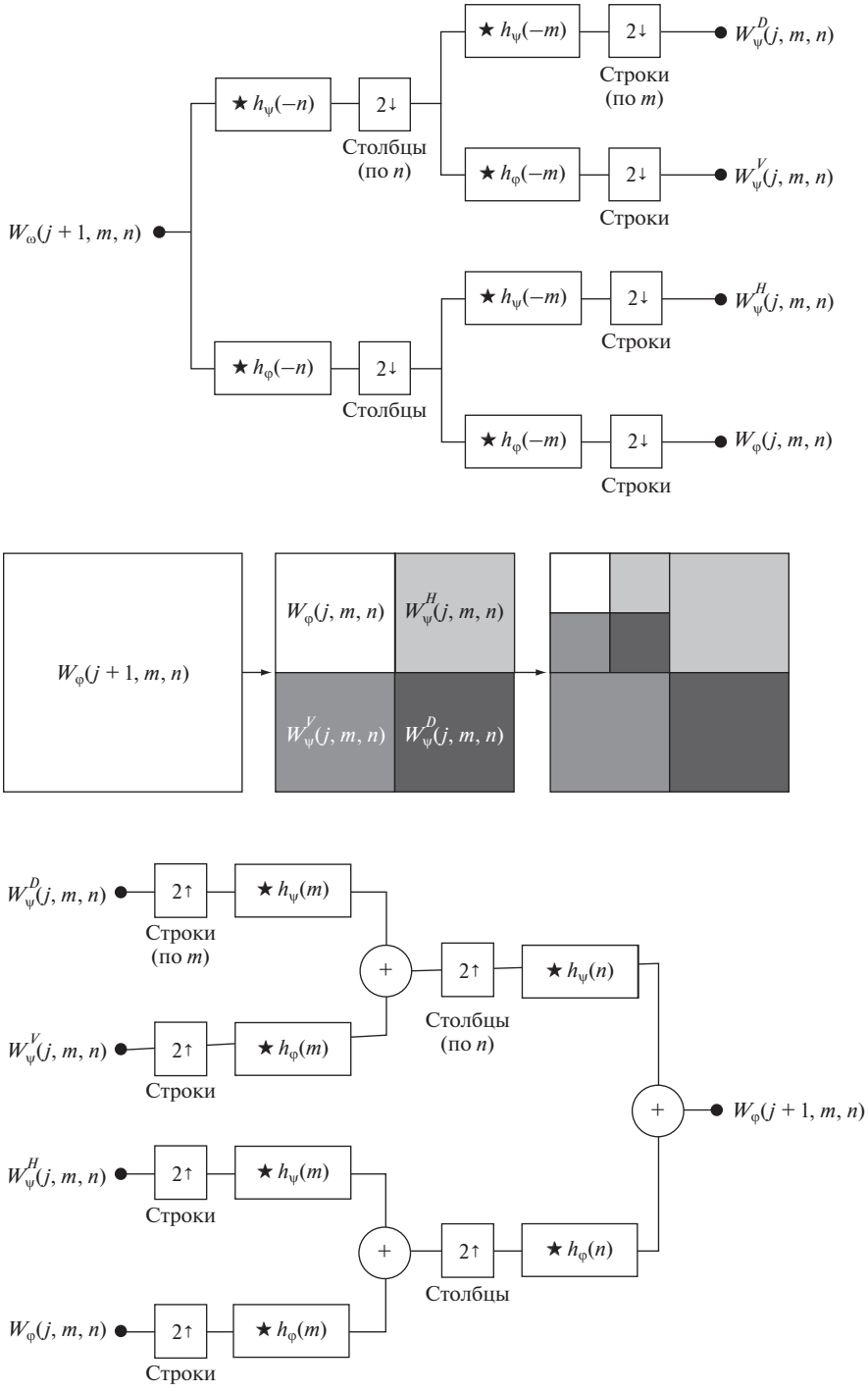


Рис. 7.24. Двумерное быстрое вейвлет-преобразование: (*а*) банк фильтров анализа; (*б*) получаемое разложение и (*в*) банк фильтров синтеза

получаем две части изображения с уменьшенным вдвое разрешением по горизонтали. Высокочастотная или детальная часть характеризует высокочастотную в вертикальном направлении составляющую изображения. Низкочастотная часть или приближение содержит информацию о низких в вертикальном направлении частотах. К обеим частям изображения затем применяется процедура фильтрации по столбцам и прореживание. Это дает на выходе четыре изображения (четыре части исходного изображения) — W_φ, W_ψ^H, W_ψ^V и W_ψ^D — каждое вдвое меньших линейных размеров, чем исходное. Эти изображения, показанные в центре рис. 7.24(*б*), представляют собой результат применения прореживающей выборки с коэффициентом 2 по обоим направлениям к массиву, элементами которого являются скалярные произведения изображения $f(x, y)$ и двумерных масштабирующих функций и вейвлет-функций соответствующего масштаба. Две итерации такой процедуры дают двухмасштабное разложение, показанное справа на рис. 7.24(*б*).

> Уточним, как W_φ, W_ψ^H, W_ψ^V и W_ψ^D расположены на рис. 7.24(*б*). Для каждого вычисляемого масштаба они замещают изображение коэффициентов приближения предыдущего масштаба, на основе которого они базируются.

На рис. 7.24(*в*) представлен банк фильтров синтеза, который служит для обращения только что рассмотренной процедуры. Как и следовало ожидать, алгоритм восстановления в двумерном случае аналогичен алгоритму восстановления в одномерном случае. Каждая итерация заключается в применении к четырем частям изображения (приближению и трем детальным частям) масштаба j операций сгущающей выборки и свертки с двумя одномерными фильтрами, один из которых действует на столбцы, а другой на строки. Сложение полученных таким образом результатов дает приближение масштаба $j + 1$, после чего процедура повторяется до тех пор, пока исходное изображение не будет восстановлено.

Пример 7.12. Вычисление двумерного быстрого вейвлет-преобразования.

■ На рис. 7.25(*а*) представлено сгенерированное на компьютере изображение размерами 128×128 элементов, состоящее из двумерных синусоподобных импульсов на черном фоне. Цель данного примера состоит в иллюстрации подробностей вычисления двумерного БВП данного изображения. На рис. 7.25(*б*)—(*г*) показаны три этапа выполнения БВП изображения рис. 7.25(*а*). Для их получения использовались двумерный банк фильтров на рис. 7.24(*а*) и фильтры разложения, показанные на рис. 7.26(*а*)—(*б*).

> Масштабирующие векторы и вейвлеты, использованные в данном примере, описаны ниже. Мы сосредоточимся здесь лишь на подробностях вычисления преобразований, которые не зависят от используемых коэффициентов фильтра.

Рис. 7.25(*б*) — одномасштабное БВП изображения на рис. 7.25(*а*); для его получения исходное изображение подавалось на вход банка фильтров на рис. 7.24(*а*). Полученные на выходе четыре результата разложения вдвое меньших размеров (т. е. приближение, горизонтальная, вертикальная и диагональная детальные

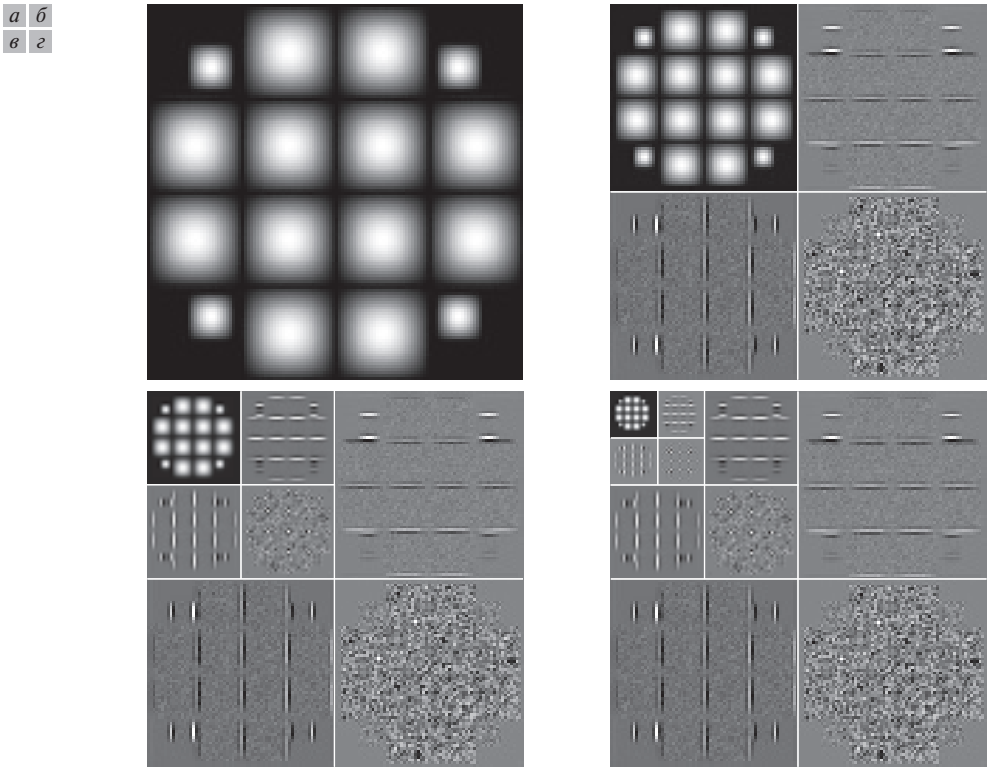


Рис. 7.25. Вычисление двумерного трехмасштабного БВП: (*а*) исходное изображение; (*б*) одномасштабное БВП; (*в*) двухмасштабное БВП; (*г*) трехмасштабное БВП

части) расположены в соответствии с рис. 7.24(*б*), сформировав изображение на рис. 7.25(*б*). Аналогичная процедура использовалась для получения двухмасштабного БВП на рис. 7.25(*в*), но на вход банка фильтров подавалось субизображение приближения, показанное в левом верхнем углу рис. 7.25(*б*) и имеющее вдвое меньший масштаб (1/4 площади исходного изображения). Как видно на рис. 7.25(*в*), это субизображение затем было в свою очередь заменено четырьмя еще вдвое уменьшенными результатами разложения (вчетверо меньшего масштаба или составляющих 1/16 площади исходного изображения), полученными на втором шаге фильтрации. Наконец, на рис. 7.25(*г*) представлено трехмасштабное БВП, которое было сформировано после того, как на вход банка фильтров было подано субизображение приближения из левого верхнего угла рис. 7.25(*в*). На каждом проходе через банк фильтров формируются четыре субизображения вдвое меньших линейных размеров, которые замещают на рисунке то изображение, из которого они были получены. Обратите внимание на характерную ориентацию изображений детальных составляющих W_{ψ}^{H}, W_{ψ}^{V} и W_{ψ}^{D} в каждом масштабе. ■

Использованные в предыдущем примере фильтры разложения принадлежат хорошо известному семейству симметричных вейвлетов, называемых для

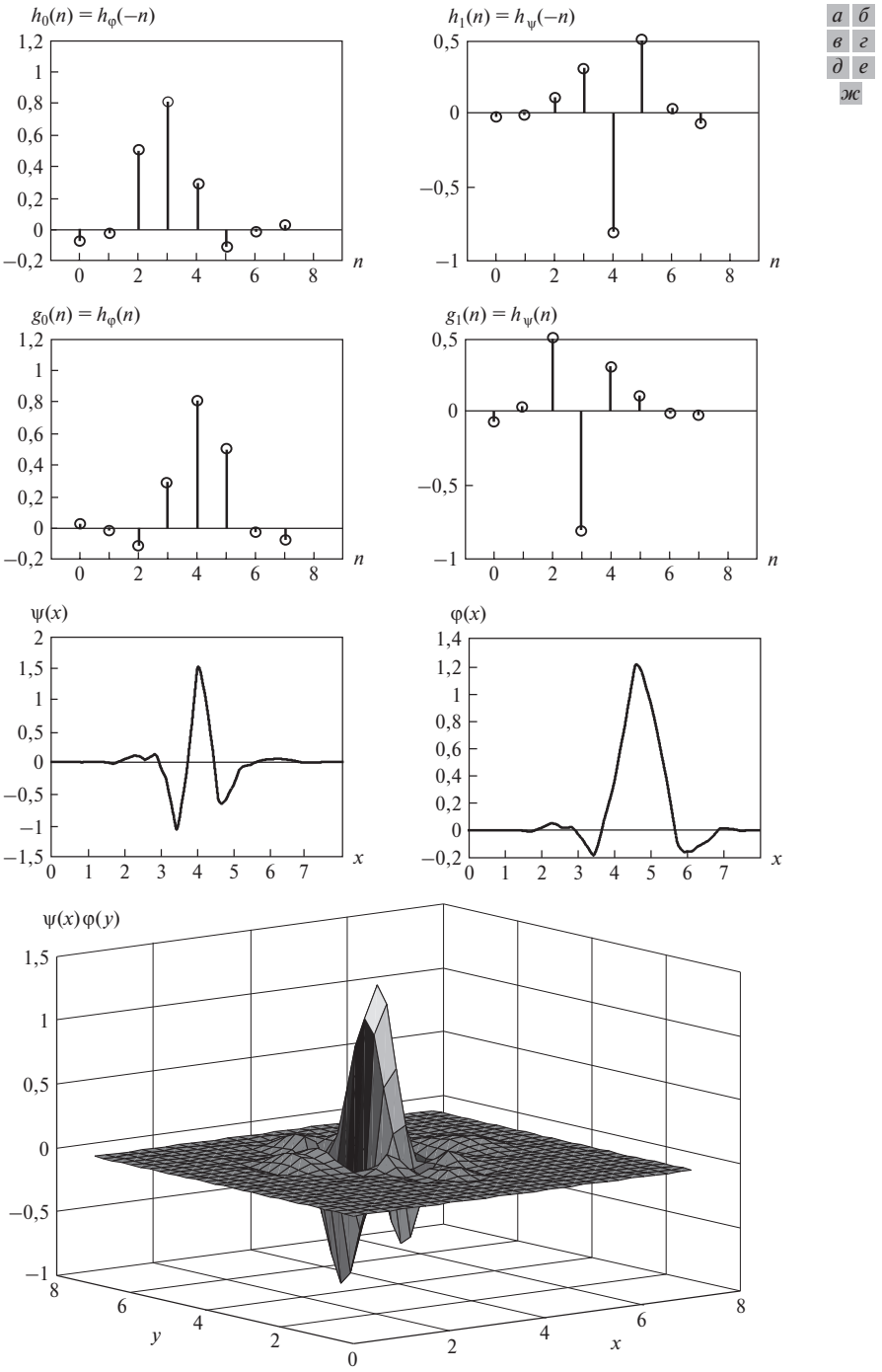


Рис. 7.26. Симметричные вейвлеты (симлеты) четвертого порядка: (*а*)—(*б*) фильтры разложения; (*в*)—(*г*) фильтры восстановления; (*д*) одномерная вейвлет-функция; (*е*) одномерная масштабирующая функция и (*ж*) один из трех двумерных вейвлетов $\psi H(x, y)$

краткости *симлетами*. Хотя симлеты не являются идеально симметричными, они сконструированы так, чтобы при заданном компактном носителе обладать минимально возможной асимметрией и наибольшим возможным числом обращающихся в нуль моментов[23] [Daubechies, 1992]. На рис. 7.26(*д*) и (*е*) представлены одномерные симлеты четвертого порядка (т. е. вейвлет и масштабирующая функция). На рис. 7.26(*а*)—(*г*) представлены соответствующие фильтры разложения и восстановления. Коэффициенты низкочастотного фильтра восстановления $g_0(n) = h_\varphi(n)$ при $0 \le n \le 7$ приведены в табл. 7.3. Коэффициенты остальных ортонормированных фильтров могут быть получены с помощью (7.1-14). На рис. 7.26(*ж*) представлено трехмерное изображение вейвлета $\psi^H(x, y)$. Оно служит для иллюстрации того, как в результате комбинации одномерных масштабирующей функции и вейвлет-функций формируется двумерный разделимый вейвлет.

Напомним, что компактным носителем функции является интервал, на котором данная функция может принимать ненулевые значения.

Таблица 7.3. Коэффициенты ортонормированного фильтра (симлета) четвертого порядка для $h_\varphi(n)$ ([Daubechies, 1992])

n	$h_\varphi(n)$
0	0,0322
1	−0,0126
2	−0,0992
3	0,2979
4	0,8037
5	0,4976
6	0,0296
7	−0,0758

В заключение раздела приведем два примера, иллюстрирующие полезность вейвлетов при обработке изображений. Как и при использовании Фурье-анализа, принцип состоит в следующем:

1. Вычисление двумерного вейвлет-преобразования изображения;
2. Изменение полученного преобразования;
3. Вычисление обратного преобразования.

Поскольку используемые уточняющие последовательности для масштабирующей функции и вейвлетов представляют собой низкочастотные и высокочастотные фильтры, то большинству методов обработки данных, основанных на Фурье-анализе, соответствуют аналогичные методы, основанные на вейвлет-анализе.

[23] K-й момент вейвлета $\psi(x)$ определяется как $m(k) = \int x^k \psi(x)\,dx$. Наличие нулевых моментов тесно связано с гладкостью масштабирующей функции и вейвлет-функции, а также с возможностью представления их в виде полиномов. Симлет порядка N имеет N обращающихся в нуль моментов.

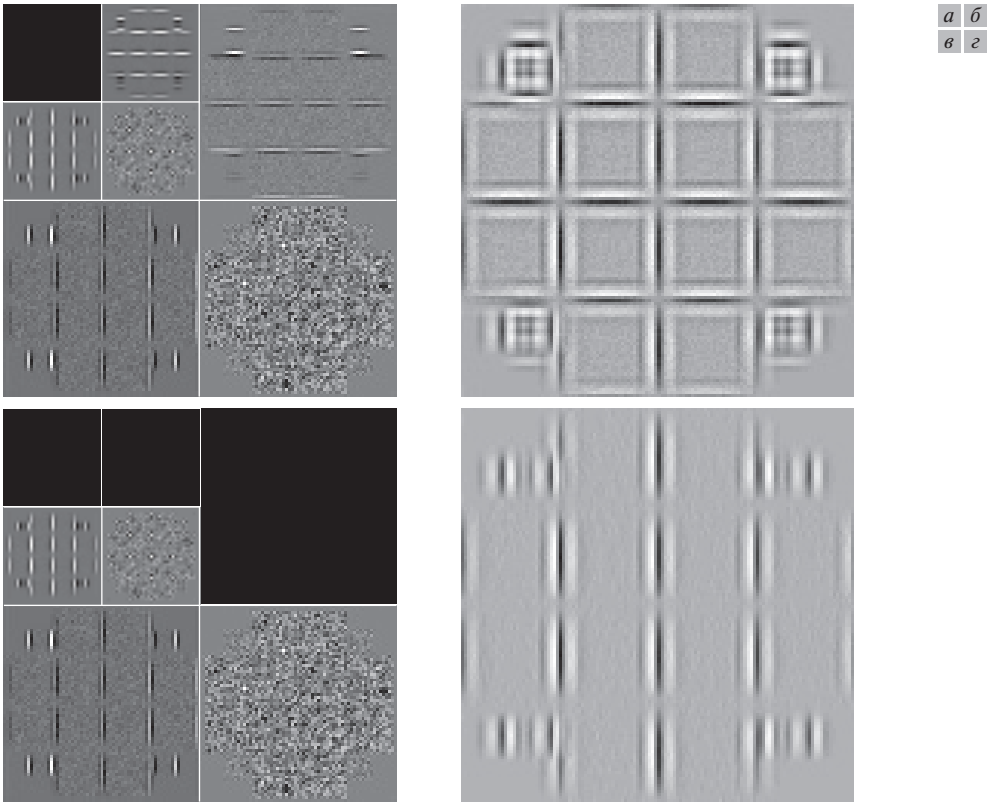


Рис. 7.27. Изменение коэффициентов ДВП для обнаружения границ объектов: (*а*) и (*в*) двухмасштабные разложения, определенные коэффициенты которых обращены в нуль; (*б*) и (*г*) соответствующие результаты восстановления

Пример 7.13. Обнаружение контуров на основе вейвлет-анализа.

■ Рис. 7.27 служит простой иллюстрацией указанных выше трех шагов. На рис. 7.27(*а*) составляющая приближения самого мелкого масштаба, представленного на рис. 7.25(*в*) дискретного вейвлет-преобразования, удалена путем обращения в нуль ее значений. Как видно на рис. 7.27(*б*), чистый эффект от вычисления обратного вейвлет-преобразования при использовании этих модифицированных коэффициентов заключается в подчеркивании контуров, что напоминает повышение резкости изображения при помощи Фурье-преобразования, обсуждаемое в разделе 4.9. Отметим, насколько хорошо очерчены границы между сигналом и фоном, несмотря на тот факт, что они являются сравнительно мягкими синусоидальными переходами. Если обратить в нуль также горизонтальную детальную составляющую (см. рис. 7.27(*в*) и (*г*)) то можно добиться выделения вертикальных контуров. ■

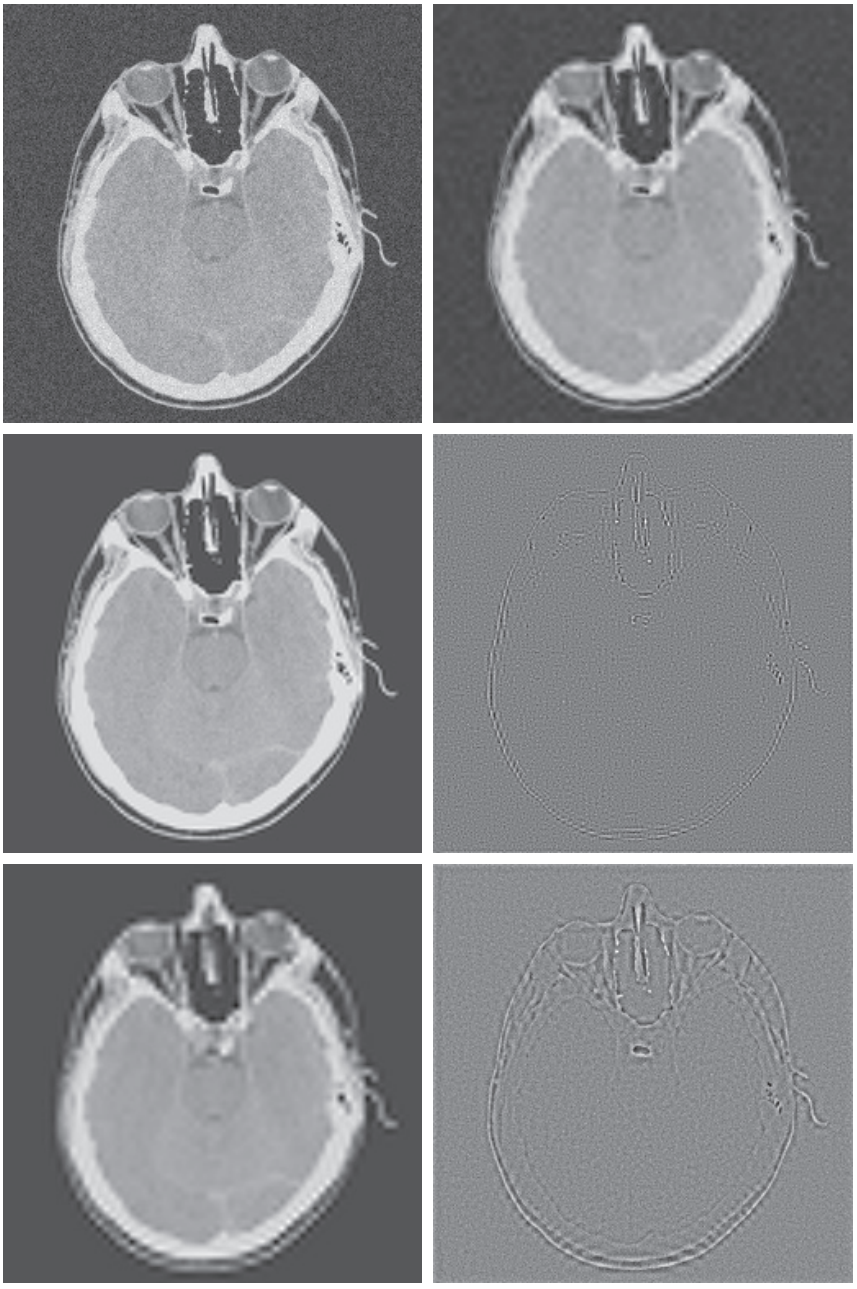
Пример 7.14. Устранение шума на основе вейвлет-анализа.

■ В качестве второго примера рассмотрим представленную на рис. 7.28(*а*) компьютерную томограмму головного мозга. Рассмотрение фоновой части изображения позволяет заключить, что изображение равномерно искажено не-

которым аддитивным белым шумом. Основанный на вейвлет-анализе общий метод устранения шума (т. е. подавления шумовой составляющей) состоит в следующем.

1. Выбирается система вейвлетов (например система Хаара, система симлетов и т. п.), а также количество уровней (масштабов) P разложения. Затем вычисляется БВП изображения с шумом.
2. Коэффициенты деталей подвергаются пороговому преобразованию. Это означает выбор некоторого значения порога и применение порогового преобразования к коэффициентам деталей в масштабах от $J - 1$ до $J - P$. При этом можно использовать как *жесткое пороговое преобразование*, так и *мягкое пороговое преобразование*. При жестком пороговом преобразовании приравниваются к нулю те значения элементов, которые меньше порога по абсолютной величине. При мягком пороговом преобразовании первоначально приравниваются к нулю значения элементов, меньшие порога по абсолютной величине, а затем к оставшимся ненулевым элементам применяется градационное преобразование, сдвигающее диапазон их значений к нулю. Мягкое пороговое преобразование позволяет избежать разрыва значений вблизи порога, которое характерно для жесткого преобразования. (См. раздел 10, где приведено обсуждение вопроса порогового преобразования.)
3. Осуществляется обратное вейвлет-преобразование (т. е. выполняется вейвлет-реконструкция), используя исходные коэффициенты приближения уровня $J - P$ и измененные коэффициенты деталей уровней от $J - P$ до $J - 1$.

На рис. 7.28(*б*) представлен результат применения описанного метода с использованием системы симлетов четвертого порядка, двух масштабов (т. е. $P = 2$) и глобального порога, который был определен интерактивно. Отметим уменьшение шума и размывание контуров на изображении. Эта потеря качества изображения контурных деталей значительно меньше на рис. 7.28(*в*), который получен простым обнулением коэффициентов деталей только в разложении наиболее крупного масштаба (но не пороговым преобразованием коэффициентов деталей в разложении мелкого масштаба) и последующим восстановлением. На этом изображении фоновый шум устранен практически полностью, а контуры (границы объектов) лишь слегка размыты. Разностное изображение на рис. 7.28(*г*) отображает ту информацию, которая оказалась потеряна в процессе устранения шума. Это изображение было получено с помощью обратного двухмасштабного БВП, которое применялось к разложению, все коэффициенты которого были обнулены, за исключением коэффициентов деталей самого крупного масштаба. Как нетрудно видеть, результирующее изображение содержит бóльшую часть шума исходного изображения и некоторую контурную информацию. Рис. 7.28(*д*) и (*е*) демонстрируют негативный результат, происходящий при стирании всех коэффициентов деталей. А именно изображение на рис. 7.28(*д*) представляет собой результат ДВП-восстановления, в котором были обнулены значения всех коэффициентов деталей на обоих уровнях. Изображение на рис. 7.28(*д*) является результатом восстановления, в котором были оставлены только коэффициенты деталей (т. е. коэффициенты приближения мелкого масштаба были удалены). Рис. 7.28(*е*) отображает информацию, кото-



а *б*
в *г*
д *е*

Рис. 7.28. Устранения шума путем изменения коэффициентов ДВП: (*а*) зашумленная компьютерная томограмма головного мозга; (*б*), (*в*) и (*д*) изображения, восстановленные после различных пороговых преобразований коэффициентов деталей; (*г*) и (*е*) информация, удаленная в процессе восстановления изображений (*в*) и (*д*). (Исходное изображение предоставлено медицинским центром университета Вандербилта)

рая оказалась потеряна. Обратите внимание на существенное увеличение количества контурной информации на рис. 7.28(*е*) и соответствующее понижение качества восстановления контуров на рис 7.28(*д*). ■

Поскольку при формировании изображения на рис. 28(*г*) были сохранены лишь коэффициенты деталей высокого разрешения, обратное преобразование иллюстрирует их вклад в изображение. Аналогично изображение на рис. 28(*е*) показывает вклад всех коэффициентов деталей.

7.6. Вейвлет–пакеты

Быстрое вейвлет-преобразование дает разложение функции в сумму масштабирующей функции и вейвлет-функций, частотные диапазоны которых находятся в логарифмическом соотношении. Таким образом, низкочастотная составляющая функции оказывается представленной масштабирующей функцией и вейвлет-функциями с узкими диапазонами, а высокочастотная составляющая — функциями с более широкими диапазонами. Это становится очевидно, если посмотреть на частотно-временную плоскость на рис. 7.23(*в*). Каждая горизонтальная полоса элемента покрытия постоянной высоты, содержащей базисные функций БВП одного масштаба, удваивается по высоте при продвижении вверх по частотной оси. Желание иметь больший контроль над разбиением частотно-временной плоскости (например получить меньшие диапазоны в области высоких частот), приводит к обобщению БВП и созданию более гибкой конструкции, называемой *вейвлет-пакетами* [Coifman, Wickerhauser, 1992]. Ценой такого обобщения является возрастание вычислительной сложности с $O(M)$ для БВП до $O(M \log_2 M)$ для вейвлет-пакетов.

Рассмотрим снова двухмасштабный банк фильтров на рис. 7.18(*а*), но разложение изобразим в виде *двоичного дерева*. На рис. 7.29(*а*) показана структура дерева и привязка соответствующих масштабных коэффициентов и вейвлет-коэффициентов БВП (см. рис. 7.18(*а*)) к *вершинам* этого дерева. *Корень* дерева отвечает коэффициентам приближения наиболее крупного масштаба, которые представляют собой отсчеты исходной функции, а *листья* дерева отвечают коэффициентам приближения и деталей на выходе преобразования. Единственная промежуточная вершина $W_\varphi(J-1, n)$, которая представляет собой приближение после фильтрации первого уровня, в обязательном порядке снова подвергается фильтрации второго уровня и превращается в два листа на выходе. Заметим, что коэффициенты в каждой вершине суть коэффициенты того линейного разложения, которое порождает ограниченные по частоте «части» корня $f(n)$. Поскольку любая такая часть есть элемент известного масштабирующего подпространства или подпространства вейвлетов (см. разделы 7.2.2 и 7.2.3), мы можем сменить обозначения для коэффициентов преобразования на рис. 7.29(*а*) на обозначения для соответствующих подпространств. В результате мы получим представленное на рис. 7.29(*б*) *дерево подпространств анализа*. Несмотря на то, что переменная W используется для обозначения как коэффициентов, так и подпространств, объекты различимы по формату нижних индексов.

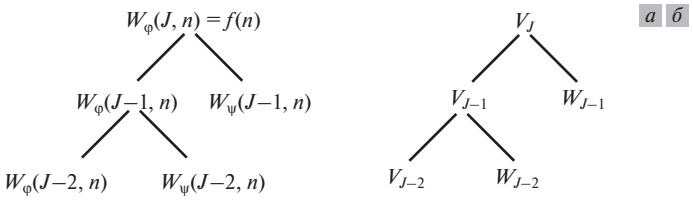


Рис. 7.29. Дерево (*а*) коэффициентов и (*б*) анализа для двухмасштабного БВП-банка анализа на рис. 7.18

Представленные концепции проиллюстрированы далее на рис. 7.30, где изображены трехмасштабный банк анализа, дерево анализа и соответствующий спектр частот. В отличие от рис. 7.18(*а*), на блок-схеме рис. 7.30(*а*) использованы обозначения, похожие на обозначения для дерева анализа на рис. 7.30(*б*) и для спектра на рис. 7.30(*в*). Так, в результате применения верхнего левого фильтра и последующего прореживания получаются величины $W_\psi(J-1, n)$, отмеченные символом W_{J-1}, который также используется для обозначения соответствующего подпространства вейвлетов. Это подпространство соответствует верхнему правому листу сопутствующего дерева анализа, а также самому правому (самому широкому) частотному диапазону соответствующего спектра.

Деревья анализа являются компактным и информативным способом представления кратномасштабных вейвлет-преобразований. Их легко рисовать, они занимают меньше места, чем соответствующие им блок-схемы, и позволяют относительно легко находить эффективные разложения. Трехмасштабное дерево анализа на рис. 7.30(*б*) делает возможным, например, следующие три варианта разложения:

$$V_J = V_{J-1} \oplus W_{J-1}; \tag{7.6-1}$$

$$V_J = V_{J-2} \oplus W_{J-2} \oplus W_{J-1}; \tag{7.6-2}$$

$$V_J = V_{J-3} \oplus W_{J-3} \oplus W_{J-2} \oplus W_{J-1}. \tag{7.6-3}$$

Эти разложения соответствуют одно-, двух- и трехмасштабному БВП-разложениям раздела 7.4 и могут быть получены из выражения (7.2-27) раздела 7.2.3 подстановкой в него $j_0 = J - P$ при $P = \{1, 2, 3\}$. В общем случае P-масштабное дерево БВП-анализа содержит P различных разложений.

Деревья анализа являются также эффективным средством представления *вейвлет-пакетов*, которые суть не что иное, как обычные вейвлет-преобразования с повторной фильтрацией деталей. Так, трехмасштабное дерево анализа для БВП на рис. 7.30(*б*) превращается в трехмасштабное дерево анализа для вейвлет-пакета на рис. 7.31. Обратим внимание на появление дополнительных нижних индексов. Когда некоторая вершина помечена парой нижних индексов, то первый нижний индекс в паре определяет масштаб БВП вершины-родителя, от которой происходит данная вершина. Второй индекс в паре, который представляет собой строку из символов A и D переменной длины, определяет путь от вершины-родителя до данной вершины. При этом символ A означает использование низкочастотного фильтра (фильтра приближения), а символ D — высокочастотного фильтра (фильтра деталей). Например,

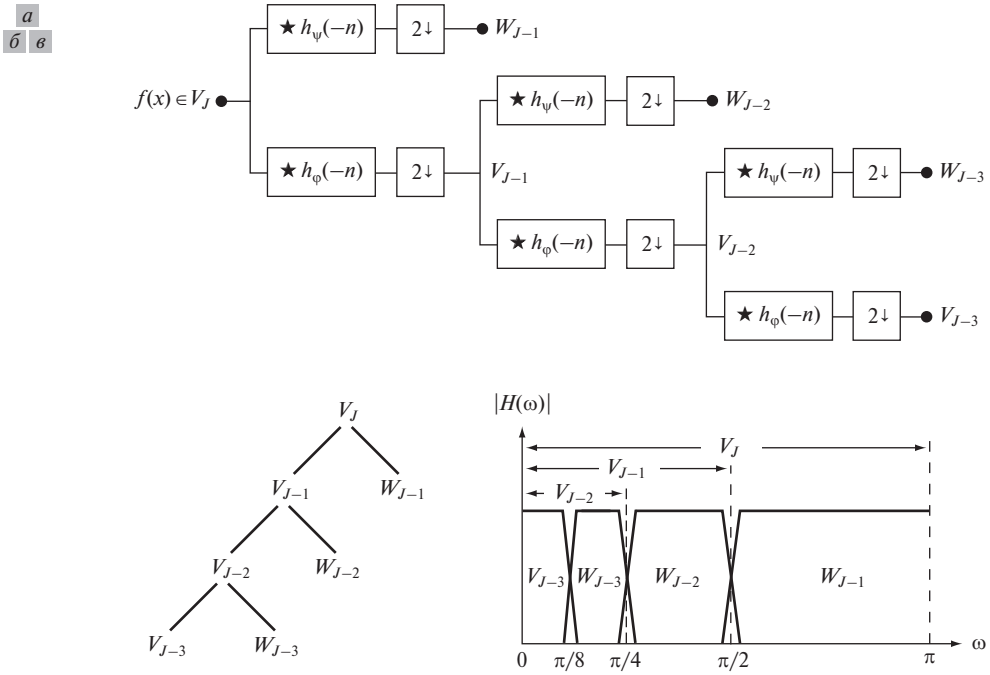


Рис. 7.30. Трехмасштабный БВП-банк фильтров: (*а*) блок-схема; (*б*) дерево пространств разложения (анализа); (*в*) спектральные характеристики разложения

подпространство $W_{J-1,DA}$ получено «пропусканием» БВП коэффициентов масштаба $J-1$ (т. е. родителя W_{J-1} на рис. 7.31) через дополнительный фильтр деталей (что дает $W_{J-1,D}$) и после этого через дополнительный фильтр приближения (что дает окончательно $W_{J-1,DA}$). На рис. 7.32(*а*) и (*б*) представлены банк фильтров и спектральные характеристики разложения, отвечающие дереву анализа на рис. 7.31. Обратим внимание, что «естественно упорядоченные» выходы банка фильтров на рис. 7.32(*а*) переупорядочены, основываясь на частотной составляющей на рис. 7.32(*б*) (см. задачу 7.25 касательно «частотно упорядоченных» вейвлетов).

Использование вейвлет-пакета, соответствующего трехмасштабному дереву на рис. 7.31, дает разложение, число частей (и связанных с ними частотно-

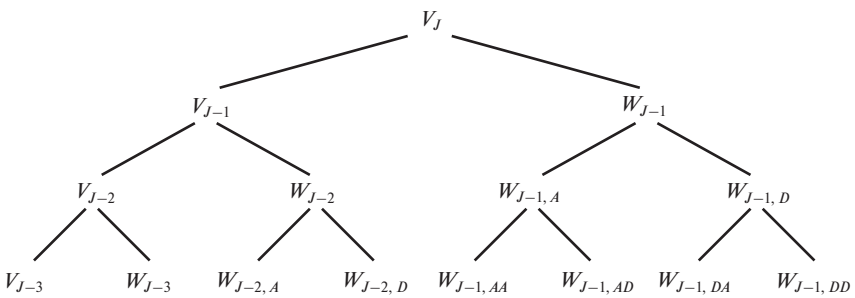


Рис. 7.31. Полное дерево анализа для трехмасштабного вейвлет-пакета

временных ячеек) которого почти втрое превосходит число частей разложения, получаемого в результате обычного вейвлет-преобразования, соответствующего трехмасштабному БВП-дереву. Напомним, что при обычном БВП мы разлагаем, фильтруем и прореживаем только низкочастотные составляющие. При этом возникает определенная логарифмическая зависимость (по основанию 2) между частотными диапазонами масштабирующих пространств и пространств вейвлетов, использующихся в представлении функции (см. рис. 7.30(*в*)). Поэтому в то время, как трехмасштабное дерево анализа БВП предполагает наличие трех возможных разложений, заданных уравнениями (7.6-1)—(7.6-3), дерево анализа вейвлет-пакета на рис. 7.31 приводит к 26 различным разложениям. Например, пространство V_J (а следовательно, и функция $f(n)$) может быть выражено в виде

$$V_J = V_{J-3} \oplus W_{J-3} \oplus W_{J-2,A} \oplus W_{J-1,D} \oplus W_{J-1,AA} \oplus W_{J-1,AD} \oplus W_{J-1,DA} \oplus W_{J-1,DD}, \qquad (7.6\text{-}4)$$

разделение спектра для которого показано на рис. 7.32(*б*); или в виде

$$V_J = V_{J-1} \oplus W_{J-1,D} \oplus W_{J-1,AA} \oplus W_{J-1,AD}. \qquad (7.6\text{-}5)$$

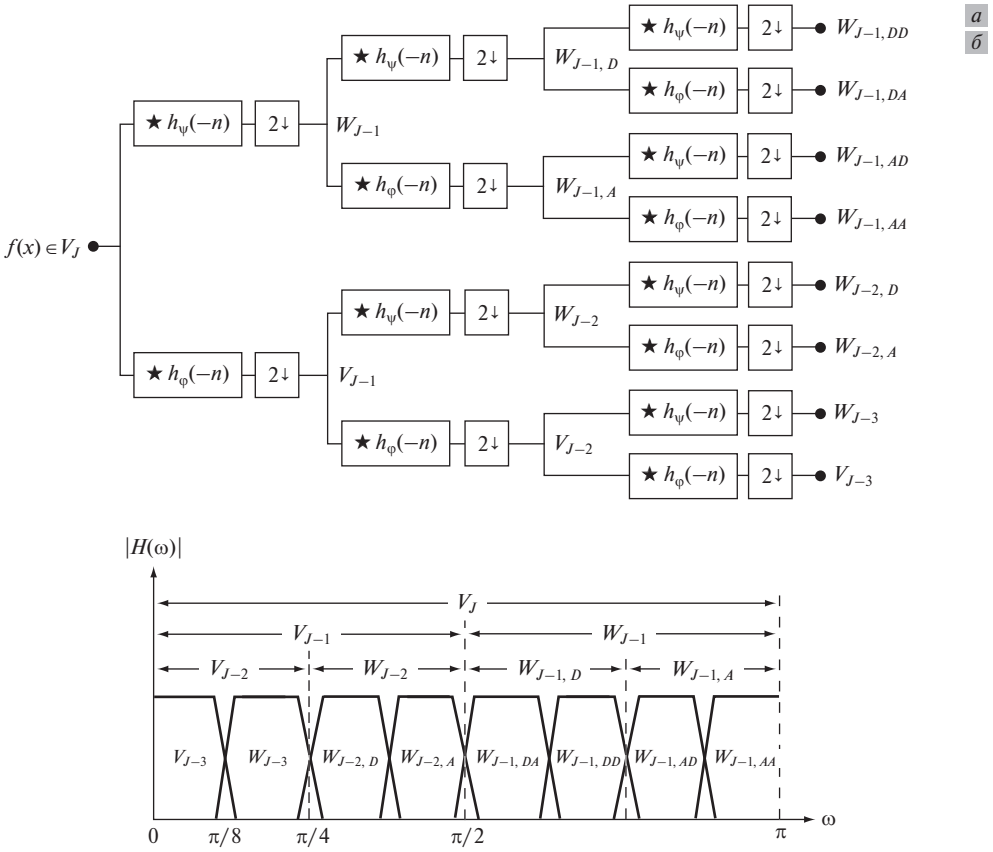


Рис. 7.32. (*а*) Банк фильтров и (*б*) спектральные характеристики разложения вейвлет-пакета, отвечающего полному трехмасштабному дереву анализа

Напомним, что знак ⊕ означает объединение пространств (по аналогии с объединением множеств). 26 разложений, ассоциируемых с рис. 7.31, определяются различными комбинациями вершин (пространств), которые могли бы быть скомбинированы для представления вершины (пространства) в корне дерева. Уравнения (7.6-4) и (7.6-5) определяют два из них.

Соответствующее разделение спектра показано на рис. 7.33. Обратим внимание на различия между этим последним спектром, спектром разложения с использованием полного вейвлет-пакета на рис. 7.32(*б*) и спектром трехмасштабного БВП-разложения на рис. 7.30(*в*). В общем случае P-масштабные преобразования на основе вейвлет-пакетов (и отвечающее им дерево анализа, состоящее из $P + 1$ уровня) дают возможность получить различные разложения в количестве

$$D(P+1)=\left[D(P)\right]^2+1, \tag{7.6-6}$$

где $D(1) = 1$. При таком большом числе допустимых разложений преобразования, основанные на применении пакетов, позволяют лучше контролировать процесс разделения спектра подлежащей разложению функции на части. Платой за это является увеличение вычислительной сложности (сравните банки фильтров на рис. 7.30(*а*) и 7.32(*а*)).

Рассмотрим теперь двумерный четырехполосный банк фильтров на рис. 7.24(*а*). Как указывалось в разделе 7.5, он делит приближение $W_\varphi(j+1,m,n)$ на четыре части: $W_\varphi(j,m,n)$, $W_\psi^H(j,m,n)$, $W_\psi^V(j,m,n)$ и $W_\psi^D(j,m,n)$. Как и в одномерном случае, повторение процесса приводит к формированию P-масштабного преобразования с масштабами $j = J-1$, $J-2$,..., $J-P$, причем $W_\varphi(J,m,n) = f(m,n)$. Разделение спектра после первой итерации (т. е. при $j + 1 = J$ в блок-схеме на рис. 7.24(*а*)) показано на рис. 7.34(*а*). Отметим, что частотная плоскость оказывается разделенной на четыре равные по площади составные части. Низкочастотная четверть диапазона в центре соответствует коэффициентам преобразования $W_\varphi(J-1,m,n)$ и масштабирующему пространству V_{J-1}. Это полностью согласуется с одномерным случаем. В двумерном случае, однако, мы имеем три (вместо одного) подпространства вейвлетов. Они обозначаются как W_{J-1}^H, W_{J-1}^V, W_{J-1}^D и отвечают коэффициентам $W_\psi^H(J-1,m,n)$, $W_\psi^V(J-1,m,n)$ и $W_\psi^D(J-1,m,n)$. На рис. 7.34(*б*) представлено соответствующее одномасштабное *квадродерево анализа БВП*. Обратим внимание на использованные верхние ин-

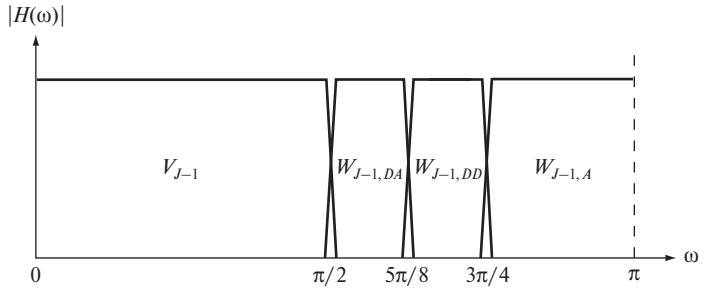


Рис. 7.33. Спектр разложения по формуле (7.6-5)

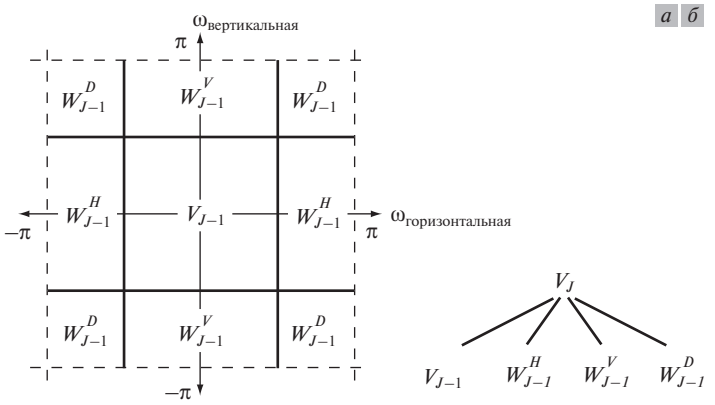


Рис. 7.34. Первое двумерное БВП-разложение: (*а*) спектр и (*б*) дерево подпространств анализа

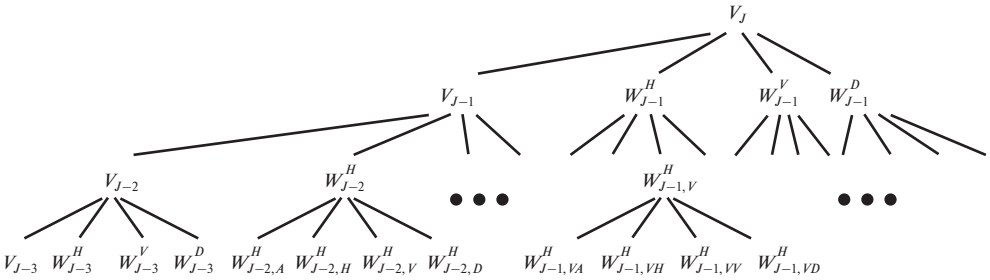


Рис. 7.35. Полное дерево анализа для трехмасштабного вейвлет-пакета. Показана лишь часть дерева

дексы, которые связывают обозначения для подпространств вейвлетов и обозначения для отвечающих им коэффициентов преобразования.

На рис. 7.35 представлена часть полного трехмасштабного двумерного дерева анализа для вейвлет-пакетов. Как и в одномерном случае на рис. 7.31, первый из нижних индексов для каждой вершины определяет масштаб обычного БВП вершины-родителя, от которой происходит данная вершина. Второй нижний индекс — строка из символов *A*, *H*, *V* и *D* переменной длины — определяет путь от *вершины-родителя* до рассматриваемой вершины. Например, вершина, отмеченная как $W^H_{J-1,VD}$, получена следующим образом. Сначала коэффициенты горизонтальных деталей двумерного БВП масштаба $J-1$ (т. е. вершины-родителя на рис. 7.35) подвергаются фильтрации с использованием банка фильтров вертикальных деталей, что дает $W^H_{J-1,V}$. Затем полученные коэффициенты подвергаются фильтрации с использованием банка фильтров диагональных деталей, что дает окончательно $W^H_{J-1,VD}$. Дерево анализа для двумерных *P*-масштабных вейвлет-пакетов дает возможность построить различные разложения в количестве

$$D(P+1)=\left[D(P)\right]^4+1, \qquad (7.6\text{-}7)$$

где $D(1) = 1$. Таким образом, полное число различных разложений, которые можно получить из трехмасштабного дерева на рис. 7.35, равно 83522. Проблема выбора среди них — предмет рассмотрения следующего примера.

Пример 7.15. Разложения с помощью двумерных вейвлет-пакетов.

■ Как отмечалось выше, одно дерево вейвлет-пакетов предоставляет многочисленные возможности для разложения. В действительности число возможных разложений часто бывает так велико, что не имеет смысла и даже невозможно перебирать или исследовать каждое из них в отдельности. Поэтому весьма желательно иметь эффективный алгоритм нахождения разложений, оптимальных по некоторым критериям, связанным с конкретными приложениями. Как будет видно, во многих случаях применимы классические критерии, основанные на вычислении некоторой функции стоимости (например энтропии или энергии), которые хорошо приспособлены для алгоритмов поиска по двоичным и квадродеревьям.

Рассмотрим задачу сокращения количества данных, требуемых для представления изображения отпечатка пальца размерами 400×480 элементов (задача сжатия), представленного на рис. 7.36(*а*). Задача сжатия изображений подробно рассматривается в главе 8. В этом примере в качестве отправной точки для процесса сжатия хотелось бы выбрать «наилучший» трехмасштабный вейвлет-пакет разложения. Использование деревьев анализа для трехмасштабных вейвлет-пакетов дает 83522 (см. (7.6-7)) потенциальных разложений. На рис. 7.36(*б*) представлено одно из таких разложений — разложение с использованием полного вейвлет-пакета, дерево которого, подобно дереву на рис. 7.35, содержит 64 листа. При этом листья дерева соответствуют полученным в результате разложения частям изображения, набор 8×8 которых составляет изображение рис. 7.36(*б*). Однако вероятность того, что это конкретное разложение, состоящее из 64 частей, является в некотором роде оптимальным для целей сжатия, относительно невелика. В отсутствие подходящего критерия мы не можем ни подтвердить, ни опровергнуть оптимальность выбранного разложения.

64 листа на рис. 7.35 соответствуют массиву 8×8 из 64 субизображений на рис. 7.36(*б*). Несмотря на внешний вид, они не являются квадратными. Данное искажение (заметное, например, в субизображении приближения) вызвано особенностями программы, формирующей данный результат.

Один из подходящих критериев выбора разложения для сжатия изображения на рис. 7.36(*а*) может быть получен при рассмотрении функции стоимости (функционала)

$$E(f) = \sum_{m,n} |f(m,n)|. \qquad (7.6\text{-}8)$$

Этот функционал может использоваться в качестве меры энергии двумерной функции f. Согласно данной мере, если $f(m, n) = 0$ для всех m и n, то энергия такой функции равна 0. Большое значение E, наоборот, указывает на то, что такая функция содержит много ненулевых значений. В большинстве схем трансформационного сжатия используется процедура усечения или порогового преобразования, при которой малые по величине коэффициенты заменяются нулевыми

а *б*

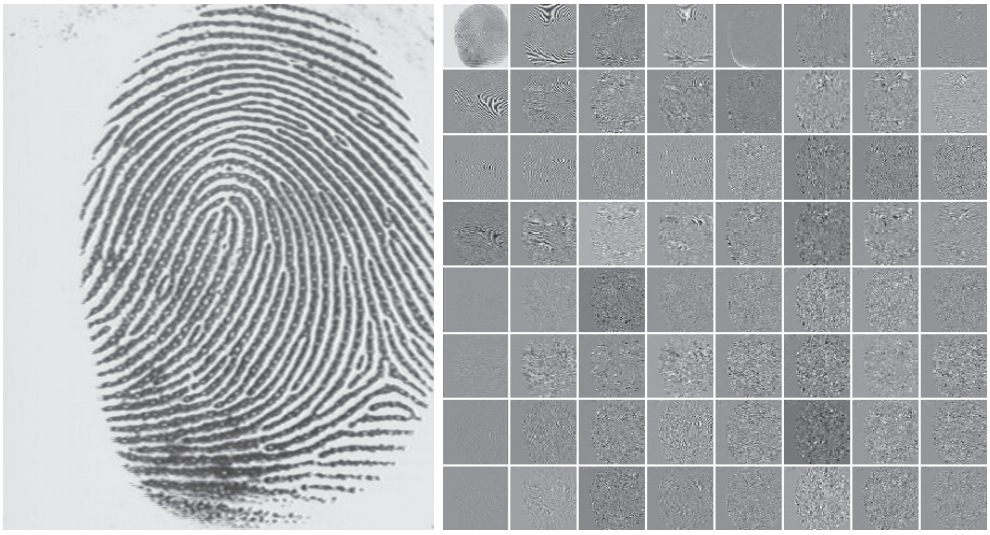


Рис. 7.36. (*а*) Изображение отпечатка пальца и (*б*) его трехмасштабное разложение с использованием полного вейвлет-пакета. (Исходное изображение предоставлено Национальным институтом стандартов и технологий)

значениями. Поэтому использование функции стоимости, минимизация которой максимизирует число значений, близких к нулевым, приводит к разумному критерию выбора «хорошего» разложения с точки зрения задачи сжатия.

> Другие возможные меры энергии включают сумму квадратов $f(m,n)$, сумму логарифмов квадратов и т.д. В задаче 7.27 определяется одна возможная функция, основанная на энтропии.

Только что рассмотренную функцию стоимости просто вычислять и легко использовать в стандартных процедурах оптимизации по деревьям. Алгоритм оптимизации должен использовать значение этой функции для минимизации «цены» листьев дерева анализа. Предпочтение должно быть отдано тому дереву, значение энергии листьев которого минимально, поскольку соответствующее разложение имеет максимальное число близких к нулю коэффициентов, что приводит к большей степени сжатия. Функция стоимости (7.6-8) локальна в том смысле, что она может быть представлена в виде суммы своих значений, вычисленных отдельно для частей (вершин), составляющих изображение. При этом вычисление значения функции стоимости для каждой вершины требует только той информации, которая содержится в этой вершине. Поэтому нетрудно предложить следующий эффективный алгоритм для нахождения решения с минимальной энергией.

Чтобы построить оптимальное дерево анализа, для каждой из вершин, начиная от корня и двигаясь вниз уровень за уровнем до достижения листьев, необходимо выполнить следующие действия.

Шаг 1. Подсчитать значение энергии E_P для данной вершины (рассматривая ее как вершину-родитель), а также значения энергии для ее четырех вершин-потомков E_A, E_H, E_V и E_D. Для разложений, основанных на использовании двумерных вейвлет-пакетов, вершине-родителю отвечает двумерный массив коэффициентов приближения или деталей, а вершинам-потомкам — полученные в результате фильтрации этого массива коэффициенты приближения и коэффициенты горизонтальных, вертикальных и диагональных деталей.

Шаг 2. Если суммарное значение энергии для вершин-потомков меньше значения энергии для вершины-родителя, т.е. $E_A + E_H + E_V + E_D < E_P$, то присоединить потомков к строящемуся дереву анализа. Если суммарное значение энергии для вершин-потомков больше или равно значению функционала для вершины-родителя, то отсечь потомков и оставить только родителя; данная вершина будет являться листом оптимального дерева анализа.

Представленный алгоритм можно использовать для (1) усечения деревьев вейвлет-пакетов или (2) конструирования процедур вычисления оптимальных деревьев с самого начала. В последнем случае несущественные потомки, исходящие из тех вершин, которые удаляются на втором шаге алгоритма, вычисляться не будут. На рис. 7.37 представлено оптимальное разложение, полученное в результате применения рассмотренного алгоритма с функционалом (7.6-8) к изображению на рис. 7.36(*а*). Соответствующее дерево анализа представлено на рис. 7.38. Отметим, что многие части представленного на рис. 7.36(*б*) полного разложения на 64 составляющие объединены, а соответствующие ветви представленного на рис. 7.35 полного дерева анализа с 64 листьями отсечены. Кроме того, значения яркости тех частей изображения на рис. 7.37, которые не были подвергнуты процедуре дальнейшего разложения, являются почти постоянными и лежат в середине яркостного диапазона. Все части изображения на этом

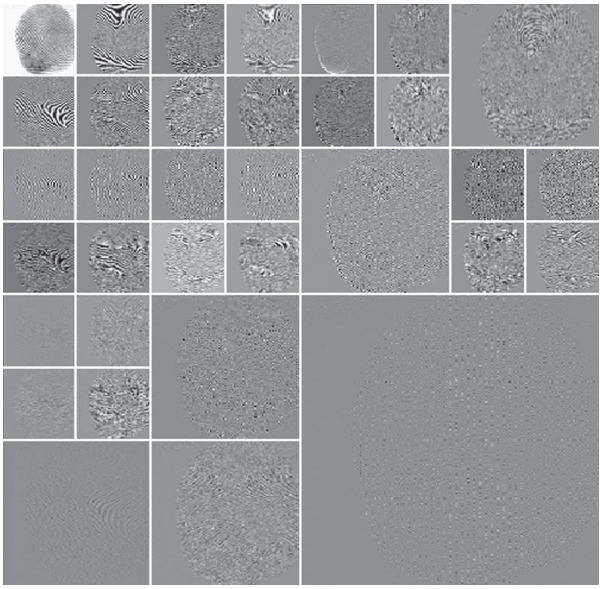


Рис. 7.37. Оптимальное разложение изображения отпечатка пальца на рис. 7.36(*а*) с помощью вейвлет-пакета

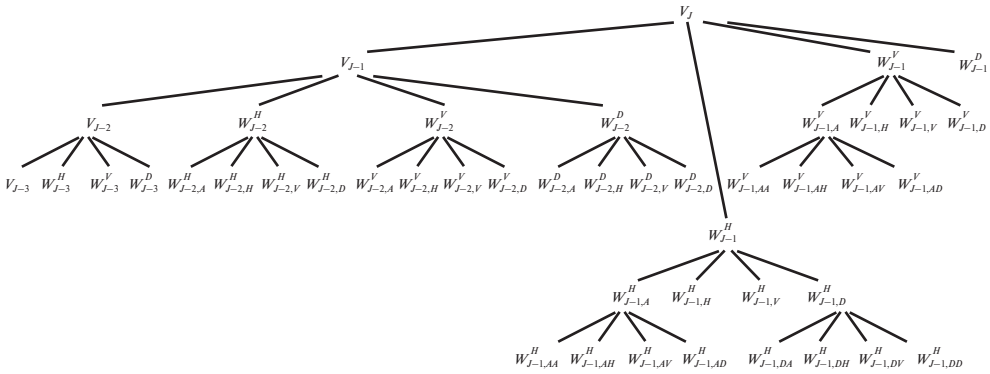


Рис. 7.38. Оптимальное дерево анализа, соответствующее разложению с помощью вейвлет-пакета, который показан на рис. 7.37

рисунке (за исключением приближения) были подвергнуты градационному преобразованию так, чтобы яркость 128 соответствовала нулевому значению коэффициентов. Как видно из рисунка, эти части изображения содержат малое количество энергии, а значит, их дальнейшее разложение не приведет к уменьшению общего значения энергии. ■

Представленный пример связан с практической задачей, которую удалось решить с применением вейвлетов. В настоящее время Федеральное бюро расследований (ФБР), обладающее огромной базой данных отпечатков пальцев, установило вейвлет-ориентированный национальный стандарт оцифровки и сжатия дактилоскопических изображений [FBI, 1993]. Использование биортогональных вейвлетов позволило достичь в этом стандарте типичного коэффициента сжатия 1:15. Преимущества вейвлет-ориентированного сжатия изображений по сравнению с более традиционными JPEG-подходами изучаются в следующей главе.

Фильтры разложения, использованные в примере 7.15 (и входящие в стандарт ФБР), являются частью широко известного семейства вейвлетов, на-

Таблица 7.4. Коэффициенты биортогонального вейвлета Коэна — Добеши — Фово [Cohen, Daubechies, Feauveau, 1992]

n	$h_0(n)$	$h_1(n)$	n	$h_0(n)$	$h_1(n)$
0	0	0	9	0,8259	0,4178
1	0,0019	0	10	0,4208	0,0404
2	−0,0019	0	11	−0,0941	−0,0787
3	−0,0170	0,0144	12	−0,0773	−0,0145
4	0,0119	−0,0145	13	0,0497	0,0144
5	0,0497	−0,0787	14	0,0119	0
6	−0,0773	0,0404	15	−0,0170	0
7	−0,0941	0,4178	16	0,0119	0
8	0,4208	−0,7589	17	0,0010	0

а б
в г
д е
ж з

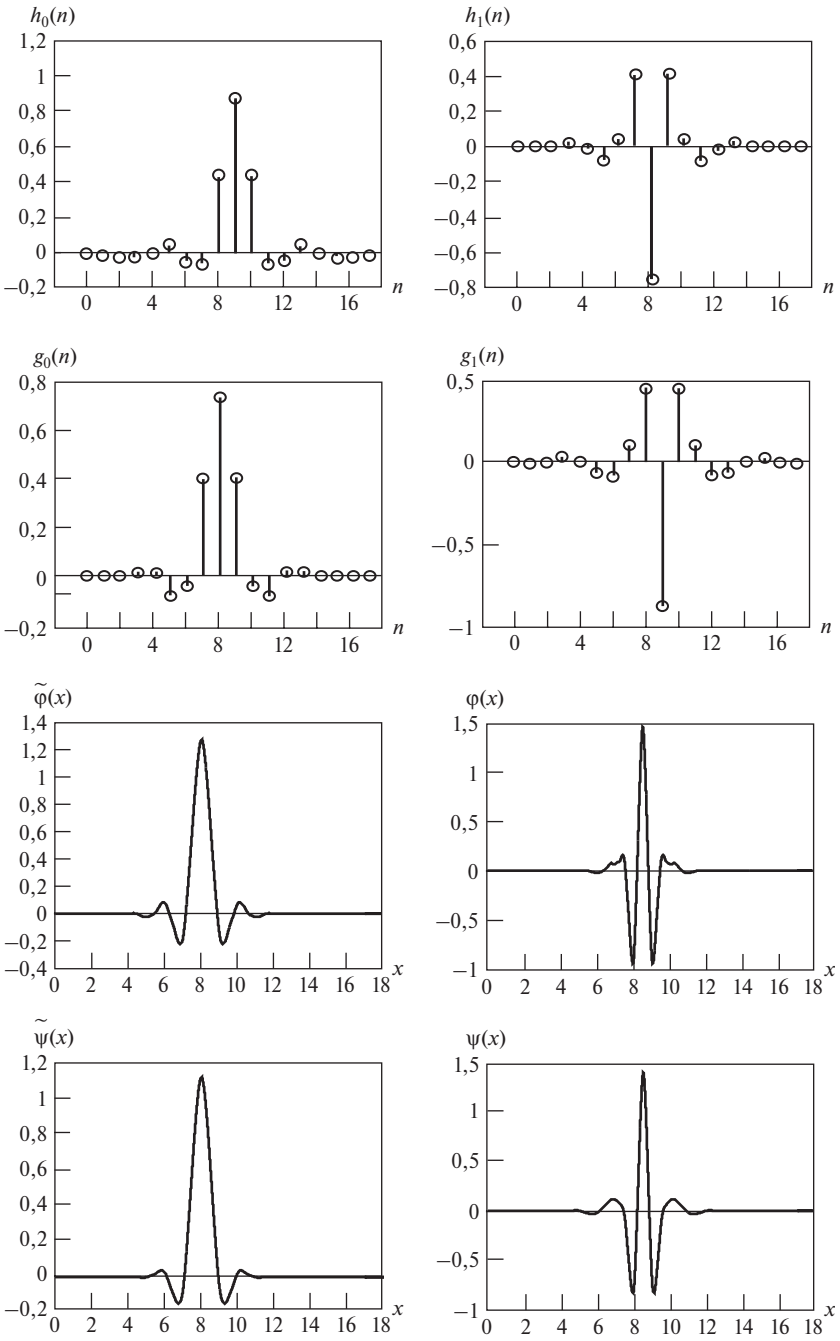


Рис. 7.39. Представитель биортогонального семейства вейвлетов Коэна — Добеши — Фово: (*а*) и (*б*) коэффициенты фильтров анализа; (*в*) и (*г*) коэффициенты фильтров синтеза; (*д*)—(*з*) двойственные пары масштабирующих функций и вейвлет-функций. См. значения $h_0(n)$ и $h_1(n)$ при $0 \le n \le 17$ в табл. 7.3

зываемого биортогональными вейвлетами Коэна — Добеши — Фово [Cohen, Daubechies, Feauveau, 1992]. Масштабирующая функция и вейвлет-функция этого семейства симметричны и имеют одинаковые размеры. Благодаря этим свойствам вейвлеты семейства Коэна — Добеши — Фово входят в число наиболее широко используемых биортогональных вейвлетов. На рис. 7.39(*д*)—(*з*) представлены двойственные пары масштабирующих функций и вейвлет-функций. На рис. 7.39(*а*)—(*г*) показаны соответствующие фильтры анализа и синтеза. Значения коэффициентов низкочастотного и высокочастотного фильтров анализа $h_0(n)$ и $h_1(n)$ при $0 \le n \le 17$ приведены в табл. 7.4. Коэффициенты соответствующих биортогональных фильтров синтеза могут быть вычислены по формуле (7.1-11), используя $g_0(n) = (-1)^{n+1} h_1(n)$ и $g_1(n) = (-1)^n h_0(n)$, т. е. фильтры синтеза являются перекрестно-модулированными копиями фильтров анализа. Заметим, что для того чтобы фильтры имели одинаковую длину, были добавлены нулевые коэффициенты, а также что табл. 7.4 и рис. 7.39 задают их для системы субполосного кодирования и декодирования на рис. 7.6(*а*); что касается БВП, $h_\varphi(-n) = h_0(n)$ и $h_\psi(-n) = h_1(n)$.

Заключение

Материал настоящей главы закладывает прочные математические основы для понимания той роли, которую играют вейвлет-методы и кратномасштабный анализ в обработке изображений, и открывает возможности для овладения этими методами. Вейвлеты и вейвлет-преобразования являются сравнительно новыми средствами обработки изображений, причем круг задач, для решения которых они применяются, стремительно расширяется. В силу известного сходства с преобразованием Фурье многие методы главы 4 находят свой аналог среди методов вейвлет-анализа. Неполный список приложений, которые допускают подход с позиций вейвлет-анализа, включает задачи сопоставления, совмещения, сегментации, подавления шумов, восстановления, улучшения, сжатия, морфологической фильтрации и вычислительной томографии. Поскольку не представляется возможным охватить все эти приложения в рамках одной главы, темы для обсуждения были выбраны исходя из их значимости для понимания основных идей и подготовки читателя к дальнейшему изучению данной области. В главе 8 будет рассмотрено применение вейвлетов в задаче сжатия изображений.

Ссылки и литература для дальнейшего изучения

Существует много хороших работ, посвященных вейвлетам и их приложениям. Некоторые из них были использованы нами при написании основных разделов настоящей главы. При изложении материала раздела 7.1.2, посвященного субполосному кодированию и цифровой фильтрации, мы следовали книге [Vetterli, Kovacevic, 1995], а при рассмотрении в разделах 7.2 и 7.4 кратномасштабного разложения и быстрого вейвлет-преобразования придерживались изложения этих

вопросов в [Burrus, Gopinath, Guo, 1998]. Ссылки на работы, ставшие основой для остального материала главы, приведены в тексте. Все примеры данной главы были вычислены с использованием пакета Matlab (см. [Gonzalez et al., 2004]).

История вейвлет-анализа описана в книге [Habbard, 1998]. Ранние прототипы вейвлетов появились одновременно в различных областях и собраны воедино в работе [Mallat, 1987]. Эта работа заложила математические основы теории вейвлетов. Многое из истории вейвлетов можно почерпнуть из работ Мейера [Meyer, 1987, 1990, 1992a, 1992б, 1993], Малла [Mallat, 1987, 1989a—в, 1998] и Добеши [Daubechies, 1988, 1990, 1992, 1993, 1996]. Большой интерес к вейвлетам был в немалой степени стимулирован работами последней. Книга Добеши [Daubechies, 1992], посвященная математической теории вейвлетов, является классической.

По поводу применения вейвлетов в обработке изображений мы отсылаем читателя к посвященным этой тематике работам общего характера, таким как [Castleman, 1996], и многочисленным специальным работам, некоторые из которых представляют собой доклады на конференциях. Среди работ, относящихся к этой последней категории, упомянем [Rosenfeld, 1984], [Prasad, Iyengar, 1997] и [Topiwala, 1998]. Следующие недавние статьи могут служить отправными точками дальнейших исследований в применении вейвлетов в различных прикладных задачах обработки изображений: обнаружение углов — [Gao et al., 2007]; поведение пространственных решеток — [Olkkonen, Olkkonen, 2007]; комплексные вейвлеты — [Selesnick et al., 2005] и [Kokare et al., 2005]; совмещение изображений — [Thévenaz, Unser, 2000]; классификация на основе текстур — [Chang, Kuo, 1995] и [Unser, 1995]; применение вейвлетов для морфологического анализа — [Heijmans, Goutsias, 2000]; восстановление изображений — [Banham et. al., 1994], [Wang, Zhang, Pan, 1995] и [Banham, Kastaggelos, 1996]; улучшение изображений — [Xu et. al., 1994] и [Chang, Yu, Vetterli, 2000]; вычислительная томография — [Delaney, Bresler, 1995] и [Westenberg, Roerdink, 2000]; описание изображений и задача совпадения — [Lee, Sun, Chen, 1995], [Liang, Kuo, 1999], [Wang, Lee, Toraichi, 1999] и [You, Bhattacharya, 2000]. По поводу наиболее важного применения вейвлетов для сжатия изображений см., например, [Brechet et al., 2007], [Demin Wang et al., 2006], [Antonini et. al., 1992], [Wei et. al., 1998], а также книгу [Topiwala, 1998]. Наконец, имеется ряд специальных выпусков журналов, посвященных вейвлетам. Упомянем специальный выпуск [*IEEE Transactions on Information Theory*, 1992], посвященный вейвлет-преобразованиям и кратномасштабному анализу сигналов, специальный выпуск [*IEEE Transactions on Signal Processing*, 1993], посвященный вейвлетам и обработке сигналов, а также специальный выпуск [*IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1989], где отдельный раздел посвящен кратномасштабным представлениям.

Хотя материал настоящей главы концентрируется на основополагающих вопросах вейвлет-анализа и его применения в области обработки изображений, значительный интерес представляют вопросы конструирования самих систем вейвлетов. Заинтересованного читателя мы отсылаем к работам [Battle, 1987, 1988], [Daubechies, 1988, 1992], [Cohen, Daubechies, 1992], [Meyer, 1990], [Mallat, 1989б], [Unser, Aldroubi, Eden, 1993] и [Gröchenig, Madych, 1992]. Данный список не является исчерпывающим, но может быть использован в качестве отправного пункта для дальнейшего чтения. Имеет смысл обратить внимание также

на список литературы по субполосному кодированию и банкам фильтров, имеющейся в [Strang, Nguyen, 1996] и [Vetterli, Kovacevic, 1995], а также на ссылки, приведенные в тексте главы в связи с рассматриваемыми примерами.

Задачи

7.1 Придумайте систему декодирования пирамиды разностей с предсказанием, которая формируется системой кодирования на рис. 7.2(*б*), и нарисуйте ее блок-схему. Предполагайте, что система кодирования не вносит ошибок квантования.

★**7.2** Постройте полностью заполненную пирамиду приближений и соответствующую пирамиду разностей с предсказанием для изображения

$$f(x,y)=\begin{bmatrix}11 & 12 & 13 & 14\\ 15 & 16 & 17 & 18\\ 19 & 20 & 21 & 22\\ 23 & 24 & 25 & 26\end{bmatrix}.$$

В процедуре фильтрации на рис. 7.2(*б*) используйте усреднение по окрестности 2×2 и предположите, что интерполяция осуществляется дублированием пикселей.

★**7.3** Пусть задано изображение размерами $2^J \times 2^J$. Меньшее или большее количество данных потребуется для представления пирамиды этого изображения, состоящей из $J+1$ уровня? Чему равен коэффициент сжатия или растяжения?

7.4 Банк фильтров двухполосного субполосного кодирования содержит следующие фильтры: $h_0(n)=\{-1/\sqrt{2},-1/\sqrt{2}\}$, $h_1(n)=\{-1/\sqrt{2},1/\sqrt{2}\}$, $g_0(n)=\{1/\sqrt{2},-1/\sqrt{2}\}$ и $g_1(n)=\{1/\sqrt{2},-1/\sqrt{2}\}$. Является ли он ортонормированным, биортогональным или тем и другим одновременно?

7.5 Для заданной последовательности $f(n)$ = {0,1; 0,25; 0,5; 1} при n = 0, 1, 2, 3 вычислите:

(а) Знако-инвертированную последовательность,

(б) Последовательно-инвертированную последовательность,

(в) Модулированную последовательность,

(г) Модулированную и затем последовательно-инвертированную последовательность,

(д) Последовательно-инвертированную и затем модулированную последовательность.

(е) Соответствуют ли результаты (г) и (д) уравнению (7.1-9)?

7.6 Вычислите коэффициенты фильтров синтеза Добеши $g_0(n)$ и $g_1(n)$ в примере 7.2. Используя (7.1-22) при m = 0, убедитесь, что эти фильтры являются ортонормированными. Являются ли эти фильтры ортогональными при m = 1?

★**7.7** Нарисуйте блок-схему двумерного четырехканального банка фильтров субполосного декодирования для восстановления входного изображения $x(m,n)$ в блок-схеме на рис. 7.7.

7.8 Выпишите матрицу преобразования Хаара для случая N = 8.

7.9 **(а)** Вычислите преобразование Хаара для следующего изображения размерами 2×2:

$$\mathbf{F} = \begin{bmatrix} 4 & -2 \\ 7 & 3 \end{bmatrix}.$$

(б) Обратное преобразование Хаара имеет вид $\mathbf{F} = \mathbf{H}^{-1}\mathbf{T}(\mathbf{H}^{-1})^T$, где $\mathbf{T}$ — преобразование Хаара от $\mathbf{F}$, $\mathbf{H}^{-1}$ — обратная матрица преобразования Хаара, а верхний индекс T — операция транспонирования. Покажите, что обратная матрица $\mathbf{H}_2^{-1} = \mathbf{H}_2^T$ и используйте ее для вычисления обратного преобразования Хаара от результата пункта (а).

7.10 Вычислите коэффициенты разложения двумерного вектора $[1, 3]^T$ относительно следующих базисов (фреймов) и выпишите соответствующие разложения.

★ **(а)** Базис из двух векторов в $\mathbb{R}^2$ $\varphi_0 = [1/\sqrt{2}, 1/\sqrt{2}]^T$ и $\varphi_1 = [1/\sqrt{2}, -1/\sqrt{2}]^T$.

(б) Базис в $\mathbb{R}^2$ $\varphi_0 = [1,0]^T$, $\varphi_1 = [1,1]^T$ и двойственный к нему базис $\tilde{\varphi}_0 = [1,-1]^T$, $\tilde{\varphi}_1 = [0,1]^T$.

(в) Фрейм из трех векторов в $\mathbb{R}^2$ $\varphi_0 = [1,0]^T$, $\varphi_1 = [-1/2, \sqrt{3}/2]^T$, $\varphi_2 = [-1/2, -\sqrt{3}/2]^T$ и двойственный к нему фрейм $\tilde{\varphi}_i = 2\varphi_i / 3$, $i = \{0, 1, 2\}$.

Указание: используйте скалярное произведение евклидового пространства.

7.11 Покажите, что масштабирующая функция

$$\varphi(x) = \begin{cases} 1 & 0{,}5 \le x < 0{,}75, \\ 0 & \text{в остальных случаях} \end{cases}$$

не удовлетворяет второму условию кратномасштабного анализа.

7.12 Выразите масштабирующее пространство V_2 через масштабирующую функцию $\varphi(x)$. Используя определение (7.2-14) для масштабирующей функции Хаара, выразите масштабирующие функции Хаара пространства V_2 для сдвигов $k = \{0, 1, 2, 3\}$.

★**7.13** Нарисуйте график вейвлета $\psi_{2,2}(x)$ системы вейвлетов Хаара. Выразите вейвлет $\psi_{2,2}(x)$ через масштабирующую функцию Хаара.

7.14 Предположим, что функция $f(x)$ принадлежит масштабирующему пространству V_2 системы Хаара, т. е. $f(x) \in V_2$. Используя (7.2-22), выразите пространство V_2 в виде прямой суммы масштабирующего пространства V_0 и необходимых пространств вейвлетов. Предполагая, что функция $f(x)$ обращается в нуль вне полуинтервала [0,1), укажите масштабирующие функции и вейвлеты, входящие в разложение $f(x)$, основанное на полученном Вами выражении.

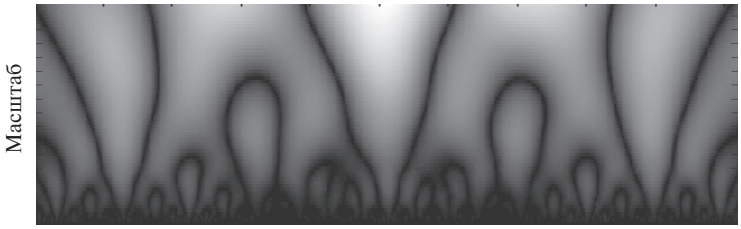
7.15 Вычислите первые четыре члена разложения функции из примера 7.7 в вейвлет-ряд с начальным масштабом $j_0 = 2$. Выпишите полученное разложение, используя соответствующие масштабирующие функции и вейвлеты. Сравните Ваш результат с результатом примера 7.7, в котором был выбран начальный масштаб $j_0 = 0$.

7.16 Заданное формулами (7.3-5) и (7.3-6) ДВП зависит от выбора начального масштаба j_0.

(а) Вычислите одномерное ДВП функции $f(n) = \{1, 3, 0, -4\}$, $0 \le n \le 3$, из примера 7.8 с начальным масштабом $j_0 = 1$ (вместо $j_0 = 0$).

(б) Используйте результат (а) для вычисления значения $f(2)$ по значениям преобразования.

★**7.17** Какие заключения относительно одномерной функции позволяет сделать ее интегральное вейвлет-преобразование, представленное ниже на рисунке?



7.18 **(а)**Интегральное (непрерывное) вейвлет-преобразование в задаче 7.17 было сформировано на компьютере. Функция, к которой применялось преобразование, была предварительно дискретизована, т. е. преобразование применялось к отсчетам функции с некоторым шагом. Что же непрерывного в этом преобразовании, т. е. что отличает его от дискретного вейвлет-преобразования той же функции?

★ **(б)** При каких условиях использование ДВП предпочтительнее использования ИВП? Существуют ли случаи, когда использование ИВП предпочтительнее использования ДВП?

★**7.19** Нарисуйте схему банка фильтров БВП, необходимого для вычисления преобразования в задаче 7.16. Напишите последовательности на входах и выходах всех элементов схемы.

7.20 Вычислительная сложность быстрого вейвлет-преобразования для M-точечного массива равна $O(M)$, т. е. число необходимых операций пропорционально M. Чем определяется коэффициент пропорциональности?

7.21 ★ **(а)** Если на вход трехмасштабного банка фильтров на рис. 7.30(*а*) подать масштабирующую функцию Хаара $\varphi(n) = 1$ при $n = 0, 1,..., 7$ и равную 0 в остальных случаях, то каков будет результат преобразования при использовании системы вейвлетов Хаара?

(б) Каков будет результат преобразования, если подать на вход вейвлет-функцию Хаара $\psi(n) = \{1, 1, 1, 1, -1, -1, -1, -1\}$ при $n = 0, 1,..., 7$?

(в) Какой последовательности на входе отвечает преобразование $\{0, 0, 0, B, 0, 0, 0, 0\}$ с ненулевым коэффициентом $W_{\psi}(1,1) = B$?

★**7.22** Двумерное быстрое вейвлет-преобразование имеет много общего с рассмотренной в разделе 7.2.1 пирамидой изображений. Чем они похожи? Имея трехмасштабное вейвлет-преобразование, представленное на рис. 7.10(*а*), как бы Вы построили соответствующую пирамиду приближений? Сколько уровней имела бы эта пирамида?

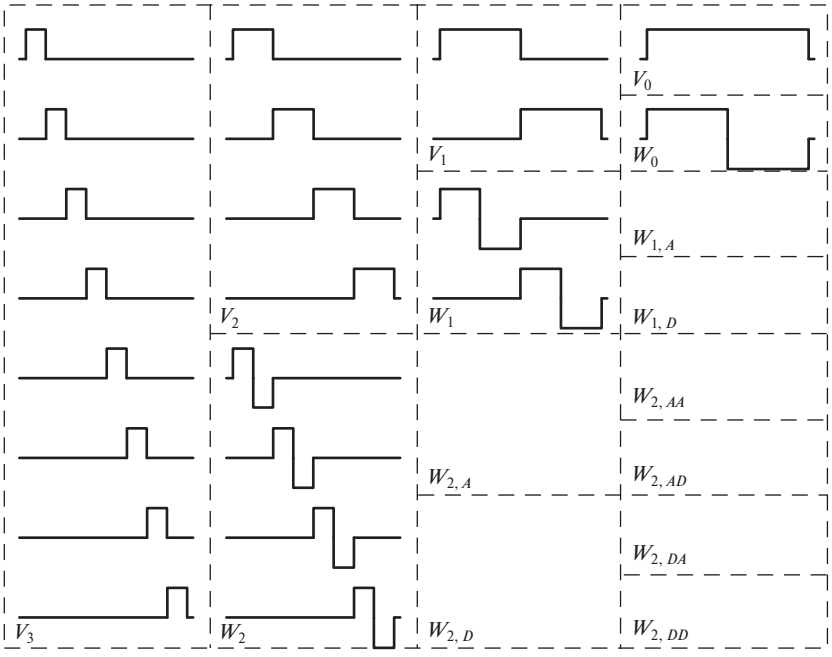
7.23 Вычислите двумерное вейвлет-преобразование по системе вейвлетов Хаара для изображения размерами 2×2 из задачи 7.9. Нарисуйте схему требуемого банка фильтров и напишите массивы на входах и выходах всех элементов схемы.

★**7.24** В случае Фурье-преобразования согласно таблице 4.3

$$f(x-x_0, y-y_0) \Leftrightarrow F(\mu, v)e^{-i2\pi(\mu x_0/M+vy_0/N)}$$

и сдвиг не меняет изображение спектра $|F(\mu,v)|$. Установите свойства вейвлет-преобразования по отношению к сдвигу, используя приведенную ниже последовательность изображений. Самое левое изображение состоит из двух белых квадратов в центре (размерами 16×16) на сером фоне (размерами 64×64). Второе слева изображение является одномасштабным вейвлет-преобразованием по системе вейвлетов Хаара. Третье изображение представляет собой вейвлет-преобразование изображения, полученного в результате сдвига исходного на 16 элементов вправо и вниз. Наконец четвертое (самое правое) изображение — вейвлет-преобразование изображения, полученного в результате сдвига исходного на 1 элемент вправо и вниз.

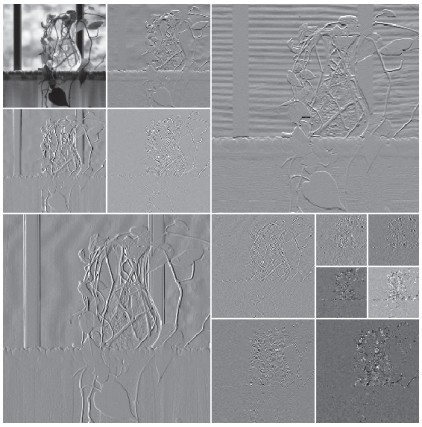
7.25 В приведенной ниже таблице нарисованы вейвлеты и масштабирующие функции Хаара для четырехмасштабного быстрого вейвлет-преобразования. Нарисуйте дополнительные базисные функции, необходимые для трехмасштабного разложения с использованием полного вейвлет-пакета. Приведите явные выражения для этих функций или выражения, с помощью которых их можно определить. Затем осуществите упорядочивание базисных функций в соответствии с их частотными составляющими и объясните результат.



7.26 Ниже представлено разложение изображения с вазой на рис. 7.1 с помощью вейвлет-пакета.

(а) Нарисуйте соответствующее дерево анализа этого разложения и подпишите его вершины, используя обозначения соответствующих масштабирующих пространств и пространств вейвлетов.

(б) Нарисуйте и подпишите частотные диапазоны разложения.



7.27 Используя вейвлеты Хаара, найдите для функции $f(n) = 0{,}5$, $n = 0, 1, 2,...,15$, пакетное разложение с минимальным количеством информации. Используйте в качестве функции стоимости ненормированную шенноновскую энтропию:

$$E[f(n)] = \sum_n f^2(n) \ln\left(f^2(n)\right).$$

Нарисуйте оптимальное дерево и напишите в вершинах найденные значения энтропии.

ГЛАВА 8

СЖАТИЕ ИЗОБРАЖЕНИЙ

Жизнь коротка, а объемы информации безграничны... Сокращение — неизбежное зло, и задача того, кто взялся кратко объяснить некое явление, — на самом высоком уровне сделать дело, дурное по своей натуре, но без которого было бы еще хуже.

Олдос Хаксли «Возвращение в дивный новый мир» (пер. Е. Сыромятникова и И. Головачева)

Введение

Сжатие изображений — наука и искусство сокращения объема данных, требуемого для представления изображения, — является в настоящее время одной из наиболее используемых и коммерчески успешных технологий в области цифровой обработки изображений. Число изображений, ежедневно подвергаемых сжатию и восстановлению, ошеломляюще велико, однако сами эти процессы фактически незаметны для пользователя. Каждый владелец цифровой камеры, пользователь Интернета или же зритель кинофильма с цифрового видеодиска (DVD) пользуется достижениями, полученными благодаря алгоритмам и стандартам, обсуждаемым в настоящей главе.

Чтобы лучше понять необходимые требования к компактному представлению изображений, рассмотрим объем данных, требуемый для представления двухчасового телефильма стандартного разрешения (SD), в котором каждый кадр имеет размеры 720×480×24 бита. Цифровое кино (или видео) есть последовательность видеокадров, в которой каждый кадр является полноцветным неподвижным изображением. Поскольку видеовоспроизводящее устройство должно отображать кадры последовательно со скоростью около 30 кадров в секунду, данные цифрового видео SD должны поступать со скоростью

$$30\frac{\text{кадров}}{\text{сек}}\times(720\times480)\frac{\text{точек}}{\text{кадр}}\times3\frac{\text{байтов}}{\text{точка}}=31104000\ \text{байт/сек},$$

а двухчасовой фильм состоит из

$$31104000\frac{\text{байтов}}{\text{сек}}\times\left(60^2\right)\frac{\text{сек}}{\text{час}}\times2\ \text{час}\cong2{,}24\cdot10^{11}\ \text{байт}$$

или 224 Гб (гигабайта) данных. Для их запоминания потребовалось бы 27 двухслойных DVD-дисков по 8,5 Гб (подразумевая обычные 12 см диски). Что-

бы записать двухчасовое кино на один диск, каждый кадр необходимо сжать (в среднем) в 26,3 раза. Для *телевидения высокого разрешения* (HDTV), в котором разрешение достигает 1920×1080×24 бит/кадр, сжатие должно быть еще выше.

Изображения, передающиеся в Интернете, и фотографии высокого разрешения в цифровых камерах также регулярно сжимаются для экономии пространства памяти и сокращения времени передачи. Так, например, домашние интернет-соединения обеспечивают скорость передачи данных от 56 Кбит/с (килобит в секунду) по обычным телефонным линиям до 12 Мбит/с (мегабит в секунду) по широкополосным выделенным линиям. Чтобы передать с этими скоростями небольшое полноцветное изображение размерами 128×128×24 бита, потребуется от 7 до 0,03 секунд. Сжатие может сократить время передачи в 2—10, а то и большее число раз. Аналогичным же образом может быть увеличено число снимков, которые можно записать на карту памяти цифрового фотоаппарата (8-мегапиксельная камера может записать на флеш-карту объемом 1 Гб всего 41 несжатый кадр объемом 24 мегабайта). Вдобавок к указанному сжатие изображений играет важную роль во многих других областях, включая телеконференции, дистанционный сбор данных, графическое представление документов, рентгенографию, факсимильную передачу данных (FAX). Все увеличивающееся число приложений во многом зависит от эффективности передачи, запоминания и манипулирования двоичными, полутоновыми и цветными изображениями.

В данной главе мы ознакомимся с теорией и практикой сжатия цифровых изображений. Мы рассмотрим наиболее часто используемые методы сжатия и опишем промышленные стандарты, в которых они применяются. Материал по сути является вводным и применим как к неподвижным изображениям, так и к видеопоследовательностям. Глава завершается знакомством с тематикой *цифровых водяных знаков* — процессом внедрения в изображение видимых или невидимых данных (например информации копирайта).

8.1. Основы

Термин *сжатие данных* означает уменьшение объема данных, используемого для представления определенного количества информации. В этом определении *данные* и *информация* являются совершенно различными понятиями; данные фактически являются тем средством, с помощью которого информация передается. Поскольку для представления одного и того же количества информации может быть использовано различное количество данных, то про представления, в которых имеется лишняя или повторяющаяся информация, говорят, что они содержат *избыточные данные*. Если через b и b' обозначить число битов (или любых других единиц информации) в двух представлениях одной и той же информации, то *относительная избыточность данных* R представления с b битами составит

$$R = 1 - \frac{1}{C}, \tag{8.1-1}$$

где значение C, обычно называемое *коэффициентом сжатия*, определяется как

$$C = \frac{b}{b'}. \tag{8.1-2}$$

Если, например, $C = 10$ (иногда пишут 10:1), то значит в представлении большего размера в среднем содержится 10 бит данных на 1 бит данных в представлении меньшего размера. Соответственно, относительная избыточность данных большего представления будет равна 0,9 ($R = 0{,}9$), что означает, что 90 % его данных избыточны.

В контексте сжатия изображений b в (8.1-2) обычно означает число битов, необходимое для представления изображения как двумерного массива значений яркости. Введенные в разделе 2.4.2 двумерные массивы яркости представляют собой удобные форматы для наблюдения и интерпретации человеком, а также стандарт, по которому оцениваются все другие представления. Однако когда идет речь о компактном представлении изображений, эти форматы оказываются далеки от оптимальности. Двумерные массивы яркостей характеризуются тремя основными типами избыточности данных, которые можно обнаружить и использовать:

1. *Кодовая избыточность*. *Код* — последовательность символов (знаков, букв, чисел, битов и тому подобное), применяемых для представления существа информации или набора событий. Каждой детали информации или события присваивается последовательность *кодовых символов*, называемая *кодовым словом*. Восьмибитовые коды, используемые для представления значений в большинстве двумерных массивах яркостей, как правило, содержат большее число битов, чем это необходимо.
2. *Пространственная* или *временна́я избыточность*. Поскольку значения пикселей в большинстве двумерных массивов яркостей пространственно коррелированы (т. е. значение каждого пикселя похоже на значения соседних пикселей или даже зависит от них), информация излишним образом дублируется в представлении коррелированных пикселей. В видеопоследовательностях информация дублируется также в коррелированных по времени пикселях (т. е. тех, значения которых похожи или зависят от значений пикселей в соседних кадрах).
3. *Лишняя информация*. Большинство двумерных массивов содержат информацию, игнорируемую зрительной системой человека и/или являю-

а *б* *в*

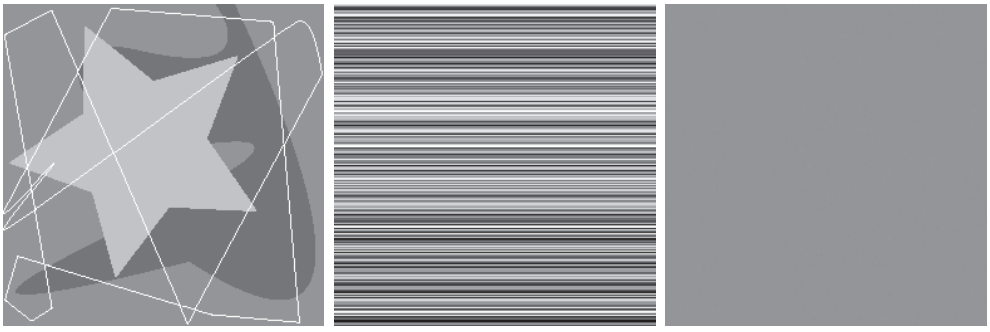


Рис. 8.1. Сгенерированные в компьютере изображения размерами 256×256×8 бит с (*а*) кодовой избыточностью, (*б*) пространственной избыточностью, (*в*) лишней информацией. (Каждое было создано для демонстрации одного основного вида избыточности, однако может также иллюстрировать и остальные)

щуюся посторонней в достижении цели, для которой используется изображение. Она является излишней в том смысле, что не используется.

Сгенерированные в компьютере изображения, показанные на рис. 8.1(*а*)—(*в*), демонстрируют каждую из указанных основных видов избыточности. Как будет показано в ближайших трех разделах, сжатие достигается, когда один или более видов избыточности уменьшается или устраняется.

8.1.1. Кодовая избыточность

В главе 3 был рассмотрен метод улучшения изображения с помощью обработки гистограммы в предположении, что значения яркости на изображении являются случайными величинами. В настоящем разделе мы используем похожий формализм, чтобы познакомиться с оптимальным кодированием информации.

Предположим, что дискретная случайная переменная r_k, распределенная на интервале [0, $L-1$], представляет значение яркости изображения размерами $M \times N$ и что каждое значение r_k появляется с вероятностью $p_r(r_k)$. Как и в разделе 3.3,

$$p_r(r_k) = \frac{n_k}{MN} \quad k = 0, 1, 2, ..., L-1, \tag{8.1-3}$$

где L — общее число уровней яркости, n_k — число пикселей изображения, имеющих значение яркости k. Если число битов, используемых для представления каждого из значений r_k, равно $l(r_k)$, то среднее число битов, требуемых для представления значения одного элемента, равно

$$L_{\text{avg}} = \sum_{k=0}^{L-1} l(r_k) p_r(r_k). \tag{8.1-4}$$

Итак, средняя длина всех кодовых слов, присвоенных различным значениям яркостей, определяется как сумма произведений числа битов, используемых для представления каждого из уровней яркостей, на вероятность появления этого уровня яркости. Таким образом, общее число битов, требуемое для кодирования изображения размерами $M \times N$, составит MNL_{avg}. Если уровни яркости изображения представляются *обычным* m-битовым двоичным кодом[1] фиксированной длины, правая часть уравнения (8.1-4) упрощается до m бит. Это легко показать. Если m подставляется вместо $l(r_k)$, константа m может быть вынесена за знак суммирования, оставив только сумму $p_r(r_k)$ по всем $0 \le k \le L-1$, которая равна 1. Таким образом, $L_{\text{avg}} = m$.

Пример 8.1. Простое объяснение неравномерного кодирования.

■ Сгенерированное компьютером изображение на рис. 8.1(*а*) имеет вероятности значений яркостей, выписанные во втором столбце табл. 8.1. Если для представления имеющихся 4 уровней используется обычный 8-битовый двоичный код (обозначенный как код 1 в табл. 8.1), то $L_{\text{avg}} = 8$ битам, поскольку $l(r_k) = 8$ битам для всех r_k. С другой стороны, если используется код 2 из табл. 8.1, то среднее

[1] В случае *обычного* (или *естественного*) двоичного кода каждому информационному элементу или событию (например значению яркости) присваивается одно из 2^m значений m-битовой двоичной последовательности.

число битов, необходимых для кодирования одного пикселя, в соответствии с выражением (8.1-4) составит:

$$L_{\text{avg}} = 0{,}25 \cdot 2 + 0{,}47 \cdot 2 + 0{,}25 \cdot 3 + 0{,}03 \cdot 3 = 1{,}81 \text{ бит}.$$

Общее число битов, требуемое для записи всего изображения, будет $MNL_{\text{avg}} = 256 \times 256 \times 1{,}81$ или 118621 бит. Из (8.1-2) и (8.1-1) результирующее сжатие и соответствующая относительная избыточность будут

$$C = \frac{256 \times 256 \times 8}{118621} = \frac{8}{1{,}81} \approx 4{,}42$$

и

$$R = 1 - \frac{1}{4{,}42} = 0{,}774$$

соответственно. Таким образом, 77,4 % данных в исходном 8-битовом двумерном массиве яркостей являются избыточными.

Сжатие, достигаемое при использовании кода 2, является результатом того, что более вероятным значениям отводится меньшее число битов, а менее вероятным — большее число битов. В получившемся неравномерном коде значению r_{128} — наиболее вероятному на изображении — присвоено однобитовое кодовое слово 1 (длины $l_2(r_{128}) = 1$), в то время как наименее вероятному значению r_{255} — трехбитовое слово 001 (длины $l_2(r_{255}) = 3$). Обратим внимание, что наилучшим равномерным кодом, который мог бы использоваться для кодирования значений изображения рис. 8.1(*a*), является обычная двухбитовая счетная последовательность {00, 01, 10, 11}, однако результирующее сжатие будет лишь 8/2 или 4:1 — приблизительно на 10 % меньше, чем 4,42:1 — сжатие, достигаемое неравномерным кодом. ■

Как видно из предыдущего примера, *кодовая избыточность* возникает, когда при выборе кодовых слов, присваиваемых событиям (например значениям яркости), знания о вероятностях событий не используются в полной мере. Если значения яркости изображения представляются обычным двоичным кодом, это происходит почти всегда. Причиной возникновения кодовой избыточности в этом случае является то, что изображения, как правило, состоят из объектов, имеющих регулярную, в некотором смысле предсказуемую морфологию (форму) и отражательные свойства поверхности, причем размеры объектов на изображении обычно намного превышают размеры пикселей. Прямым следствием этого является тот факт, что на большинстве изображений определенные зна-

Таблица 8.1. Пример неравномерного кодирования

r_k	$p_r(r_k)$	Код 1	$l_1(r_k)$	Код 2	$l_2(r_k)$
$r_{87} = 87$	0,25	01010111	8	01	2
$r_{128} = 128$	0,47	10000000	8	1	1
$r_{186} = 186$	0,25	11000100	8	000	3
$r_{255} = 255$	0,03	11111111	8	001	3
r_k для $k \neq 87, 128, 186, 255$	0	—	8	—	0

чения яркости оказываются более вероятными, чем другие (т. е. гистограммы большинства изображений не являются равномерными). Обычное двоичное кодирование значений яркости таких изображений присваивает кодовые слова одинаковой длины как более вероятным, так и менее вероятным значениям. В результате не обеспечивается минимизация уравнения (8.1-4) и появляется кодовая избыточность.

8.1.2. Пространственная и временна́я избыточность

Рассмотрим сгенерированный в компьютере набор строк постоянной яркости, составляющих изображение на рис. 8.1(*б*), и сделаем некоторые наблюдения:

1. Все значения яркостей равновероятны. Как видно на рис. 8.2, гистограмма изображения равномерная.
2. Поскольку яркость каждой строки выбиралась случайно, значения ее пикселей в вертикальном направлении независимы друг от друга[2].
3. Поскольку значения пикселей вдоль каждой строки равны, они максимально коррелированы (полностью зависят друг от друга) в горизонтальном направлении.

Первое наблюдение говорит, что изображение на рис. 8.1(*б*), будучи представленным в виде обычного 8-битового массива значений, не может быть сжато при помощи одного только неравномерного кода. В отличие от изображения на рис. 8.1(*а*) (и в примере 8.1), гистограмма которого *не* была равномерной, в данном случае 8-битовый равномерный код минимизирует выражение (8.1-4). Наблюдения 2 и 3 показывают значительную пространственную избыточность, которая может быть уменьшена, например, записью изображения на рис. 8.1(*б*) в виде последовательности *пар длин серий*, в которой каждая пара длин серий задает начало серии новой яркости и число последовательных пикселей, имеющих новое значение. Представление в виде пар длин серий позволяет сжать рассматриваемый двумерный 8-битовый массив с коэффициентом сжатия (256×256×8)/((256+256)×8), или 128:1. Каждая строка из 256 пикселей исходного

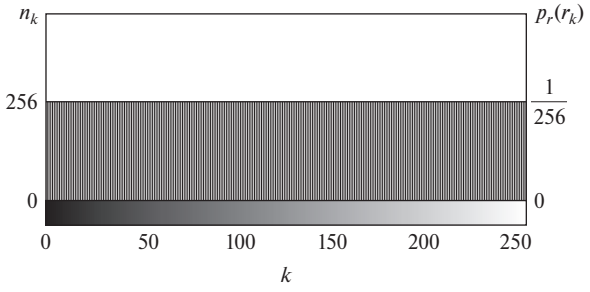


Рис. 8.2. Гистограмма яркостей элементов изображения на рис. 8.1(*б*)

[2] Вообще говоря, для данного изображения это не так. Легко видеть, что значения строк на нем не повторяются, а следовательно, они и не могут быть совершенно независимыми. Однако данное замечание не является принципиальным для последующих рассуждений. — *Прим. перев.*

изображения в представлении длин серий заменяется на одно 8-битовое значение яркости и одно значение длины[3].

В большинстве случаев пиксели изображения являются коррелированными пространственно (по направлениям как *x*, так и *y*), а также по времени, если изображение — часть видеопоследовательности. Поскольку значения большинства пикселей могут быть достаточно хорошо предсказаны по соседним значениям, то, значит, количество информации, которое несет отдельный пиксель, сравнительно невелико; подавляющая часть его визуального вклада является избыточной в том смысле, что она может быть определена исходя из значений соседей. Чтобы уменьшить избыточность, связанную с пространственной и временно́й корреляциями пикселей, двумерный массив яркостей должен быть трансформирован в более эффективную, но, как правило, «не визуальную» форму. Например, могут быть использованы длины серий или же разности между соседними пикселями. Преобразования такого типа называют *отображениями*. Отображение называют *обратимым*, если значения пикселей исходного двумерного массива яркостей могут быть безошибочно восстановлены из преобразованного набора данных; в противном случае отображение называют *необратимым*.

8.1.3. Лишняя информация

Один из простейших способов сжатия набора данных состоит в удалении из него (пропуске) лишней информации. В контексте сжатия изображений очевидными кандидатами на пропуск становятся виды информации, игнорируемой зрительной системой человека или же являющейся посторонней по отношению к намеченному использованию изображения. Так, сгенерированное изображение на рис. 8.1(*в*), поскольку оно выглядит как просто однородное серое поле, может быть представлено единственной величиной (средней) яркости — одним 8-битовым значением. Исходный массив яркостей размерами 256×256×8 сокращается до одного байта[4], и коэффициент сжатия составляет (256×256×8)/8, или 65536:1. Конечно, первоначальные 256×256×8 битов изображения должны быть еще восстановлены для наблюдения и/или анализа, но снижение качества исходного изображения будет небольшим или даже незаметным.

На рис. 8.3(*а*) представлена гистограмма изображения рис. 8.1(*в*). Заметим, что в действительности на нем присутствует несколько значений яркости — от 125 до 131. Зрительная система человека усредняет имеющиеся яркости, воспринимая лишь среднее значение и игнорируя небольшие изменения, которые на самом деле имеются. На изображении рис. 8.3(*б*), полученном после эквализации гистограммы, изменения яркости стали хорошо видны, *а также* обнару-

---

[3] Заметим, что в данном случае значение длины 256 также можно уместить в один байт, поскольку длины 0 при кодировании длин серий (как правило) не бывает. — *Прим. перев.*

[4] В этом утверждении кроется некоторое лукавство. Либо на приемном конце должны заранее знать, что передается значение массива, являющегося константой, либо же необходимо передать дополнительную информацию, говорящую об этом (например число одинаковых элементов), что потребует дополнительных битов передачи. — *Прим. перев.*

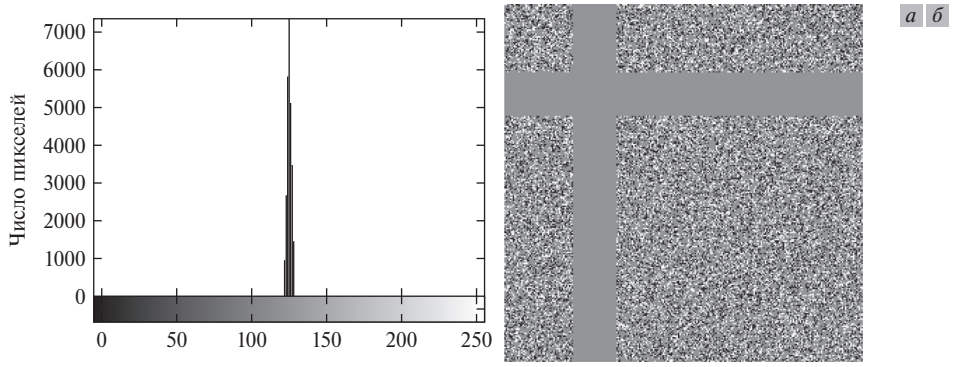


Рис. 8.3. (*а*) Гистограмма изображения на рис. 8.1(*в*) и (*б*) результат эквализации гистограммы данного изображения

жились две ранее неразличимые полоски постоянной яркости — одна вертикальная, а другая горизонтальная. Если изображение на рис. 8.1(*в*) представить лишь одним значением средней яркости, эта «невидимая» структура, т. е. константные области и случайные вариации яркости вне этих областей, оказывается утерянными. Должна или нет сохраняться подобная информация, зависит от области использования данных. Если информация важна, как это бывает в медицинских приложениях (например при архивировании цифровых рентгенограмм), она не должна теряться. В иных случаях информация, возможно, является избыточной и может быть исключена ради достижения более высокой степени сжатия.

В завершение данного раздела необходимо отметить, что обсуждаемая здесь избыточность фундаментально отличается от избыточности, рассматриваемой в разделах 8.1.1 и 8.1.2. Ее устранение допустимо лишь постольку, поскольку данная информация является не существенной для нормального зрительного процесса и/или запланированного использования изображения. Т. к. такой пропуск приводит к потере количественной информации, данный процесс в большинстве случаев называют *квантованием*. Такая терминология не противоречит обычному использованию данного слова, которое обычно означает отображение широкого диапазона входных значений в ограниченный набор выходных значений (см. раздел 2.4). Поскольку при квантовании информация теряется, то данный процесс является необратимой операцией.

8.1.4. Измерение содержащейся в изображении информации

В предыдущих разделах было представлено несколько путей уменьшения объема данных, требуемых для представления изображения. Естественно, возникает следующий вопрос: насколько много бит в действительности необходимо для представления информации на изображении? Другими словами, существует ли минимальное количество данных, которых достаточно для полного описания изображения без потери информации? Математическую основу для ответа на этот и аналогичные вопросы дает *теория информации*. Ее фундаментальная предпосылка заключается в том, что источник информации может быть описан как вероятностный процесс, который может быть измерен естественным обра-

зом. В соответствии с этим предположением говорят, что случайное событие E, имеющее вероятность $P(E)$, содержит

$$I(E) = \log \frac{1}{P(E)} = -\log P(E) \tag{8.1-5}$$

единиц информации. Если $P(E) = 1$ (то есть событие возникает всегда), то $I(E) = 0$ и данному событию не приписывают никакой информации. Это означает, что если нет никакой неопределенности, связанной с событием, то сообщение о том, что данное событие произошло, не содержит никакой информации (что имеет место *всегда*, если $P(E) = 1$).

Для ознакомления с теорией информации и теорией вероятностей обратитесь к интернет-сайту книги.

Основание логарифма в (8.3-5) задает единицу измерения количества информации. Если используется основание m, то говорят о единицах измерения по основанию m (m-ичная система счисления). Когда основание равно 2, единица информации называется *бит*. Заметим, что если $P(E) = ½$, то $I(E) = -\log_2 ½$, или один бит. Таким образом, 1 бит есть количество информации, передаваемое сообщением о том, что произошло одно из двух возможных равновероятных событий. Простой пример сообщения такого рода — сообщение о результате подбрасывания монеты.

Рассмотрим источник статистически независимых случайных событий из дискретного набора случайных значений $\{a_1, a_2, ..., a_J\}$ с ассоциированными вероятностями $\{P(a_1), P(a_2), ..., P(a_J)\}$. Среднее количество информации, приходящейся на один символ источника, называемое *энтропией* (или *неопределенностью источника*) и обозначаемое H, равно[5]

$$H = -\sum_{j=1}^{J} P(a_j) \log P(a_j). \tag{8.1-6}$$

Элементы a_j в данной формуле называют *символами источника* (или *буквами*), а набор исходных символов $\{a_1, a_2, ..., a_J\}$ — *алфавитом источника*. Если символы источника статистически независимы, его называют *источником без памяти*.

Если рассматривать изображение как выход воображаемого «источника яркостей» без памяти, то можно использовать гистограмму наблюдаемого изображения для оценки вероятностей символов источника. Тогда энтропия источника яркостей будет

$$\tilde{H} = -\sum_{k=0}^{L-1} p_r(r_k) \log_2 p_r(r_k), \tag{8.1-7}$$

где переменные L, r_k и $p_r(r_k)$ являются такими, как были определены в разделах 8.1.1 и 3.3. Поскольку использован логарифм по основанию 2, то энтропия (8.1-7) дает значение средней информации на один символ источника яркостей в битах. Кодировать *значения яркостей* такого воображаемого источника (а значит, и образца изображения) со средней кодовой длиной, меньшей, чем $\tilde{H}$ бит/пиксель, невозможно.

[5] Формулу (8.1-6) также называют *формулой Шеннона. — Прим. перев.*

Выражение (8.1-6) описывает источники без памяти с J символами источника; выражение (8.1-7) использует оценки вероятностей для L значений яркостей изображения.

Пример 8.2. Оценки энтропии изображения.

■ Оценить энтропию изображения на рис. 8.1(*а*) можно подстановкой вероятностей значений яркости из табл. 8.1 в (8.1-7):

$$\tilde{H} = -[0{,}25\log_2 0{,}25 + 0{,}47\log_2 0{,}47 + 0{,}25\log_2 0{,}25 + 0{,}03\log_2 0{,}03] \approx$$
$$\approx -[0{,}25(-2) + 0{,}47(-1{,}09) + 0{,}25(-2) + 0{,}03(-5{,}06)] \approx 1{,}6614 \text{ бит/пиксель.}$$

Аналогичным образом можно показать, что энтропии изображений на рис. 8.1(*б*) и (*в*) равны 8 бит/пиксель и 1,566 бит/пиксель соответственно. Обратим внимание, что изображение рис. 8.1(*а*) выглядит как содержащее наибольшее количество визуальной информации, но в то же время имеет почти минимальную энтропию — 1,66 бит/пиксель. У изображения (*б*) энтропия почти в 5 раз выше, чем у (*а*), но кажется, что оно имеет такую же (или даже меньшую) визуальную информацию; а изображение (*в*), которое будто бы совсем не содержит либо содержат очень мало информации, имеет энтропию почти такую же, как и изображение (*а*). Очевидный вывод из этого состоит в том, что значение энтропии, а следовательно, и количество информации в изображении далеко от интуитивной оценки. ■

Первая теорема Шеннона

Напомним, что неравномерный код в примере 8.1 способен отобразить значения яркости изображения на рис. 8.1(*а*), используя лишь 1,81 бит/пиксель. Хотя это и превышает оценку энтропии в примере 8.2, которая равна 1,6614 бит/пиксель, *первая терема Шеннона*, называемая *теоремой кодирования канала без шума*[6] [Shannon, 1948], утверждает, что для представления данного изображения достаточно всего лишь 1,6614 бит/пиксель. Чтобы доказать ее для общего случая, Шеннон рассматривал использование одного кодового слова для группы из n последовательных символов источника (а не одного кодового слова для одного символа источника) и показал, что

$$\lim_{n\to\infty}\left[\frac{L_{\text{avg},n}}{n}\right] = H, \tag{8.1-8}$$

где $L_{\text{avg},n}$ — среднее число кодовых символов, требуемых для представления всех возможных групп из n символов. При доказательстве он ввел определение n-кратного расширения источника без памяти как гипотетического источника, генерирующего блоки по n символов[7] из одиночных символов первоначального

[6] Также ее называют основной *теоремой о кодировании при отсутствии помех. — Прим. перев.*

[7] Выходом n-кратного расширения источника является группа из n последовательных символов первоначального источника одиночных символов. Данная группа рассматривается как *блоковая случайная переменная*, вероятность которой равна произведению вероятностей входящих в нее отдельных символов. Энтропия n-кратного расширения источника $H_n = n \cdot H$, где H — энтропия первоначального источника одиночных символов.

источника, и вычислял значение $L_{\text{avg},n}$, применяя выражение (8.1-4) к кодовым словам, используемым для представления n-символьных блоков. Уравнение (8.1-8) говорит, что отношение $L_{\text{avg},n}/n$ может быть сделано произвольно близким к H путем кодирования выхода бесконечнократного расширения источника одиночных символов. Т. е. возможно представление выхода источника без памяти кодом, имеющим в среднем H единиц информации на один символ источника.

Если теперь вернуться к идее, что изображение есть «экземпляр» выхода источника, его породившего, то блок из n символов источника соответствует группе из n соседних пикселей. Чтобы построить неравномерный код для n-пиксельных блоков, необходимо вычислить относительные частоты появления таких блоков. Но n-кратное расширение гипотетического источника яркостей с 256 возможными значениями имеет 256^n возможных n-пиксельных блоков. Даже для простого случая с $n = 2$ необходимо построить гистограмму распределения 65536 значений и сгенерировать 65536 кодовых слов. Для $n = 3$ потребуется уже 16777216 кодовых слов. Так что даже при малых значениях n польза от практического применения подхода, основанного на расширении источника, лимитируется его вычислительной сложностью.

В заключение заметим, что хотя выражение (8.1-7) определяет нижнюю границу сжатия, которая может быть достигнута при прямом кодировании статистически независимых значений пикселей, она преодолевается, когда значения пикселей становятся коррелированными. Блоки из коррелированных пикселей могут кодироваться с меньшим средним числом бит на пиксель, чем допускается для источника без памяти. Вместо того чтобы использовать расширения источника, обычно выбирают менее коррелированные дескрипторы (например длины серий яркости), кодируемые без расширения. Именно такой подход использовался для сжатия изображения на рис. 8.1(*б*) в разделе 8.1.2. В случае если выход источника информации зависит от конечного числа предыдущих выходов, такой источник называют *марковским источником*, или *источником с конечной памятью.*

8.1.5. Критерии верности воспроизведения

В разделе 8.1-3 отмечалось, что удаление «визуально лишней» информации влечет потерю реальной количественной информации, содержащейся в изображении. Поскольку информация теряется, требуется способ количественного описания характера потерь. В основу такого определения могут быть положены как объективные, так и субъективные критерии верности (точности) воспроизведения.

Если степень потери информации может быть выражена в виде математической функции исходного (входного) изображения и сжатого, а затем восстановленного (выходного) изображения, то такой подход называют *объективным критерием верности воспроизведения.* Примером может служить величина среднеквадратического отклонения (СКО) выходного и входного изображений. Пусть $f(x, y)$ — входное изображение, а $\hat{f}(x, y)$ — его приближение, полученное в результате операций сжатия $f(x, y)$ и последующей распаковки. Для любых x и y отклонение (ошибка, невязка) между значениями $f(x, y)$ и $\hat{f}(x, y)$ будет

$$e(x, y) = \hat{f}(x, y) - f(x, y), \tag{8.1-9}$$

так что суммарное отклонение между двумя изображениями составляет

$$\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left|\hat{f}(x,y)-f(x,y)\right|,$$

где размеры изображения равны $M\times N$. Величина e_{rms} — среднеквадратическое отклонение изображений $f(x,y)$ и $\hat{f}(x,y)$ — будет равна

$$e_{\text{rms}}=\left[\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left(\hat{f}(x,y)-f(x,y)\right)^2\right]^{1/2}. \qquad (8.1\text{-}10)$$

Если с помощью простой перестановки членов в выражении (8.1-9) рассматривать изображение $\hat{f}(x,y)$ как сумму исходного изображения $f(x,y)$ и ошибки, т. е. сигнала «шума» $e(x,y)$, то *средний квадрат отношения сигнал—шум* выходного изображения, обозначаемый SNR_{ms}, может быть определен так же, как в разделе 5.8:

$$\text{SNR}_{\text{ms}}=\frac{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\hat{f}(x,y)^2}{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left(\hat{f}(x,y)-f(x,y)\right)^2}. \qquad (8.1\text{-}11)$$

Среднеквадратическое отношение сигнал—шум, обозначаемое SNR_{rms}, получается извлечением квадратного корня из правой части выражения (8.1-11).

Объективные критерии верности воспроизведения предоставляют простой и удобный способ оценивания информационных потерь, однако восстановленные изображения в конце концов рассматриваются человеком. Следовательно, определение качества изображения с помощью субъективного оценивания часто является предпочтительным. Это может быть достигнуто путем показа восстановленного изображения группе наблюдателей (экспертов) и усреднения их оценок. Оценивание может производиться как с использованием абсолютной шкалы оценок, так и путем попарного сравнения изображений $f(x,y)$ и $\hat{f}(x,y)$. В табл. 8.2 приведен один из возможных вариантов абсолютной шкалы оценивания. Попарные сравнения с использованием такой шкалы могут быть сформированы, например, в виде набора следующих чисел: {–3, –2, –1, 0, 1, 2, 3}, отражающего, соответственно, субъективные оценки рейтинга: {*значительно хуже*, *хуже*, *слегка хуже*, *одинаково*, *слегка лучше*, *лучше*, *значительно лучше*}. В любом случае эти оценки основаны на субъективных критериях верности воспроизведения.

Пример 8.3. Сравнение оценок верности воспроизведения изображения.

■ На рис. 8.4 представлены три различных приближения изображения рис. 8.1(*а*). Подставляя в (8.1-10) изображение рис. 8.1(*а*) в качестве $f(x,y)$, а изображения рис. 8.4(*а*)—(*в*) в качестве $\hat{f}(x,y)$, после вычислений получим следующие значения СКО: 5,17, 15,67 и 14,17 уровней яркости соответственно. Согласно этим значениям — как объективным критериям верности воспроизведения — три изображения на рис. 8.4 ранжируются в порядке убывания качества в следующем порядке: {(*а*), (*в*), (*б*)}.

Изображения на рис. 8.4(*а*) и (*б*) представляют собой типичные примеры сжатия и последующего восстановления. Оба сохранили самую существенную информацию исходного изображения — такую как пространственные и яр-

а б в

Рис. 8.4. Три приближения изображения на рис. 8.1(*а*)

костные характеристики объектов. И значения их СКО грубо соответствуют ощущаемому качеству. Изображение рис. 8.4(*а*), которое практически столь же хорошо, как и исходное изображение, имеет наименьшее значение СКО; СКО изображения рис. 8.4(*б*) заметно больше, но на нем прослеживаются заметные искажения на границах между объектами. Так что результат совпадает с ожидаемым.

Изображение на рис. 8.4(*в*) — искусственно сгенерированное изображение, которое демонстрирует ограниченность объективного критерия верности. Заметим, что изображение потеряло значительную часть нескольких важных линий (т. е. визуальной информации), а также на нем появилось несколько темных квадратов (артефактов) в районе центра изображения. Визуальное содержание изображения (*в*) перестало совпадать с исходным, и оно определенно не такое точное, как изображение (*б*), однако его СКО меньше, чем у изображения (*б*) — 14,17 против 15,67 уровней яркости. Субъективные оценки данных трех изображений согласно табл. 8.2 могут выглядеть как отлично для (*а*), приемлемо или плохо для (*б*) и очень плохо или даже неприемлемо для (*в*). С другой стороны, оценки СКО ставят изображение (*в*) перед (*б*). ■

Таблица 8.2. Шкала оценок качества изображений Организации по вопросам предоставления телеканалов [Frendendall, Behrend]

Значение	Оценка	Описание
1	Отлично	Изображение чрезвычайно высокого качества, настолько хорошо, насколько только возможно.
2	Хорошо	Изображение высокого качества, оставляющее приятное впечатление. Искажения не наблюдаются.
3	Приемлемо	Изображение приемлемого качества. Искажения не наблюдаются.
4	Плохо	Изображение плохого качества; кажется, что его можно улучшить. Наблюдаются некоторые искажения.
5	Очень плохо	Очень плохое, но возможное для наблюдения изображение. Наблюдаются многочисленные искажения.
6	Неприемлемо	Изображение настолько плохо, что неприемлемо для наблюдения.

8.1.6. Модели сжатия изображений

Как видно из рис. 8.5, система сжатия содержит два принципиально разных структурных блока: *кодер* и *декодер*. Кодер осуществляет сжатие данных, а декодер выполняет обратную операцию — восстановление изображения. Обе операции могут выполняться как программно, что осуществляется многими программами редактирования или визуализации изображений, так и аппаратно, как, например, в DVD-плеерах. Устройство или программу, которая может выполнять как кодирование, так и декодирование, называют *кодеком*.

Исходное изображение $f(x,...)$ подается на кодер, который создает сжатое представление входных данных. Это представление либо запоминается для дальнейшего использования, либо передается для запоминания или использования на приемном конце. Когда сжатые данные поступают на декодер, создается восстановленное изображение $\hat{f}(x,...)$. В приложениях, связанных с неподвижными изображениями, в качестве кодируемого входа и декодированного выхода выступают $f(x,y)$ и $\hat{f}(x,y)$; в видеоприложениях таковыми соответственно являются $f(x,y,t)$ и $\hat{f}(x,y,t)$, где параметр t означает время. Вообще $\hat{f}(x,...)$ может являться точной копией $f(x,...)$, а может таковой и не быть. В первом случае систему сжатия называют *безошибочной*, *без потерь* или *сохраняющей информацию*. Во втором случае восстановленное изображение оказывается искаженным, и такой вариант сжатия называют сжатием *с потерями*.

> Нотация $f(x,...)$ обозначает здесь, что данной функцией может быть как $f(x,y)$, так и $f(x, y, t)$.

Процесс кодирования или сжатия

Конструкция блока кодера на рис. 8.5 предназначена для устранения видов избыточности данных, описанных в разделах 8.1.1—8.1.3, при их прохождении через серию из трех независимых операций. Первым шагом процесса кодирования *преобразователь* трансформирует $f(x,...)$ в формат (обычно уже не визуальный), предназначенный для сокращения пространственной и временно́й избыточности. Как правило, данная операция является обратимой и может непосредственно как сокращать, так и не сокращать количество данных, необходимое для представления изображения. Примером преобразователя, которым достигается сокращение уже на первом шаге, может служить кодирование длин серий[8] (см. разделы 8.1.2 и 8.1.5). Преобразование изображения в набор коэффициентов трансформации (см. разделы 8.2.8) является примером обратного случая; для достижения сжатия коэффициенты должны быть обработаны на дальнейших стадиях. В видеоприложениях преобразователь использует предыдущие (и в некоторых случаях также последующие) видеокадры для содействия удалению временно́й избыточности.

---

[8] Следует заметить, что это происходит только тогда, когда сигнал изображения состоит из сравнительно длинных последовательностей одинаковых значений. В иных случаях это может привести к противоположному результату, т. е. увеличению общего объема данных. — *Прим. перев.*

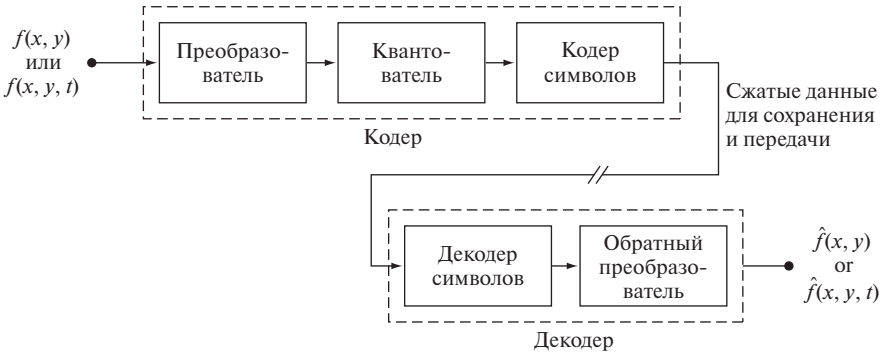


Рис. 8.5. Функциональная блок-схема обобщенной системы сжатия изображения

Квантователь на рис. 8.5 снижает точность данных на выходе преобразователя в соответствии с предварительно заданным критерием верности. Цель состоит в отбрасывании несущественной информации из сжатого представления. Как уже отмечалось в разделе 8.1.3, данная операция является необратимой; она должна быть опущена, если желательно сжатие без потерь. В видеоприложениях часто замеряется *скорость передачи данных* (в битах в секунду) кодированного выхода, и полученная величина используется для корректировки параметров квантователя с целью поддержания заранее заданной выходной скорости передачи. Тем самым, визуальное качество на выходе может меняться от кадра к кадру в зависимости от видеоконтента.

На третьей и финальной стадии процесса кодирования кодер символов на рис. 8.5 генерирует равномерный или неравномерный код, способный представить данные на выходе квантователя, и отображает их в соответствии с выбранным кодом. Во многих случаях используется неравномерный код. Наиболее короткие кодовые слова присваиваются наиболее часто появляющимся значениям на выходе квантователя, минимизируя тем самым кодовую избыточность. Данная операция обратима. Завершение данной операции означает, что изображение обработано с точки зрения устранения всех трех видов избыточности, описанных в разделах 8.1.1—8.1.3.

Процесс декодирования или восстановления

Декодер на рис. 8.5 содержит лишь два компонента: *декодер символов* и *обратный преобразователь*. Они осуществляют (в обратном порядке) операции, обратные действиям кодера символов и преобразователя в блоке кодера. Поскольку квантование приводит к необратимой потере информации, то «инверсный квантователь» в общей модели декодера отсутствует[9]. В видеоприложениях возникающие на выходе декодированные кадры временно сохраняются в некотором внутреннем устройстве памяти (не показано) и используются для внесения обратно временно́й избыточности, удаленной кодером.

[9] Однако иногда подобная операция может и присутствовать (см., например, рис. 8.39), но в подобных случаях она, как правило, означает масштабирование — обратное отображение полученного кода в соответствующее значение уровня квантования. — *Прим. перев.*

8.1.7. Форматы изображений, контейнеры и стандарты сжатия

В контексте цифрового формирования изображения стандартным способом организации и хранения данных изображения является *формат файла изображения*. Он определяет используемый способ размещения и сжатия (если оно имеется) данных. *Контейнер изображения* подобен формату файла, но может содержать разнородные типы данных изображения. *Стандарты сжатия* изображений задают процедуры сжатия и восстановления изображений, служащие для сокращения количества данных, необходимых для представления изображения. Данные стандарты являются той основой, которая способствовала широкому признанию технологии сжатия изображений.

На рис. 8.6 приведен список используемых в настоящее время наиболее важных стандартов сжатия изображений, файловых форматов и контейнеров, сгруппированных по типу поддерживаемых изображений. Черным цветом отмечены стандарты, которые одобрили следующие международные организации по стандартам: *Международная организация по стандартизации* (*International Organization for Standardization*, ISO), *Международная электротехническая комиссия*, МЭК (*International Electrotechnical Commission*, IEC) и/или *Сектор стандартизации телекоммуникаций Международного союза электросвязи* (*International Telecommunications Union — Telecommunications*, ITU-T) — специальный орган ООН, организация, ранее называвшаяся *Международный консультационный комитет по телефонии и телеграфии*, МККТТ (*Consultative Committee of the International Telephone and Telegraph*, CCITT). Также включены два стандарта сжатия видео: VC-1 от *Общества инженеров кино и телевидения*, ОИКиТ (*Society of Motion Pictures and Television Engineers*, SMPTE), а также AVS от *Китайского министерства по информационной отрасли* (*Chinese Ministry of Information Industry*, MII).

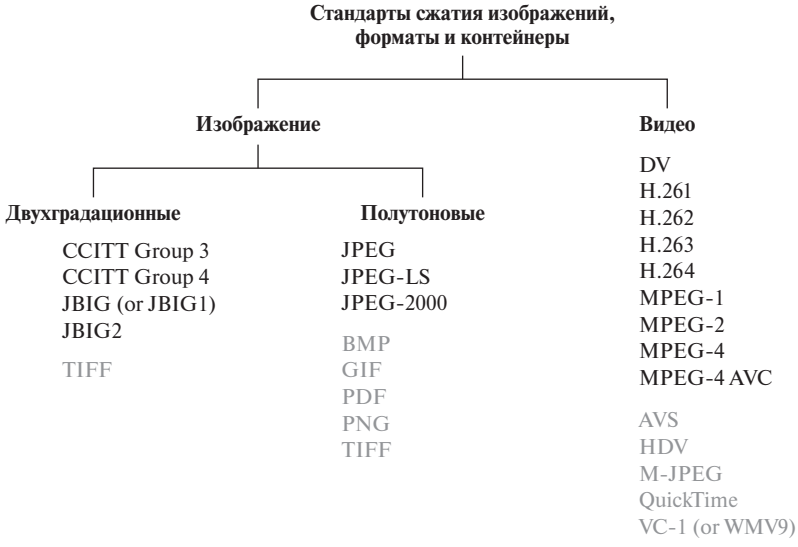


Рис. 8.6. Некоторые популярные стандарты сжатия изображений, файловые форматы и контейнеры. Стандарты, одобренные международными комитетами, приведены черным; остальные — серым цветом

Таблица 8.3. Стандарты сжатия изображений, одобренные соответствующими международными комитетами. В квадратных скобках приведены ссылки на разделы настоящей главы

Название	Организация	Описание
Двухградационные изображения		
CCITT Group 3	ITU-T	Разработан как факсимильный (FAX) метод для передачи двухградационных документов по телефонным линиям. Поддерживает одномерное и двумерное кодирование длин серий [8.2.5] и кодирование Хаффмана [8.2.1].
CCITT Group 4	ITU-T	Упрощенная и модифицированная версия стандарта CCITT Group 3, поддерживающая только двумерное кодирование длин серий.
JBIG *или* JBIG1	ISO/IEC/ ITU-T	Стандарт Объединенной группы экспертов по двухградационным изображениям (*Joint Bi-level Image Experts Group*) для последовательного сжатия двухградационных изображений без потерь. Допускает сжатие полутоновых изображений, содержащих до 6 битовых плоскостей [8.2.7]. Используется контекстно-зависимый алгоритм арифметического кодирования [8.2.3]; изначально низкое разрешение изображения может увеличиваться постепенным добавлением дополнительных сжатых данных.
JBIG2	ISO/IEC/ ITU-T	Модификация JBIG1 для двухградационных изображений в экранных, интернет- и факс-приложениях. Метод сжатия зависит от контента; для текста и полутоновых областей используется метод, основанный на словаре шаблонов [8.2.6], а для остальных — кодирование Хаффмана [8.2.1] или арифметическое [8.2.3] кодирование.
Полутоновые изображения		
JPEG	ISO/IEC/ ITU-T	Стандарт *Объединенной группы экспертов по фотографии* (*Joint Photographic Experts Group*) для изображений фотографического качества. Его базовая система сжатия с потерями (реализованная в подавляющем большинстве случаев) использует квантование коэффициентов дискретного косинусного преобразования (ДКП) по блокам изображения размерами 8×8 пикселей [8.2.8], кодирование Хаффмана [8.2.1] и кодирование длин серий [8.2.5]. Является одним из наиболее популярных способов кодирования изображений в Интернете.
JPEG-LS	ISO/IEC/ ITU-T	Стандарт сжатия больших полутоновых изображений без потерь (или с ограниченными, задаваемыми пользователем потерями). Базируется на адаптивном предсказании [8.2.9], контекстном моделировании [8.2.3] и кодировании Голомба [8.2.2].
JPEG-2000	ISO/IEC/ ITU-T	Модификация JPEG для повышенного сжатия изображений фотографического качества. Используются квантование дискретного вейвлет-преобразования (ДВП) [8.2.10] и арифметическое кодирование [8.2.3]. Сжатие может быть с потерями или без потерь.

Таблица 8.3. (*Продолжение*)

Название	Организация	Описание
Видео		
DV	IEC	Цифровое видео (*Digital Video*). Видеостандарт, разработанный для домашнего видео и полупрофессиональных видеоприложений и оборудования — таких как накопление электронных новостей и записывающие видеокамеры. Кадры сжимаются независимо для упрощения редактирования. Используется сжатие, основанное на ДКП [8.2.8], аналогичное JPEG.
H.261	ITU-T	Стандарт двусторонней видеоконференции для линий цифровых сетей с интеграцией обслуживания (*integrated services digital network*, ISDN). Поддерживает изображения размерами 352×288 и 176×144 без чередования, называемые CIF (*Common Intermediate Format*) и QCIF (*Quarter CIF*) соответственно. Используется основанный на ДКП вариант сжатия [8.2.8], аналогичный JPEG. Для уменьшения временно́й избыточности используется разность с предсказанием по соседнему кадру [8.2.9]. Для компенсации движения между кадрами используется метод сдвига блоков.
H.262	ITU-T	См. MPEG-2 ниже.
H.263	ITU-T	Улучшенный вариант H.261, разработанный для обычных телефонных модемов (т. е. 28,8 Кбод) с дополнительными вариантами разрешения: SQCIF (*Sub-Quarter CIF*, 128×96), 4CIF (704×576) и 16CIF (1408×512).
H.264	ITU-T	Расширение H.261—H.263 для видеоконференций, интернет-потока и широковещательного телевидения. Используются разности с внутрикадровыми предсказаниями [8.2.9], переменный размер блока (4×4 или 8×8), целочисленные преобразования (а не ДКП) и контекстно-адаптивное арифметическое кодирование [8.2.3].
MPEG-1	ISO/IEC	Стандарт *Группы экспертов по движущимся изображениям* (*Motion Pictures Expert Group*) для компакт-дисков (CD-ROM) и видеопотоков скоростью до 1,5 Мбит/с без чередования. Он подобен H.261, но предсказание кадров может базироваться на предыдущем кадре, следующем кадре или на их интерполяции. Поддерживается почти всеми компьютерами и DVD-плеерами.
MPEG-2	ISO/IEC	Расширение MPEG-1, предназначенное для DVD со скоростями передачи до 15 Мбит/с. Поддерживает чересстрочное видео и телевидение высокой четкости (HDTV). Наиболее успешный видеостандарт на данный момент.
MPEG-4	ISO/IEC	Расширение MPEG-2, поддерживающее блоки различного размера и разности с внутрикадровыми предсказаниями [8.2.9].
MPEG-4 AVC	ISO/IEC	MPEG-4, часть 10. *Advanced Video Coding* (AVC). Идентичен H.264 выше.

Таблица 8.4. Популярные стандарты сжатия изображений, файловые форматы и контейнеры, не вошедшие в табл. 8.3.

Название	Организация	Описание
Полутоновые изображения		
BMP	Microsoft	*Windows Bitmap*. Файловый формат, использующийся в основном для простых несжатых изображений.
GIF	CompuServe	*Graphic Interchange Format*. Файловый формат, использующий LZW-кодирование без потерь [8.2.4] для изображений глубиной от 1 до 8 битов. Часто используется для изготовления небольших анимаций и коротких фильмов малого разрешения для Интернета.
PDF	Adobe Systems	*Portable Document Format*. Формат представления документов независимо от устройства вывода и его разрешения. Может служить контейнером для JPEG, JPEG-2000, CCITT и других сжатых изображений. Некоторые версии PDF стали стандартами ISO.
PNG	Интернет-консорциум (W3C)	*Portable Network Graphics*. Файловый формат, сжимающий без потерь изображения в натуральных цветах с добавлением прозрачности (до 48 бит/пиксель). Использует кодирование разности между значениями каждого пикселя и предшествующих ему пикселей [8.2.9].
TIFF	Aldus	*Tagged Image File Format*. Гибкий файловый формат, поддерживающий множество разных стандартов сжатия, включая JPEG, JPEG-LS, JPEG-2000, JBIG2 и другие.
Видео		
AVS	MII	*Audio-Video Standard*. Формат, аналогичный H.264, но использующий экспоненциальные коды Голомба [8.2.2]. Разработан в Китае.
HDV	Консорциум компаний	*High Definition Video*. Расширение DV для телевидения высокого разрешения; используется сжатие MPEG-2, включая устранение временно́й избыточности при помощи разности с предсказанием [8.2.9].
M-JPEG	Различные компании	*Motion JPEG*. Формат сжатия, в котором каждый кадр сжимается независимо, используя алгоритм JPEG.
QuickTime	Apple Computer	Контейнер мультимедийных данных, поддерживающий DV, H.261, H.262, H.264, MPEG-1, MPEG-2, MPEG-4 и другие форматы сжатия видео.
VC-1 WMV9	SMPTE Microsoft	Наиболее используемый в Интернете видеоформат. Принят для HD и *Blu-ray* DVD высокого разрешения. Аналогичен H.264/AVC в использовании целочисленного ДКП с различными размерами блоков [8.2.8 и 8.2.9] и контекстно-зависимыми неравномерными кодовыми таблицами [8.2.1], но без использования внутрикадрового предсказания.

Последние отмечены на рис. 8.6 серым цветом, использующимся для пометки стандартов, не одобренных международными организациями по стандартам.

Краткие описания стандартов, форматов и контейнеров, приведенных на рис. 8.6, сведены в табл. 8.3 и 8.4, где также указаны несущие ответственность организации, намеченные области применения и ключевые методы сжатия. Сами же методы сжатия являются предметом рассмотрения следующего раздела. В обеих таблицах в квадратных скобках указаны прямые ссылки на соответствующие подразделы раздела 8.2.

8.2. Некоторые основные методы сжатия

В данном разделе рассматриваются важнейшие на сегодняшний день методы сжатия — как без потерь, так и с потерями. Основное внимание уделено методам, доказавшим свою актуальность применением в основных стандартах сжатия двоичных, полутоновых изображений и видеопоследовательностей. Сами стандарты используются для иллюстрации обсуждаемых методов.

8.2.1. Кодирование Хаффмана

Один из наиболее известных методов сокращения кодовой избыточности был предложен Хаффманом [Huffman, 1952]. При независимом кодировании символов источника информации *коды Хаффмана* обеспечивают наименьшее число кодовых символов (битов) на символ источника. В терминах первой теоремы Шеннона (см. раздел 8.1.4) результат является оптимальным для фиксированного значения n при условии, что символы источника кодируются *по отдельности*. На практике исходными символами могут быть как значения яркости элементов изображения, так и выходы некоторых преобразований яркостей (разность значений пикселей, длины серий и т. д.).

Согласно таблицам 8.3 и 8.4 коды Хаффмана используются в стандартах сжатия:

- CCITT,
- JBIG2,
- JPEG,
- MPEG-1, 2, 4,
- H.261, H.262, H.263, H.264

и некоторых других.

Первым шагом в подходе Хаффмана является построение серии (множества уровней) редуцированных источников путем упорядочивания вероятностей рассматриваемых символов и «склеивания» символов с наименьшими вероятностями в один символ, который будет замещать их в редуцированном источнике следующего уровня. Рис. 8.7 иллюстрирует этот процесс для двоичного кодирования (аналогично могут быть построены K-символьные коды Хаффмана). В левой колонке приведен гипотетический набор символов источника, а в следующей — их вероятности; символы упорядочены сверху вниз в порядке убывания вероятностей. Для формирования первой редукции источника символы

Рис. 8.7. Редукция источника по Хаффману

с наименьшими вероятностями — в данном случае 0,06 и 0,04 — объединяются в некоторый «составной символ» с суммарной вероятностью 0,1. Этот составной символ и связанная с ним вероятность помещаются в список символов редуцированного источника, который опять упорядочивается в порядке убывания значений полученных вероятностей. Этот процесс повторяется до тех пор, пока не образуется редуцированный источник всего лишь с двумя оставшимися символами (самая правая колонка на рисунке).

Второй шаг в процедуре кодирования по Хаффману состоит в кодировании каждого из редуцированных источников, начиная с источника с наименьшим числом символов (т. е. правого на рис. 8.11) и возвращаясь обратно к исходному источнику. Для источника с двумя символами наименьшим двоичным кодом являются, конечно, символы 0 и 1. Как показано на рис. 8.8, эти символы приписываются символам источника справа (выбор символов произволен — изменение 0 на 1 и наоборот даст абсолютно тот же результат). Поскольку символ текущего редуцированного источника, имеющий вероятность 0,6, был получен объединением двух символов предыдущего редуцированного источника, то кодовый бит 0, выбранный для данного варианта, приписывается *каждому* из двух соответствующих символов предыдущего источника, затем эти коды произвольным образом дополняются символами 0 и 1, для отличия их друг от друга. Эта операция затем повторяется для редуцированных источников всех остальных уровней, пока не будет достигнут уровень исходного источника. Результирующий код приведен в самой левой колонке (Код) на рис. 8.8. Средняя длина этого кода составит

$$L_{\text{avg}} = 0{,}4 \cdot 1 + 0{,}3 \cdot 2 + 0{,}1 \cdot 3 + 0{,}1 \cdot 4 + \cdot 0{,}06 \cdot 5 + 0{,}04 \cdot 5 = 2{,}2 \text{ бит/символ.}$$

Энтропия источника, согласно (8.1-6), равна 2,14 бита/символ.

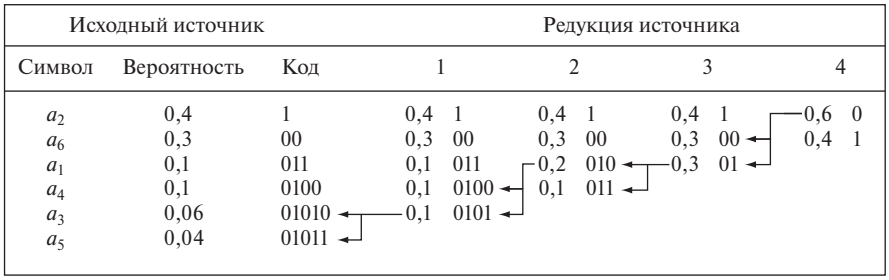


Рис. 8.8. Процедура построения кода Хаффмана

Процедура Хаффмана строит оптимальный код для набора символов и их вероятностей при том ограничении, что символы кодируются по отдельности. После того, как код построен, процесс кодирования/декодирования без потерь осуществляется простым табличным преобразованием. Код Хаффмана является *мгновенным однозначно декодируемым блоковым кодом*. Он называется *блоковым кодом*, поскольку каждый символ источника отображается в фиксированную последовательность кодовых символов. Он является *мгновенным*, потому что каждое кодовое слово в строке кодовых символов может быть декодировано независимо от последующих символов. Он является *однозначно декодируемым*, т. к. любая строка из кодовых символов может быть декодирована единственным образом. Таким образом, любая строка кодированных по Хаффману символов может декодироваться анализом отдельных символов в строке слева направо. Для двоичного кода, представленного на рис. 8.8, анализ слева направо показывает, что в закодированной строке 010100111100 первым правильным кодовым словом является 01010, которое есть код для символа a_3. Следующим правильным кодовым словом является 011, что соответствует символу a_1. Продолжая эти действия, получим декодированное сообщение в виде $a_3a_1a_2a_2a_6$.

Пример 8.4. Кодирование по Хаффману.

■ Монохромное изображение размерами 256×256×8 бит на рис. 8.9(*а*) имеет гистограмму яркостей, представленную на рис. 8.9(*б*). Поскольку яркости его не равновероятны, использовалась процедура Хаффмана, реализованная в MATLAB. Включая в общий объем полученную таблицу кодов Хаффмана, требуемую для восстановления яркостей исходного 8-битового изображения, для кодирования потребовалось в среднем 7,428 бит/пиксель. Полученный сжатый вариант превышает оценку энтропии изображения (7,3838 бит/пиксель согласно уравнению (8.1-7)) на 5122×(7,428 – 7,3838) или 11587 бит, т. е. около 0,6 %. Итоговый коэффициент сжатия и соответствующая относительная избыточность будут равны: C = 8/7,428 = 1,077 и R = 1 – (1/1,077) = 0,0715. Таким образом, исходный вариант представления изображения 8-битовым равномерным кодом удалось сократить на 7,15 % за счет кодовой избыточности. ■

В случае кодирования большого числа символов построение оптимального кода Хаффмана становится нетривиальной задачей. В общем случае для *J* сим-

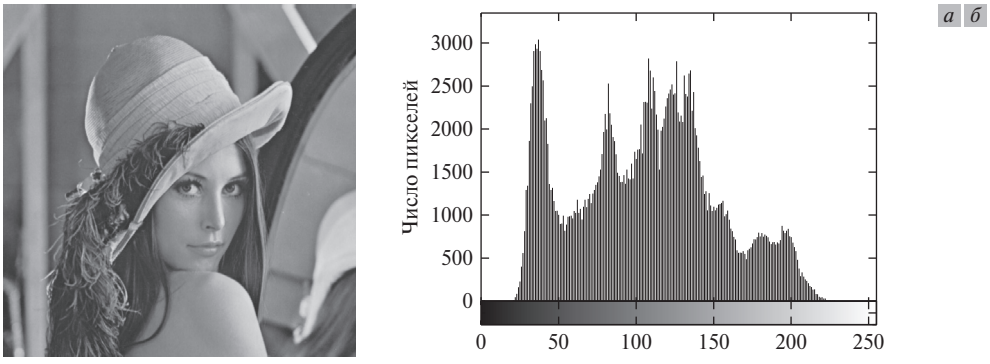


Рис. 8.9. (*а*) 8-битовое изображение размерами 256×256 и (*б*) его гистограмма

волов источника и J вероятностей символов требуется $J-2$ редукций источника и $J-2$ перекодировок. Если вероятности символов источника можно оценить заранее, то используя предварительно вычисленные коды Хаффмана, зачастую можно достичь «почти оптимального» кодирования. Некоторые из распространенных стандартов сжатия, включая обсуждаемые в разделах 8.2.8 и 8.2.9 стандарты JPEG и MPEG, имеют фиксированные таблицы кодов Хаффмана, которые были вычислены изначально на основе экспериментальных данных.

8.2.2. Кодирование Голомба

Рассмотрим кодирование неотрицательных целых значений с экспоненциально затухающим распределением вероятностей. Данные такого типа могут быть оптимально кодированы (в смысле первой теоремы Шеннона) с помощью семейства кодов, которые в вычислительном отношении проще кодов Хаффмана. Такие коды первоначально были предложены для представления неотрицательных длин серий [Golomb, 1966]. В последующем рассмотрении нотация $\lfloor x \rfloor$ означает «наибольшее целое, равное или меньше, чем x»; $\lceil x \rceil$ — «наименьшее целое, равное или больше, чем x»; а запись $x \bmod y$ — «x по модулю y», т. е. остаток от деления x на y.

Согласно таблицам 8.3 и 8.4 коды Голомба используются в стандартах сжатия:
JPEG-LS,
AVS.

Пусть задано некоторое неотрицательное целое n и положительный целый *делитель* $m > 0$. *Код Голомба* значения n по отношению к m, обозначаемый $G_m(n)$, есть комбинация унарного кода частного $\lfloor n/m \rfloor$ и двоичного представления *остатка* $n \bmod m$. $G_m(n)$ находится следующим образом.

Шаг 1. Сформировать унарный код частного $\lfloor n/m \rfloor$. (*Унарный код* целого q определяется как q единиц, за которыми идет 0.)

Шаг 2. Пусть $k = \lceil \log_2 m \rceil$, $c = 2^k - m$, $r = n \bmod m$; усеченный остаток r' вычисляется как

$$r' = \begin{cases} r \text{ усеченное до } k-1 \text{ бита} & \text{если } 0 \le r < c, \\ r+c \text{ усеченное до } k \text{ бит} & \text{в остальных случаях.} \end{cases} \qquad (8.2\text{-}1)$$

Шаг 3. Выполнить конкатенацию результатов шагов 1 и 2.

В качестве примера вычислим $G_4(9)$. На шаге 1 определим унарный код частного $\lfloor 9/4 \rfloor = \lfloor 2{,}25 \rfloor = 2$, который будет 110. На шаге 2 найдем $k = \lceil \log_2 4 \rceil = 2$, $c = 2^2 - 4 = 0$, $r = 9 \bmod 4$, что в двоичной форме будет 1001 mod 0100, т. е. 0001. В соответствии с (8.2-1) r' равен r (т. е. 0001), усеченному до 2 бит, что дает 01 (результат шага 2). На шаге 3 осуществляем конкатенацию результатов шага 1 (110) и шага 2 (01) и получаем 11001, что и составит $G_4(9)$.

Для частного случая $m = 2^k$, $c = 0$ и $r' = r = n \bmod m$ согласно (8.2-1) обрезается до k бит для всех n. Деления, требуемые для формирования кода Голомба, становятся операциями сдвига двоичного кода; данные коды, более простые в вычислительном отношении, называются кодами *Голомба — Райса* или *кодами Райса* [Rice, 1975]. В колонках 2, 3 и 4 табл. 8.5 приведен список кодов G_1, G_2 и G_4

Таблица 8.5. Несколько кодов Голомба для значений 0—9

n	$G_1(n)$	$G_2(n)$	$G_4(n)$	$G^0_{\exp}(n)$
0	0	00	000	0
1	10	01	001	100
2	110	100	010	101
3	1110	101	011	11000
4	11110	1100	1000	11001
5	111110	1101	1001	11010
6	1111110	11100	1010	11011
7	11111110	11101	1011	1110000
8	111111110	1111000	11000	1110001
9	1111111110	1111001	11001	1110010

для первых десяти неотрицательных целых. Поскольку m является степенью 2 (т. е. $1 = 2^0$, $2 = 2^1$ и $4 = 2^2$), они составляют первые три кода Голомба — Райса. Более того, G_1 является унарным кодом неотрицательных целых, поскольку $\lfloor n/1 \rfloor = n$ и $n \bmod 1 = 0$ для всех n.

Поскольку коды Голомба могут отражать лишь неотрицательные значения и существует много кодов Голомба, из которых можно выбирать, ключевым шагом на пути их эффективного использования является выбор делителя m. В случае *геометрического распределения* исходных целых значений с *функцией вероятности*[10] (ФВ), равной

$$P(n) = (1-\rho)\rho^n \tag{8.2-2}$$

для некоторого $0 < \rho < 1$, можно показать, что коды Голомба являются оптимальными — в том смысле, что $G_m(n)$ обеспечивает минимальную среднюю кодовую длину среди всех однозначно декодируемых кодов — когда [Gallager, Voorhis, 1975]

$$m = \left\lceil \frac{\log_2(1+\rho)}{\log_2(1/\rho)} \right\rceil. \tag{8.2-3}$$

Дискретное распределение вероятностей, определенное согласно (8.2-2), называется геометрическим распределением. Его непрерывным аналогом является экспоненциальное распределение.

На рис. 8.10(*а*) приведены графики функции (8.2-2) для трех значений ρ; данные распределения вероятностей символов хорошо поддерживаются кодом Голомба (т. е. с достаточной кодовой эффективностью). Как видно на рисунке, малые значения являются значительно более вероятными, чем большие.

[10] *Функция вероятности* (ФВ) — функция, определяющая вероятности дискретных случайных величин. Принципиальное отличие ФВ от *плотности* распределения вероятностей (ПРВ) в том, что значение ПРВ не является вероятностью как таковой; вероятностью станет лишь интеграл ПРВ на некотором интервале.

а б в

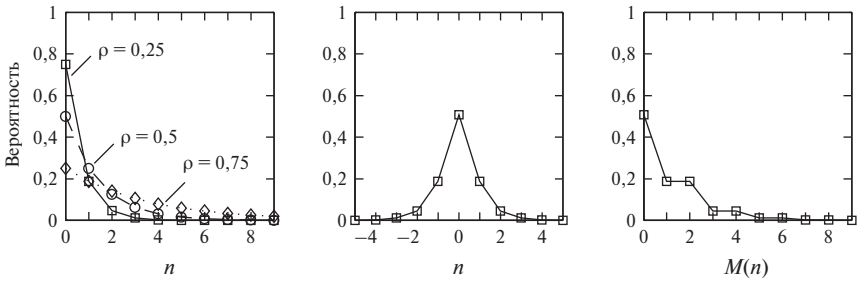


Рис. 8.10. (*а*) Три односторонних геометрических распределения согласно формуле (8.2-2); (*б*) двустороннее экспоненциально спадающее распределение; (*в*) переупорядоченный с помощью (8.2-4) вариант (*б*)

Графическим представлением ФВ является гистограмма.

Поскольку едва ли вероятности значений яркостей реального изображения (см., например, гистограмму на рис. 8.9(*б*)) соответствуют вероятностям, задаваемым формулой (8.2-2) и показанным на рис. 8.10(*а*), то коды Голомба редко применяются для кодирования яркостей. Однако когда кодируются разности яркостей, вероятности получающихся «значений разностей» (см. раздел 8.2.9) — с известным исключением отрицательных значений — часто имеют сходство с функцией (8.2-2) и рис. 8.10(*а*). Для включения отрицательных разностей в работу с кодами Голомба, которые могут представлять лишь неотрицательные целые, обычно используется отображение (переупорядочивание) вида

$$M(n)=\begin{cases} 2n & \text{если } n\geq 0, \\ 2|n|-1 & \text{если } n<0. \end{cases} \qquad (8.2\text{-}4)$$

Используя это отображение, двусторонняя ФВ, показанная на рис. 8.10(*б*), трансформируется в одностороннюю функцию на рис. 8.10(*в*). Значения переупорядочены так, что исходные положительные значения раздвинуты, а отрицательные заняли освободившиеся позиции нечетных положительных значений. Если $P(n)$ была центрированной в нуле двусторонней функцией, то $P(M(n))$ становится односторонней. Отображенные значения $M(n)$ теперь могут быть эффективно кодированы при помощи подходящего кода Голомба — Райса [Weinberger et al., 1996].

Пример 8.5. Кодирование Голомба — Райса.

■ Рассмотрим опять изображение рис. 8.1(*в*) и отметим, что его гистограмма (рис. 8.3(*а*)) подобна двустороннему распределению на рис. 8.10(*б*). Если через n ($0 \le n \le 255$) обозначить значение яркости на изображении, а через μ — среднюю яркость, то $P(n - \mu)$ будет двусторонним распределением, показанным на рис. 8.11(*а*). Этот график получен нормировкой гистограммы рис. 8.3(*а*) на общее число пикселей изображения и сдвигом влево на 128 (что эквивалентно вычитанию средней яркости, равной 128, из значений элементов изображения). После этого согласно (8.2-4) сформировано одностороннее распределение $P(M(n - \mu))$, показанное на рис. 8.11(*б*). Если переупорядоченные значения затем кодировать кодом Голомба, используя реализацию MATLAB кода G_1

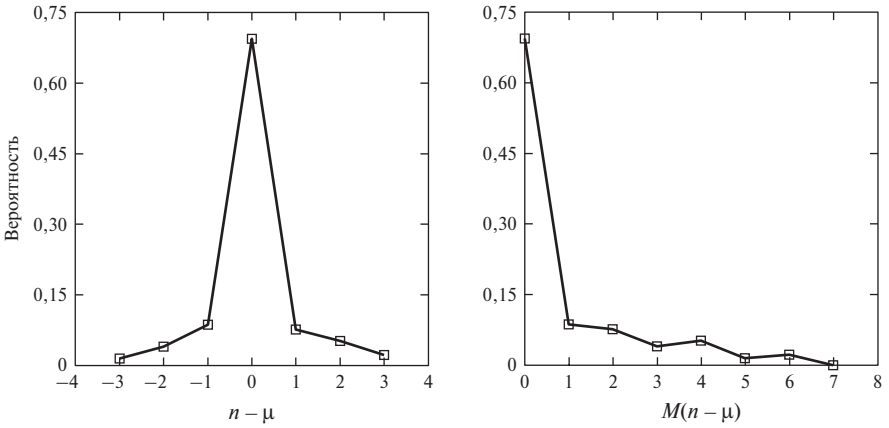


Рис. 8.11. (*а*) Распределение вероятностей значений элементов изображения рис. 8.1(*в*) после вычитания средней яркости; и (*б*) вариант (*а*), переупорядоченный согласно (8.2-4)

в колонке 2 табл. 8.5, результат кодирования окажется в 4,5 раза меньше исходного изображения ($C = 4{,}5$). Т. е. код G_1 реализует 4,5/5,1 или 88 % теоретически возможного сжатия, достижимого неравномерным кодированием. (Основываясь на вычисленной в примере 8.2 энтропии, максимально возможный коэффициент сжатия при помощи неравномерного кодирования составляет $C = 8/1{,}566 \approx 5{,}1$.) Более того, кодирование Голомба достигает 96 % от сжатия, обеспечиваемого кодированием Хаффмана, также реализованного в MATLAB, и к тому же не требует вычисления индивидуальной таблицы, аналогичной кодам Хаффмана.

Теперь рассмотрим изображение на рис. 8.9(*а*). Если его значения кодировать тем же кодом Голомба G_1, как и выше, получим $\underline{C} = 0{,}09022$. Это означает, что возникло увеличение объема данных. Это произошло потому, что вероятности яркостей изображения на рис. 8.9(*а*) очень сильно отличаются от вероятностей, определяемых формулой (8.2-2). Аналогичным образом и коды Хаффмана могут привести к увеличению объема данных, если применять их для кодирования множества символов, вероятности которых отличаются от того распределения, для которого код был построен. На практике, чем дальше отходить от предположений о вероятностях исходных данных, для которых код построен, тем больше риск получения плохих характеристик сжатия и увеличения объема данных. ■

Если C меньше 1 в (8.1-2), это означает увеличение объема данных.

Завершая рассмотрение кодов Голомба, обратимся к колонке 5 табл. 8.5, содержащей первые 10 кодов нулевого порядка *экспоненциального кода Голомба*, обозначенного $G^0_{\exp}(n)$. Экспоненциальные коды Голомба полезны при кодировании длин серий, потому что и короткие, и длинные серии кодируются эффективно. Экспоненциальный код Голомба $G^k_{\exp}(n)$ порядка k вычисляется следующим образом.

Шаг 1. Найти целое $i \geq 0$, такое, что

$$\sum_{j=0}^{i-1} 2^{j+k} \leq n < \sum_{j=0}^{i} 2^{j+k} \tag{8.2-5}$$

и сформировать унарный код значения i. Если $k = 0$, то $i = |\log_2(n + 1)|$; такой код также известен как *гамма-код Элайеса*.

Шаг 2. Обрезать двоичное представление

$$n - \sum_{j=0}^{i-1} 2^{j+k} \tag{8.2-6}$$

до $k + i$ младших битов.

Шаг 3. Выполнить конкатенацию результатов шагов 1 и 2.

Чтобы, например, найти $G^8_{\exp}(8)$, положим $i = |\log_2(9)|$ или 3 на шаге 1, поскольку $k = 0$. Тогда неравенство (8.2-5) выполняется, поскольку

$$\sum_{j=0}^{3-1} 2^{j+0} \leq 8 < \sum_{j=0}^{3} 2^{j+0}$$

$$\sum_{j=0}^{2} 2^{j} \leq 8 < \sum_{j=0}^{3} 2^{j}$$

$$2^0 + 2^1 + 2^3 \leq 8 < 2^0 + 2^1 + 2^2 + 2^3$$

$$7 \leq 8 < 15.$$

Унарным кодом 3 является 1110. На втором шаге согласно (8.2-6) получим

$$8 - \sum_{j=0}^{3-1} 2^{j+0} = 8 - \sum_{j=0}^{2} 2^{j} = 8 - (2^0 + 2^1 + 2^2) = 8 - 7 = 1 = 0001,$$

что затем обрезается до 3+0 младших битов и становится 001. Конкатенация результатов шагов 1 и 2 дает код 1110001. Заметим, что в пятом столбце табл. 8.5 это код в строчке с $n = 8$. Напоследок заметим, что, как и коды Хаффмана в предыдущем разделе, коды Голомба являются мгновенно и однозначно декодируемыми неравномерными блоковыми кодами.

8.2.3. Арифметическое кодирование

В отличие от неравномерных кодов, рассмотренных в предыдущих двух разделах, арифметическое кодирование создает неблоковые коды. В арифметическом кодировании, история которого может быть прослежена вплоть до работ Элайеса (Elias, см. [Abramson, 1963]), не существует однозначного соответствия между символами источника и кодовыми словами. Вместо этого вся последовательность символов источника (т. е. все сообщение) соотнесена с одним арифметическим кодовым словом. Само по себе кодовое слово задает интервал вещественных чисел между 0 и 1. С увеличением числа символов в сообщении интервал, необходимый для их представления, уменьшается, а число единиц информации (скажем, битов), требуемых для представления интервала, увеличивается. Каждый символ в сообщении уменьшает размер интервала в соответствии с вероятностью своего появления. Поскольку метод не требует, как например подход Хаффмана, чтобы каждый исходный символ отображался в целое число кодо-

вых слов (т. е. чтобы символы кодировались по одному), он достигает (в теории) границы, установленной первой теоремой Шеннона (см. раздел 8.1.4).

Согласно таблицам 8.3 и 8.4 арифметическое кодирование используется в стандартах сжатия:

- JBIG1,
- JBIG2,
- JPEG-2000,
- H.264,
- MPEG-4 AVC,

а также в других.

На рис. 8.12 проиллюстрирован основной процесс арифметического кодирования. Здесь кодируется сообщение из пяти символов, порожденное четырехсимвольным источником $a_1a_2a_3a_3a_4$. В начале процесса кодирования предполагается, что сообщение занимает весь полуоткрытый интервал [0, 1). Этот интервал изначально делится на четыре отрезка пропорционально вероятностям символов источника, которые приведены в табл. 8.6. Символу a_1, например, соответствует подынтервал [0, 0,2). Поскольку это первый символ кодируемого сообщения, то, значит, интервал оставшейся части сообщения ($a_2a_3a_3a_4$) будет сужен до [0, 0,2). На рис. 8.12 полученный интервал растянут на полную высоту рисунка, и приведенные значения на его концах соответствуют значениям суженного диапазона. Затем суженный диапазон также делится на отрезки пропорционально вероятностям символов источника, и процесс повторяется со следующим символом сообщения. Таким образом, символ a_2 сузит подынтервал до [0,04, 0,08), a_3 — до [0,056, 0,072) и т. д. Последний символ сообщения, который должен быть зарезервирован для специального индикатора окончания сообщения, сужает диапазон до [0,06752, 0,0688). Конечно, любое число в этом подынтервале — например 0,068 — может быть использовано для представления сообщения.

Таблица 8.6. Пример арифметического кодирования

Символ источника	Вероятность	Исходный подинтервал
a_1	0,2	[0,0, 0,2)
a_2	0,2	[0,2, 0,4)
a_3	0,4	[0,4, 0,8)
a_4	0,2	[0,8, 1,0)

Сообщение из пяти символов, приведенное на рис. 8.12, после арифметического кодирования требует для записи всего трех десятичных цифр. Это соответствует 3/5 или 0,6 десятичных знаков на символ источника и весьма близко энтропии источника, которая, согласно (8.1-6), составляет 0,58 десятичных знаков на символ источника. При увеличении длины кодируемой последовательности результирующий арифметический код приближается к границе, устанавливаемой первой теоремой Шеннона. На практике два фактора мешают кодовым характеристикам приблизиться к данной границе вплотную: (1) необходимость включения некоторого символа окончания, позволяющего отделять одну кодо-

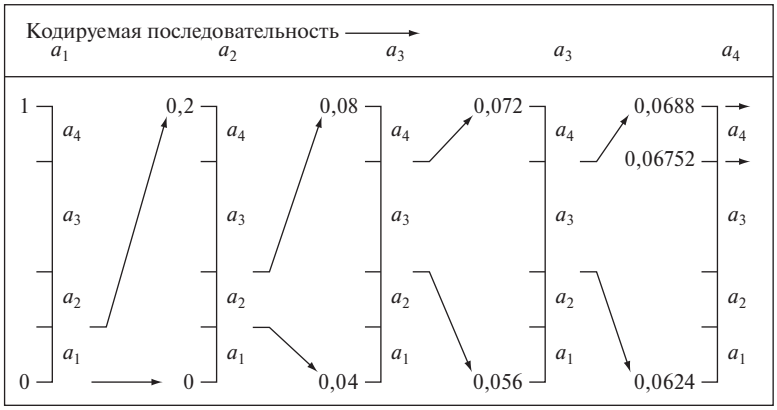


Рис. 8.12. Процедура арифметического кодирования

вую последовательность от другой, и (2) использование арифметики конечной точности. Для преодоления последней проблемы при практической реализации арифметического кодирования применяются стратегии масштабирования и округления [Langdon, Rissanen, 1981]. Согласно стратегии масштабирования каждый подинтервал перед разбиением его на отрезки, пропорциональные вероятностям символов, растягивается до диапазона [0, 1). Стратегия округления гарантирует, что ограничения, связанные с конечной точностью вычислений, не препятствуют точному представлению кодовых подынтервалов.

Адаптивные контекстно-зависимые оценки вероятностей

В случае точных *вероятностных моделей* символов, т. е. моделей, дающих верные вероятности множества кодируемых символов, арифметическое кодирование становится близко к оптимальному в смысле минимизации среднего числа кодовых символов, требуемых для представления исходных кодируемых символов. Как и в случае с кодами Хаффмана и Голомба, неверные вероятностные модели могут привести к неоптимальному результату. Простой способ повышения точности используемых вероятностей состоит в использовании адаптивных, контекстно-зависимых вероятностных моделей. *Адаптивные* вероятностные модели обновляют вероятности символов по мере того, как символы кодируются или становятся известными. *Контекстно-зависимые* модели дают вероятности, базирующиеся на предварительно определенной окрестности пикселей, называемой *контекстом*, вокруг кодируемого символа. Как правило, используется *каузальный контекст* — ограниченный набором символов, которые уже закодированы. Как Q-кодер [Pennebaker et al., 1988], так и MQ-кодер [ISO/IEC, 2000] — два хорошо известных метода арифметического кодирования, которые встроены в JBIG, JPEG-2000 и другие важные стандарты сжатия изображений — используют вероятностные модели, являющиеся как адаптивными, так и контекстно-зависимыми. Q-кодер динамически обновляет вероятности символов на этапе перенормировки интервалов, который является частью алгоритма арифметического кодирования. Адаптивные контекстно-зависимые модели используются также в кодировании Голомба, например в стандарте сжатия JPEG-LS.

На рис. 8.13 представлена диаграмма шагов адаптивного, контекстно-зависимого процесса арифметического кодирования символов *двоичного* источ-

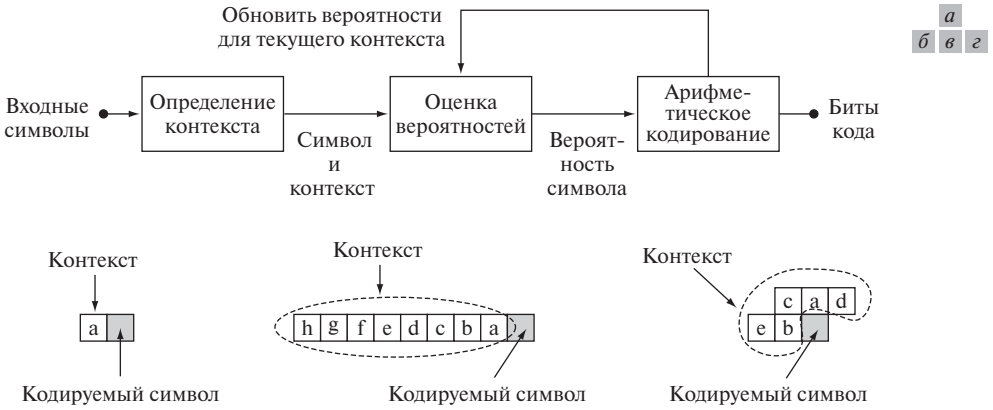


Рис. 8.13. (*а*) Адаптивный контекстно-зависимый процесс арифметического кодирования (часто используется для кодирования символов двоичного источника). (*б*)—(*г*) Три возможных модели контекста

ника. Арифметическое кодирование часто применяется, когда нужно кодировать двоичные последовательности. Как только очередной символ (или бит) начинает процесс кодирования, его контекст формируется в блоке *«Определение контекста»* рис. 8.13(*а*). На рис. 8.13(*б*)—(*г*) представлены три возможные контекста, которые могут использоваться: (*б*) ближайший предшествующий символ, (*в*) группа предшествующих символов и (*г*) несколько предшествующих символов плюс несколько символов на предыдущей строке. Для этих трех случаев блок *«Оценка вероятностей»* должен поддерживать 2^1 (или 2), 2^8 (или 256) и 2^5 (или 32) значений контекста и связанных с ними вероятностей. Например, если используется контекст рис. 8.13(*б*), должны прослеживаться следующие условные вероятности: $P(0|a = 0)$ (вероятность, что кодируемый символ равен 0 при условии, что предыдущий символ равен 0), $P(1|a = 0)$, $P(0|a = 1)$ и $P(1|a = 1)$. Соответствующие вероятности как функция текущего контекста затем передаются в блок *«Арифметическое кодирование»*, и запускают процесс арифметического кодирования и формирования выходной последовательности в соответствии с процедурой, проиллюстрированной на рис. 8.12. Вероятности, ассоциированные с контекстом символа, кодируемого на текущем шаге, будут обновлены на следующем шаге, отражая тем самым факт обработки следующего символа со своим контекстом.

В заключение отметим, что множество методов арифметического кодирования защищены патентами США (могут быть дополнительно защищены и юрисдикцией других государств). Из-за наличия таких патентов и возможного судебного разбирательства в случае их нарушения большинство реализаций стандарта сжатия JPEG, допускающего варианты как кодирования Хаффмана, так и арифметического кодирования, реально поддерживают лишь только кодирование Хаффмана.

8.2.4. LZW-кодирование

Технологии, охваченные предыдущими разделами, ориентированы на устранение кодовой избыточности. В данном разделе будет рассмотрен подход к сжатию без потерь, который также нацелен на устранение пространственной

избыточности на изображении. Метод, называемый методом кодирования Лемпеля — Зива — Уэлша (Lempel — Ziv — Welch, LZW), отображает последовательности символов источника различной длины на равномерный код, причем не требует априорного знания вероятностей появления кодируемых символов. Вспомним из раздела 8.1.4, что в доказательстве своей первой теоремы Шеннон использовал идею кодирования последовательности исходных символов в противовес кодированию отдельных символов. Ключевой особенностью LZW-сжатия является то, что оно не требует предварительного знания вероятностей появления кодируемых символов. Несмотря на тот факт, что до последнего времени метод LZW-сжатия был защищен патентами США, он интегрирован во многие широко используемые файловые форматы изображений, включая GIF, TIFF и PDF. Формат PNG был разработан с целью обойти лицензионные требования LZW.

Согласно таблицам 8.3 и 8.4 кодирование LZW используется в форматах:

- GIF,
- TIFF,
- PDF,

но ни в одном из стандартов сжатия, одобренных соответствующими международными организациями.

Пример 8.6. LZW-кодирование изображения на рис. 8.9(*a*).

■ Вернемся опять к изображению на рис. 8.9(*a*), имеющему размеры 512×512×8 бит. Запись этого изображения на диск при помощи Adobe Photoshop в несжатом TIFF-формате занимает 286740 байтов — 264144 байта для самого изображения (512×512, 8 бит) плюс 24596 байтов служебной информации. Используя вариант LZW-сжатия TIFF-формата, результирующий файл будет занимать 224420 байтов. Коэффициент сжатия $C = 1{,}28$. Напомним, что при использовании кодирования Хаффмана (пример 8.4) коэффициент сжатия того же изображения составлял $C = 1{,}077$. Дополнительное сжатие, достигаемое при использовании LZW подхода, возникает благодаря удалению части пространственной избыточности изображения. ■

Концептуально LZW-кодирование является очень простым[11] [Welch, 1984]. При запуске процесса кодирования строится начало кодовой книги или *словарь*,

[11] Метод кодирования LZW базируется на том, что кодовая книга формируется в процессе кодирования последовательности символов источника (т. е. пикселей) и является индивидуальной для каждого сеанса. Каждое слово кодовой книги отображает серию из одного или нескольких символов источника. Формирование кода и его запись в кодовую последовательность происходят лишь тогда, когда при поступлении очередного символа источника образуется новая серия, не присутствующая в кодовой книге, а значит, не отображаемая ни одним из имеющихся кодовых слов. При этом в кодовую последовательность записывается код предыдущей, ранее уже известной серии и осуществляется пополнение кодовой книги новой серией. Затем кодер начинает анализировать следующую последовательность символов, начиная с того, на котором оборвалась предыдущая серия. Перед началом кодирования строится короткая тривиальная и заранее известная кодовая книга, а алгоритм ее пополнения дает возможность и кодеру, и декодеру строить одинаковые кодовые книги и передавать по каналу лишь уже известные к настоящему моменту кодовые слова. — *Прим. перев.*

содержащий лишь кодируемые символы источника. Для 8-битового монохромного изображения словарь имеет размеры в 256 слов и отображает значения яркостей 0, 1, 2,..., 255. Кодер последовательно анализирует символы источника (т.е. значения пикселей), и при появлении отсутствующей в словаре серии она помещается в определяемую алгоритмом (следующую свободную) позицию словаря. Если первые два пикселя изображения, например, были белыми (255-255), эта серия может быть приписана позиции 256, являющейся следующей свободной после зарезервированных для уровней яркостей позиций с 0 по 255. В следующий раз, когда встретится серия из двух белых пикселей, для их представления будет использовано кодовое слово 256 как адрес позиции, содержащей серию 255-255. В случае 9-битового словаря, содержащего 512 кодовых слов, исходные $8+8=16$ битов, требуемые для представления двух пикселей, будут заменены одним 9-битовым кодовым словом. Ясно, что допустимый размер словаря является важнейшим параметром. Если он слишком мал, то обнаружение совпадающих серий яркостей будет маловероятно; если слишком велик, то размер кодового слова будет ухудшать характеристики сжатия[12].

Пример 8.7. Пример LZW-кодирования.

■ Рассмотрим следующее 8-битовое изображение размерами 4×4, содержащее вертикальный контур:

39	39	126	126
39	39	126	126
39	39	126	126
39	39	126	126

В табл. 8.7 описываются шаги, используемые при кодировании его 16 пикселей. Подготавливается словарь на 512 кодовых слов

Позиция в словаре	Запись словаря
0	0
1	1
⋮	⋮
255	255
256	—
⋮	⋮
511	—

В начальный момент позиции с 256 по 511 еще не используются.

При кодировании пиксели изображения обрабатываются слева направо и сверху вниз. Осуществляется конкатенация (присоединение) каждого следующего значения яркости с имеющейся на данный момент серией, называемой «распознанная серия», которая приведена в позиции 1 табл. 8.7. Как можно

[12] В практических реализациях алгоритма LZW размер кодового слова изменяется (возрастает) в процессе кодирования. — *Прим. перев.*

Таблица 8.7. Пример LZW-кодирования

Распознанная серия	Обрабатываемый пиксель	Выход кодера	Позиция кодового слова в словаре	Запись словаря
	39			
39	39	39	256	39-39
39	126	39	257	39-126
126	126	126	258	126-126
126	39	126	259	126-39
39	39			
39-39	126	256	260	39-39-126
126	126			
126-126	39	258	261	126-126-39
39	39			
39-39	126			
39-39-126	126	260	262	39-39-126-126
126	39			
126-39	39	259	263	126-39-39
39	126			
39-126	126	257	264	39-126-126
126		126		

увидеть, вначале эта переменная обнулена или пуста. Словарь просматривается на предмет совпадения с каждой очередной серией, и если таковая обнаруживается, как происходит в первой строке таблицы, то серия заменяется кодом (номером позиции) совпадающей и распознанной (т. е. имеющейся в словаре) серии, что отмечено в первой колонке второй строки. При этом еще не порождается никакого кода и не происходит обновления словаря. Если же совпадения серии и словаря не обнаруживается (как происходит во второй строке таблицы), то номер позиции распознанной к настоящему моменту серии (39) подается на выход в качестве очередного кода; текущая нераспознанная серия пополняет словарь, а состояние распознанной серии инициируется последним поступившим символом. Последние две колонки таблицы описывают коды и серии яркостей, которые последовательно добавляются к словарю при кодировании всего изображения размерами 4×4 элемента. Добавляются девять дополнительных кодовых слов. По завершении кодирования словарь содержит 265 кодовых слов; при этом LZW-алгоритм успешно обнаружил несколько повторяющихся серий яркостей, что позволило ему сократить исходное 128-битовое изображение до 90-битового изображения (т. е. до 10 кодов из 9 битов). Кодовая последовательность на выходе образуется при чтении третьей колонки («Выход кодера») сверху вниз. Результирующий коэффициент сжатия равен 1,42:1. ■

Уникальным качеством только что продемонстрированного LZW-кодирования является то, что кодовая книга (словарь) создается в процессе

кодирования данных. Примечательно, что LZW-декодер строит идентичный словарь восстановления, если он декодирует поток данных синхронно с кодером. Пользователю предлагается в качестве упражнения (см. задачу 8.20) декодировать выход, полученный в предыдущем примере, и восстановить кодовую книгу. Хотя в данном примере это не требуется, большинство практических приложений предусматривают стратегию действий при переполнении словаря. Простым решением является очистка, или инициализация, словаря при его заполнении и затем продолжение кодирования с чистым словарем. Более сложным вариантом может быть слежение за характеристиками сжатия и очистка словаря, если он становится неудовлетворительным или недопустимо большим. В качестве альтернативы можно прослеживать и удалять наименее часто используемые входы словаря, заменяя их по мере необходимости.

8.2.5. Кодирование длин серий

Как уже отмечалось в разделе 8.1.2, изображения с повторяющимися значениями яркостей вдоль строк (или столбцов) часто удается сжать путем представления последовательностей одинаковых значений в виде *пар — длин серий*, где одно число в паре означает яркость новой серии, а второе — число повторяющихся пикселей данной яркости. Данный метод, известный как *кодирование длин серий* (run-length encoding, RLE), был разработан в 1950-х годах и наряду со своим двумерным расширением оставался стандартным подходом к сжатию при кодировании факсимильных (FAX) данных. Сжатие достигается за счет устранения простой формы пространственной избыточности — групп пикселей одинаковой яркости. Если же таких групп нет или слишком мало, то кодирование длин серий приводит к расширению данных.

Согласно таблицам 8.3 и 8.4 кодирование длин серий используется в следующих стандартах сжатия и файловых форматах[13]:

- CCITT,
- JBIG2,
- JPEG,
- M-JPEG,
- MPEG-1,2,4,
- BMP,

а также в некоторых других.

Пример 8.8. Кодирование длин серий в формате файлов BMP.

■ Формат файлов BMP использует форму кодирования длин серий, в которой данные изображения могут быть представлены в двух различных режимах: кодированном и абсолютном, причем каждый из них может появляться где угодно на изображении. В кодированном режиме используется двухбайтовое RLE-представление. Первый байт указывает число последовательных пиксе-

[13] Кодирование длин серий используется также форматами TIFF и PCX; последний был весьма распространен лет 20 назад, но в настоящее время мало применяется и, видимо, по этой причине не упомянут в книге. — *Прим. перев.*

лей в серии, имеющих одинаковый цвет, индекс которого содержится во втором байте. 8-битовый индекс цвета задает яркость (или цвет) серии из таблицы в 256 возможных яркостей (цветов).

В *абсолютном* режиме первый байт равен 0, а второй задает один из четырех возможных случаев, как это показано в табл. 8.8. Если второй байт равен 0 или 1, то это означает, что достигнут конец строки либо конец изображения. Если он равен 2, то следующие 2 байта задают беззнаковые горизонтальное и вертикальное смещения к новой пространственной точке (пикселю) изображения. Если значение второго байта находится в диапазоне от 3 до 255, то он задает число последующих несжатых индексных значений пикселей — каждый следующий байт содержит индекс одного пикселя. Общее число байтов должно быть выровнено по границе 16-битовых слов.

Несжатый BMP-файл изображения на рис. 8.9(*а*) размерами 512×512×8 бит (записанный при помощи Photoshop) занимает 263244 байта. Сжатый при помощи RLE-варианта BMP файл увеличивается до 267706 байтов, т. е. коэффициент сжатия $C = 0{,}98$. В изображении недостаточно много серий одинаковой яркости, чтобы кодирование длин серий было эффективным; наблюдается небольшое увеличение объема файла. Однако изображение на рис. 8.1(*в*) RLE-вариант BMP сжимает с коэффициентом сжатия $C = 1{,}35$. ■

Несжатый BMP-файл короче несжатого TIFF-файла в примере 8.7 из-за разницы в длине служебной информации.

Кодирование длин серий особенно эффективно при сжатии двоичных изображений. Поскольку имеются лишь две возможные яркости (черная и белая), то весьма вероятно, что соседние пиксели будут одинаковыми. Кроме того, каждая строка изображения может быть представлена последовательностью одних только длин, а не пар длина—яркость, как было в примере 8.8. Основная идея заключается в том, чтобы последовательную группу (серию) из нулей или единиц в строке кодировать значением ее длины, задав *при этом* правило определения значения яркости серии. Наиболее часты два правила: (1) задавать значение первой серии каждой строки или (2) декларировать, что каждая строка начинается с белой серии, но при этом ее длина может быть равна 0.

Хотя кодирование длин серий само по себе является весьма эффективным способом сжатия двоичных изображений, обычно можно дополнительно повысить степень сжатия путем неравномерного кодирования самих значений

Таблица 8.8. Коды абсолютного режима BMP-формата. В этом режиме первый байт BMP-пары равен 0

Значение второго байта	Событие
0	Конец строки
1	Конец изображения
2	Переход к новой точке
3—255	Индивидуальное задание значений

длин серий. К тому же длины черных и белых серий могут кодироваться по отдельности, используя разные неравномерные коды, каждый их которых оптимизирован по своей статистике. Например, допуская, что символ a_j представляет черную серию длины j, можно оценить вероятность того, что символ a_j может быть порожден гипотетическим источником длин черных серий путем деления числа черных серий длины j изображения на общее число черных серий. Оценка энтропии этого источника длин черных серий, обозначаемая H_0, получается подстановкой этих вероятностей в (8.1-6). Аналогичным образом можно подсчитать энтропию источника длин белых серий, обозначаемую H_1. Приближенное значение общей энтропии изображения, кодированного длинами серий, составит

$$H_{\mathrm{RL}} = \frac{H_0 + H_1}{L_0 + L_1}, \tag{8.2-7}$$

где L_0 и L_1 означают средние значения длин черных и белых серий. Формула (8.2-7) дает оценку среднего числа битов на пиксель, требуемых для сжатия двоичного изображения посредством кодирования длин серий.

Двумя из наиболее старых и широко используемых стандартов кодирования являются стандарты MKKTT группы 3 и группы 4 для сжатия бинарных (двухградационных) изображений. Хотя они используются во множестве компьютерных приложений, изначально они разрабатывались как методы факсимильного (FAX) кодирования документов для передачи по телефонным линиям. В стандарте группы 3 применена технология кодирования длин серий, согласно которой в каждой группе из K строк (для $K = 2$ или 4) все строки, кроме первой по выбору могут кодироваться двумерным способом. Стандарт группы 4 является упрощенным или модифицированным вариантом стандарта группы 3, в котором допускается лишь двумерное кодирование. В обоих стандартах использована одна и та же схема двумерного кодирования, которая является двумерной лишь в том смысле, что при кодировании текущей строки используется информация из предыдущей строки. И одномерный, и двумерный варианты кодирования рассмотрены ниже.

Одномерное MKKTT-сжатие

В одномерном стандарте сжатия MKKTT группы 3 каждая строка изображения[14] кодируется в виде последовательности слов неравномерного кода Хаффмана, которые представляют длины чередующихся белых и черных серий при сканировании строки слева направо. Используемый метод сжатия обычно называют *модифицированным кодом Хаффмана* (modified Huffman, MH). В нем применяются кодовые слова двух типов: *коды завершения* и *составные коды*. Если длина серии r меньше 63, для ее представления используется код завершения из табл. А-1 в приложении А. Отметим, что для белых и черных серий стандарт предусматривает различные коды завершения. Если $r > 63$, используется два кода: составной код для кодирования частного $\lfloor r/64 \rfloor$ и код завершения для остатка $r \bmod 64$. Составные коды приведены в табл. А-2; среди них есть как зависящие от яркости (черное или белое), так и не зависящие. Для $\lfloor r/64 \rfloor < 1792$

[14] Согласно стандарту изображения называются *страницами*, а последовательность изображений — *документом*.

используются различные коды черных и белых серий, в противном случае применяются одинаковые коды. Стандарт требует, чтобы каждая строка начиналась с кодового слова длины белой серии, которая может иметь нулевую длину — тогда ее код будет 00110101. Наконец, в окончании строки, равно как и в начале первой строки каждого нового изображения, используется уникальное кодовое слово 000000000001, означающее конец строки (EOL). Сигналом конца последовательности изображений являются шесть последовательных слов EOL.

Как и в разделе 8.2.2, нотация $\lfloor x \rfloor$ означает наибольшее целое, которое меньше или равно x.

Двумерное MKKTT-сжатие

В двумерном способе сжатия, используемом стандартами MKKTT групп 3 и 4, применен метод последовательного кодирования строк, в котором позиция каждого перехода (изменения) яркости (черной на белую или наоборот) кодируется по отношению к позиции *опорного элемента* a_0, расположенного в текущей *строке кодирования*. Предыдущая закодированная строка называется *опорной строкой*; для первой строки каждого нового изображения опорной строкой является воображаемая белая строка. Применяемый способ двумерного кодирования называется кодированием *указателя относительного адреса элемента* (relative element address designate, READ). В стандарте группы 3 допускается наличие одной или трех кодированных способом READ строк между кодированными последовательным (MH) образом строками; такой способ называется *модифицированное READ* (modified READ, MR) кодирование. В стандарте Группы 4 возможно включение большого числа READ строк, и такой способ называют *дважды модифицированным READ* (modified modified READ, MMR) кодированием. Как уже отмечалось ранее, кодирование является двумерным лишь в том смысле, что при кодировании текущей строки используется информация из предыдущей строки. Никакие двумерные преобразования при этом не используются.

На рис. 8.14 представлена блок-схема двумерного процесса кодирования одной строки; повторим, что такая строка называется *строкой кодирования*, а предыдущая (уже закодированная ранее) — *опорной строкой*. Начальные шаги процедуры направлены на локализацию (определение положения) нескольких ключевых *переходных элементов*: a_0, a_1, a_2, b_1 и b_2. Переходной элемент определяется стандартом как пиксель, значение которого отличается от значения предшествующего пикселя в той же строке. Важнейшим переходным элементом является a_0 (опорный элемент), который либо устанавливается в позицию воображаемого белого переходного элемента слева от первого пикселя каждой новой строки кодирования, либо задается из предшествующего режима кодирования. (Режимы кодирования рассматриваются в следующем абзаце.) После того, как a_0 локализован, a_1 отыскивается как первый переходной элемент справа от a_0 в строке кодирования, a_2 — как следующий переходной элемент справа от a_0, (т. е. первый справа от a_1). Элемент b_1 — переходной элемент противоположного (по отношению к a_0) значения в опорной (предыдущей) строке, расположенный

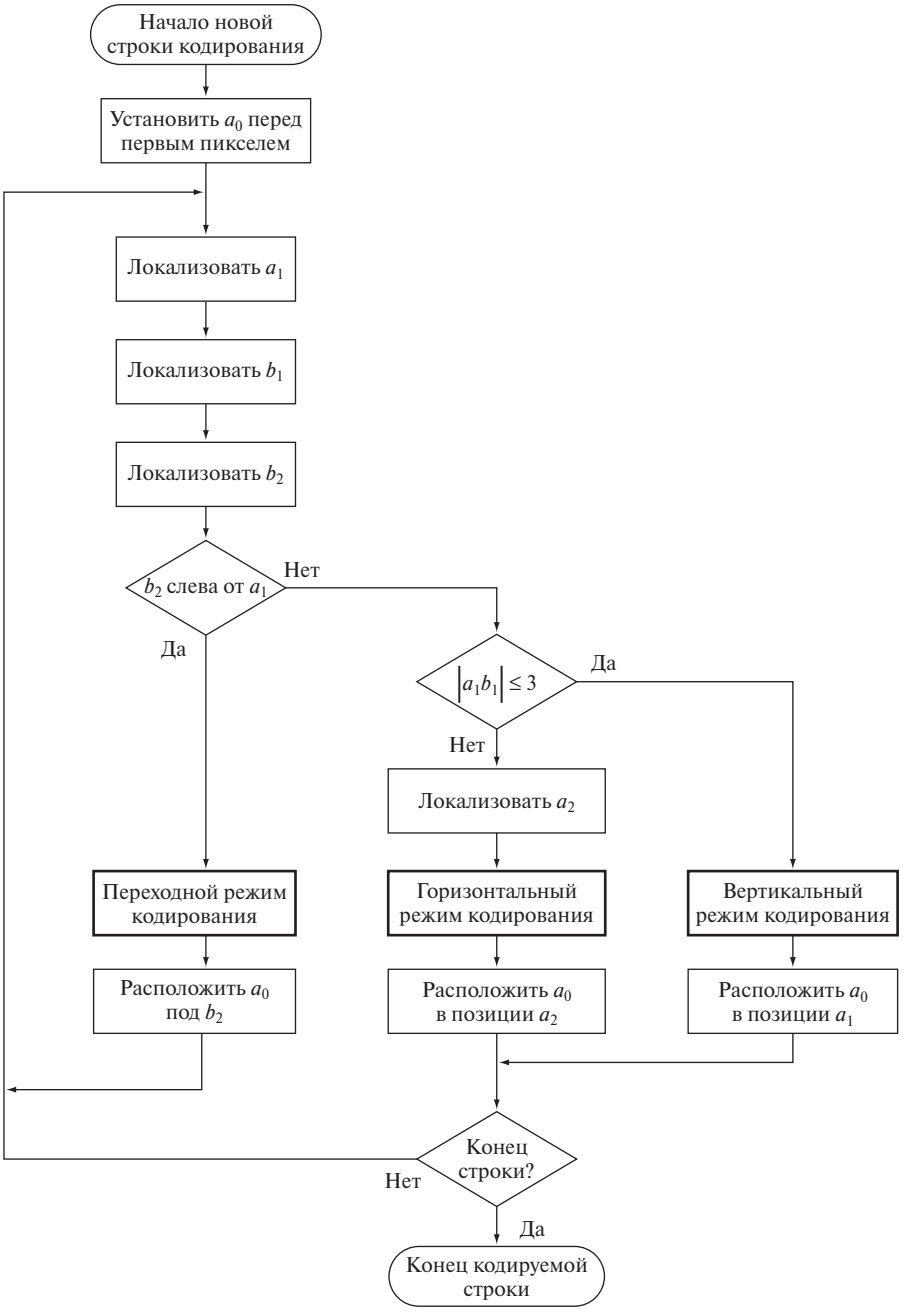


Рис. 8.14. Двумерная READ-процедура кодирования MKKTT. Нотация $|a_1\ b_1|$ означает абсолютную величину расстояния между переходными элементами a_1 и b_1

а
б

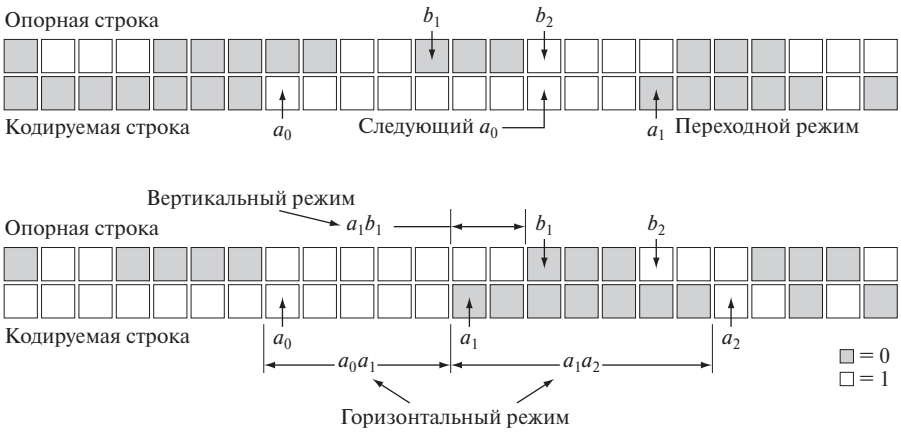


Рис. 8.15. Параметры процедуры двумерного кодирования MKKTT: (*а*) переходной режим кодирования, (*б*) горизонтальный и вертикальный режимы кодирования

правее a_0, b_2 — следующий переходной элемент справа от b_1 в опорной строке. Если какой-то из этих переходных элементов не находится, то он локализуется в позиции воображаемого пикселя справа от последнего пикселя соответствующей строки. На рис. 8.15 приведены две иллюстрации взаимосвязей между различными переходными элементами.

После локализации текущего опорного элемента и ассоциированных с ним переходных элементов выполняются две проверки условий, на основании которых выбирается один из трех возможных режимов кодирования: переходной режим, вертикальный режим или горизонтальный режим. Начальная проверка, соответствующая первой точке ветвления на блок-схеме рис. 8.14, заключается в сравнении координат b_2 и a_1. Вторая проверка, соответствующая второй точке ветвления на блок-схеме, определяет расстояние (в пикселях) между координатами a_1 и b_1 и сравнивает полученную величину с поро-

Таблица 8.9. Таблица двумерных кодов MKKTT

Режим кодирования	Кодовое слово
Переходной	0001
Горизонтальный	$001 + M(a_0a_1) + M(a_1a_2)$
Вертикальный	
a_1 под b_1	1
a_1 на 1 элемент справа от b_1	011
a_1 на 2 элемента справа от b_1	000011
a_1 на 3 элемента справа от b_1	0000011
a_1 на 1 элемент слева от b_1	010
a_1 на 2 элемента слева от b_1	000010
a_1 на 3 элемента слева от b_1	0000010
Расширение	0000001×××

говым значением 3. В зависимости от результатов этих проверок управление передается на вход одного из трех выделенных блоков режимов кодирования на рис. 8.14 (переходной, вертикальный или горизонтальный) и выполняется соответствующая процедура кодирования. После чего, согласно блок-схеме рис. 8.14, определяется следующий опорный элемент в подготовке очередной итерации кодирования.

В табл. 8.9 приведены коды, используемые в каждом из трех возможных режимов кодирования. В переходном режиме, из которого исключен случай, когда b_2 расположен непосредственно над a_1, требуется лишь кодовое слово 0001. Как видно на рис. 8.15(*а*), в этом режиме идентифицируются белые или черные серии опорной строки, которые не перекрывают текущую белую или черную серию в строке кодирования. В горизонтальном режиме кодирования расстояния от a_0 до a_1 и от a_1 до a_2 кодируются согласно кодам завершения и составным кодам из табл. A-1 и A-2 приложения А, после чего выполняется конкатенация полученных кодов к коду горизонтального режима 001. В табл. 8.9 это обозначено нотацией 001 + $M(a_0a_1)$ + $M(a_1a_2)$, где a_0a_1 и a_1a_2 означают расстояния от a_0 до a_1 и от a_1 до a_2 соответственно. Наконец, в вертикальном режиме кодирования расстоянию между a_1 и b_1 присваивается один их шести возможных неравномерных кодов. Рис. 8.15(*б*) иллюстрирует параметры, использующиеся как в горизонтальном, так и вертикальном режимах кодирования. Код режима расширения внизу табл. 8.9 используется для указания дополнительного факсимильного режима кодирования. Так, например, код 0000001111 используется для начала несжатого режима передачи.

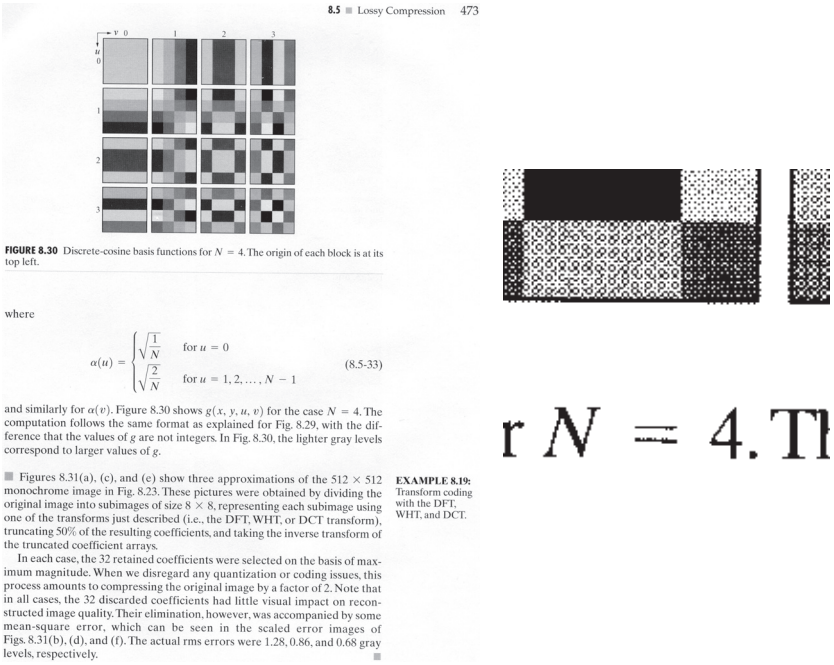
Пример 8.9. Пример вертикального режима кодирования MKKTT.

■ Хотя рис. 8.15(*б*) снабжен параметрами и для горизонтального, и для вертикального режимов кодирования (для облегчения вышеизложенного рассмотрения), расположение черных и белых пикселей на рисунке является случаем вертикального режима кодирования. Т. е. поскольку b_2 расположен правее a_1, первая проверка (на переходной режим) на рис. 8.14 дает отрицательный результат. Вторая проверка, осуществляющая выбор между вертикальным и горизонтальным режимами кодирования, показывает, что должен использоваться вертикальный режим, поскольку расстояние между a_1 и b_1 меньше 3. В соответствии с табл. 8.9 получим в результате кодовое слово 000010, означающее, что a_1 находится на два пикселя левее b_1. При подготовке к следующей итерации кодирования a_0 помещается в позицию a_1. ■

Пример 8.10. Пример кодирования MKKTT.

■ Изображение на рис. 8.16(*а*) представляет собой результат сканирования с разрешением 300 dpi страницы книги размерами 18×24 см, изображенной в масштабе приблизительно 1:3. Отметим, что около половины страницы занимает текст, около 9 % отведено под полутоновое изображение, а остаток страницы — белый фон. Увеличенный фрагмент страницы показан на рис. 8.16(*б*). Не забудем, что мы имеем дело с двухградационным изображением; иллюзия серых тонов создается благодаря описанному в разделе 4.5.4 способу отображения полутонов при печати. Если значения бинарных пикселей изображения упаковывать

а *б*



8.5 ■ Lossy Compression 473

FIGURE 8.30 Discrete-cosine basis functions for $N = 4$. The origin of each block is at its top left.

where

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} & \text{for } u = 0 \\ \sqrt{\frac{2}{N}} & \text{for } u = 1, 2, \ldots, N-1 \end{cases} \qquad (8.5\text{-}33)$$

and similarly for $\alpha(v)$. Figure 8.30 shows $g(x, y, u, v)$ for the case $N = 4$. The computation follows the same format as explained for Fig. 8.29, with the difference that the values of g are not integers. In Fig. 8.30, the lighter gray levels correspond to larger values of g.

EXAMPLE 8.19: Transform coding with the DFT, WHT, and DCT.

■ Figures 8.31(a), (c), and (e) show three approximations of the 512 × 512 monochrome image in Fig. 8.23. These pictures were obtained by dividing the original image into subimages of size 8 × 8, representing each subimage using one of the transforms just described (i.e., the DFT, WHT, or DCT transform), truncating 50% of the resulting coefficients, and taking the inverse transform of the truncated coefficient arrays.

In each case, the 32 retained coefficients were selected on the basis of maximum magnitude. When we disregard any quantization or coding issues, this process amounts to compressing the original image by a factor of 2. Note that in all cases, the 32 discarded coefficients had little visual impact on reconstructed image quality. Their elimination, however, was accompanied by some mean-square error, which can be seen in the scaled error images of Figs. 8.31(b), (d), and (f). The actual rms errors were 1.28, 0.86, and 0.68 gray levels, respectively. ■

r N = 4. Tl

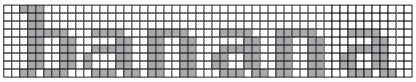
Рис. 8.16. Двухградационный скан страницы книги: (*а*) вид всей страницы; (*б*) увеличенный фрагмент, показывающий двоичные элементы, используемые в растрировании при печати

по 8 пикселей на байт, то сканированное изображение, обычно называемое документом, имеющее размеры 1952×2687 бит, потребует 658068 байт. Несжатый PDF-файл документа, созданный программой Photoshop, занимает 663445 байт. Сжатие MKKTT группы 3 сокращает файл до размера 123497 байт, давая в результате коэффициент сжатия $C = 5{,}37$; использование сжатия MKKTT группы 4 позволяет сократить файл до 110456 байт, увеличивая тем самым коэффициент сжатия почти до 6. ■

Не следует путать PDF здесь, являющееся аббревиатурой Portable Document Format («формат переносимого документа»), с PDF, используемым в предыдущих главах и разделах, означающим probability density function («плотность распределения»).

8.2.6. Кодирование на базе шаблонов

В кодировании на базе *шаблонов* или *индексов* изображение представляется в виде набора часто появляющихся подызображений, называемых *шаблонами*. Каждый такой шаблон запоминается в виде образа в *словаре шаблонов*, а само изображение кодируется как множество триплетов $\{(x_1, y_1, t_1), (x_2, y_2, t_2), \ldots\}$, в которых *индекс* t_i задает шаблон (подызображение), определяя его адрес в словаре, а пара значений (x_i, y_i) — координаты его расположения на изображении. Тем самым, каждый триплет представляет экземпляр из словаря шаблонов на изображении. Запоминая



Индекс	Шаблон
0	
1	
2	

Триплет
(0, 2, 0)
(3, 10, 1)
(3, 18, 2)
(3, 26, 1)
(3, 34, 2)
(3, 42, 1)

а б в

Рис. 8.17. (*а*) Двухградационный документ, (*б*) словарь шаблонов, (*в*) набор триплетов, используемых для локализации шаблонов в документе

повторяющиеся шаблоны лишь однажды, можно значительно сжать изображение — особенно при хранении документов и в поисковых системах, где шаблоны часто являются растровыми образами, повторяющимися по много раз.

> Согласно табл. 8.3 и 8.4 кодирование на базе шаблонов используется в стандарте сжатия JBIG2.

Рассмотрим простое двухградационное изображение на рис. 8.17(*а*). Оно содержит одно слово *banana*, которое состоит из трех отдельных шаблонов: одного *b*, трех *a* и двух *n*. Положим, что *b* является первым шаблоном, идентифицируемым в процессе кодирования, а его образ размерами 9×7 запоминается в позиции 0 в словаре шаблонов. Как видно на рис. 8.17(*б*), индекс, идентифицирующий шаблон *b*, есть 0. Таким образом, первым триплетом в кодированном представлении изображения (см. рис. 8.17(*в*)) будет $(0, 2, 0)$, означая тем самым, что шаблон *b* должен быть помещен в позицию $(0, 2)$ на декодируемом изображении. После того, как образы шаблонов *a* и *n* идентифицированы и добавлены в словарь, остаток изображения может быть закодирован пятью дополнительными триплетами. Сжатие достигается в случае, если шесть триплетов, требуемых для локализации шаблонов, вместе с тремя образами, необходимыми для запоминания используемых шаблонов, будут по объему меньше исходного изображения. В рассматриваемом примере исходное изображение имеет размеры 9×51×1 = 459 бит; полагая, что каждый триплет состоит из 3 байт, сжатое представление будет иметь (6×3×8)+[(9×7)+(6×7)+(6×6)] = 285 бит. Результирующий коэффициент сжатия $C = 1{,}61$. Декодирование шаблонного представления на рис. 8.17(*в*) сводится к чтению шаблонов, указанных в триплетах, и размещению их в позициях изображения, заданных в каждом триплете.

Кодирование на базе шаблонов было предложено в начале 1970-х [Ascher, Nagy, 1974], но стало использоваться на практике лишь недавно. Прогресс в области распознавания символов (см. главу 12) и возросшие скорости компьютерной обработки сделали возможным в реальном времени реализацию как отождествления образов в словаре шаблонов, так и отыскания места локализации их на изображении. Как и во многих других методах сжатия, декодирование на базе шаблонов значительно быстрее, чем кодирование. Наконец отметим, что как шаблоны, размещенные в словаре, так и набор триплетов, описывающий их локализацию, сами могут быть кодированы для дальнейшего улучшения характеристик

сжатия. Если — как на рис. 8.17 — допустимо лишь точное совпадение шаблона с изображением, сжатие в результате будет без потерь; если же допустить наличие некоторых различий, возникнет некоторый уровень ошибок восстановления.

Сжатие JBIG2

JBIG2 является международным стандартом сжатия двухградационных изображений. Изображение сегментируется на пересекающиеся и/или непересекающиеся области, содержащие контент одного из следующих типов: текстовый, полутоновой или обобщенный. Для каждого из них используются специально оптимизированные методы сжатия.

- Текстовые области состоят из символов и идеально подходят для применения методов кодирования на базе шаблонов. Как правило, каждый текстовый символ будет соответствовать двухградационному образу символа — подызображению, представляющему некоторый отдельный знак текста. Обычно в словаре шаблонов для каждого символа используемого шрифта на каждом из регистров (верхнем или нижнем) используется только один двухградационный образ. Так, должен быть один образ для а, один для А, один для б, и так далее.

  В сжатии JBIG2 *с потерями*, часто называемом *без заметных* (или *видимых*) *потерь*, пренебрегают незначительными расхождениями между образами в словаре шаблонов (т. е. образцами или эталонными растровыми представлениями символов) и конкретным видом соответствующих символов на изображении. В случае сжатия *без потерь* возникающие разности связываются с конкретным триплетом, кодирующим данный символ, и запоминаются; при восстановлении разность добавляется декодером, формируя точный вид изображения. Все растровые представления в словаре шаблонов сжимаются либо при помощи арифметического кодирования, либо MMR-кодированием (см. раздел 8.2.5); набор сформированных триплетов также кодируется либо арифметическим кодированием, либо кодом Хаффмана.
- *Полутоновые области* подобны текстовым областям в том, что состоят из узоров, расположенных на регулярной решетке. Однако шаблоны, которые сохраняются в словаре, являются не образами знаков, а периодическими узорами, представляющими яркости (фотографического изображения), растрированные при формировании двухградационного изображения для печати.
- *Обобщенные области* включают детали, не относящиеся к тексту или полутонам, такие как линии рисунков и шум; они сжимаются либо при помощи арифметического кодирования, либо MMR-кодированием.

Как и многие другие стандарты сжатия, JBIG2 определяет поведение декодера. Он не задает стандарт кодера исключительным образом и является достаточно гибким, допуская различные конструкции кодера. Хотя исполнение кодера остается неуточненным, тем не менее оно важно, поскольку именно им определяется достигаемая степень сжатия. Ведь кодер должен сегментировать изображение на области, выбирать текстовые и полутоновые шаблоны, которые сохранены в словарях, решать, в каких случаях шаблоны по существу совпадают либо отличаются от потенциальных образцов символов на изображении.

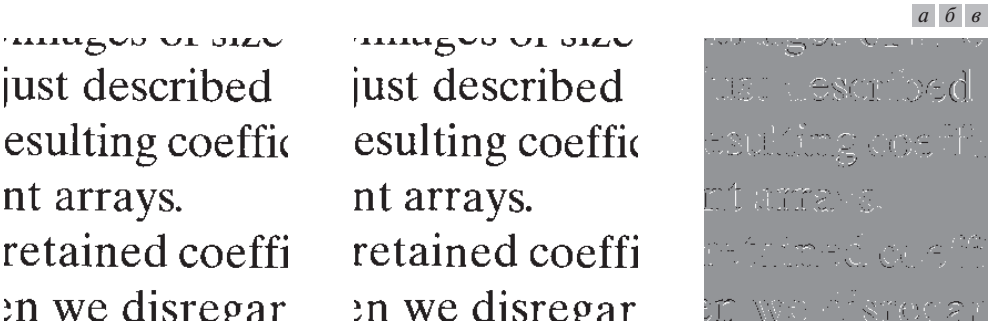


Рис. 8.18. Сравнение JBIG2 сжатия: (*а*) сжатие без потерь и результат восстановления; (*б*) без ощущаемых потерь; (*в*) разница между двумя вариантами

Декодер же попросту использует эту информацию для воссоздания исходного изображения.

Пример 8.11. Пример сжатия JBIG2.

■ Опять рассмотрим двухградационное изображение на рис. 8.16(*а*). На рис. 8.18(*а*) показан восстановленный участок изображения после JBIG2 сжатия без потерь (при помощи коммерческого приложения для сжатия документов); это точное повторение исходного изображения. Обратим внимание, что буквы d на восстановленном тексте слегка отличаются, несмотря на тот факт, что они были сгенерированы из одного и того же шаблона d в словаре. Взятый из словаря образец уточнялся путем использования отличий образа d в словаре и символов d на изображении. Стандартом определен алгоритм для выполнения этой операции во время декодирования закодированного в словаре образа шаблона. При обсуждении допустимо считать, что к образу шаблона, взятому из словаря, прибавляется запомненная разность между видом конкретного символа на изображении и шаблоном в словаре.

На рис. 8.18(*б*) представлен другой вариант реконструкции области (*а*) — после сжатия JBIG2 без заметных потерь. Заметим, что теперь d на изображении стали идентичными — они скопированы непосредственно из словаря шаблонов. Восстановление называется «без заметных потерь», поскольку текст читаем и даже шрифт тот же самый; небольшие различия — едва заметные на рис. 8.18(*в*) между d на исходном изображении и в словаре — считаются несущественными, поскольку они не влияют на читабельность. Напоминаем, что мы имеем дело с двухградационными изображениями, поэтому на рис. 8.18(*в*) присутствуют лишь три значения яркостей. Значением 128 отмечены области, где нет отличий между изображениями рис. 8.18(*а*) и (*б*); значения 0 (черное) и 255 (белое) индицируют наличие на двух изображениях пикселей противоположных яркостей — черного на одном и белого на другом либо наоборот.

Сжатие JBIG2 без потерь, используемое для формирования изображения рис. 8.18(*а*), сократило исходный несжатый PDF-файл изображения, имевший размер 663445 байт, до размера 32705 байт; тем самым коэффициент сжатия $C = 20{,}3$. Использование сжатия JBIG2 без заметных потерь позволило еще уменьшить изображение до 23913 байт, подняв коэффициент сжатия примерно

до 27,7. Как видно, удалось достичь в 4 и 5 раз более высокой степени сжатия, чем стандартом МККТТ групп 3 и 4 в примере 8.10. ■

8.2.7. Кодирование битовых плоскостей

Методы кодирования длин серий, а также кодирования на базе шаблонов, рассмотренные в предыдущих разделах, могут применяться и к изображениям большей глубины, чем двухградационные, путем раздельной обработки битовых плоскостей изображения. Такой метод, называемый *кодирование битовых плоскостей*, основан на идее предварительного разложения многоградационного изображения (черно-белого или цветного) на серию двоичных изображений (см. раздел 3.2.4) и последующего кодирования каждого из них при помощи одного или нескольких хорошо известных алгоритмов сжатия двоичных изображений. Ниже рассматриваются два наиболее известных подхода к разложению.

Согласно табл. 8.3 и 8.4 кодирование битовых плоскостей используется в стандартах сжатия и файловых форматах:

- JBIG1,
- JPEG-2000.

Уровни яркости m-битового монохромного изображения могут быть представлены в форме полинома с основанием 2:

$$a_{m-1}2^{m-1} + a_{m-2}2^{m-2} + \cdots + a_1 2^1 + a_0 2^0 . \qquad (8.2\text{-}8)$$

Основанный на этом свойстве простой метод разложения многоградационного изображения на множество двоичных изображений заключается в разделении m коэффициентов полинома на m однобитовых плоскостей. Как уже отмечалось в разделе 3.2.4, плоскость низшего порядка (т. е. плоскость, соответствующая наименее значащему биту) образуется выделением битов (или коэффициентов) a_0 каждого элемента, а битовая плоскость самого высокого порядка — выделением битов a_{m-1}. Вообще каждая битовая плоскость формируется установкой значений ее элементов равным значениям соответствующих битов или полиномиальных коэффициентов элементов исходного изображения. Недостаток, присущий данному разложению, состоит в том, что малые изменения яркостей могут существенно влиять на сложность битовых плоскостей. Так, если пиксель со значением 127 (01111111_2) изменит значение на 128 (10000000_2), то во всех битовых плоскостях произойдет переход с 1 на 0 (или с 0 на 1). Например, поскольку старшие биты двух двоичных кодов для 127 и 128 различаются, то и пиксель старшей битовой плоскости, имевший первоначальное значение 0, изменится на 1.

Альтернативным подходом к разложению, который уменьшает эффект переноса битов при малых изменениях яркостей, является представление изображения в виде m-битового *кода Грея*. Соответствующий код Грея, записываемый в виде $g_{m-1}...g_1g_0$, может быть вычислен по коэффициентам полинома (8.2-8) следующим образом:

$$g_i = a_i \oplus a_{i+1} \qquad 0 \le i \le m-2,$$
$$g_{m-1} = a_{m-1}. \tag{8.2-9}$$

Здесь знак $\oplus$ означает операцию исключающего ИЛИ. Этот код[15] имеет то уникальное свойство, что идущие друг за другом кодовые слова различаются только в одной битовой позиции. Таким образом, малые изменения яркости будут с меньшей вероятностью воздействовать на все m битовых плоскостей. Например, если происходит переход с уровня яркости 127 на уровень 128, то переход с 0 на 1 возникнет только в битовой плоскости старшего порядка, поскольку коды Грея для 127 и 128 равны 01000000_2 и 11000000_2 соответственно.

Пример 8.12. Кодирование битовых плоскостей.

■ 8-битовое монохромное изображение ребенка на рис. 8.19(*а*) представлено на рис. 8.19 и 8.20 в виде восьми двоичных битовых плоскостей, а также восьми битовых плоскостей кода Грея. Отметим, что старшие битовые плоскости являются значительно менее сложными, чем младшие, то есть они содержат протяженные области с меньшим количеством деталей или случайных изменений (старшие битовые плоскости a_7 и g_7 совпадают). Кроме того, битовые плоскости кода Грея являются менее сложными, чем соответствующие двоичные битовые плоскости. Оба эти наблюдения отражены в табл. 8.10, где приведено сравнение степени сжатия битовых плоскостей и плоскостей кода Грея при помощи JBIG2-кодирования. Отметим, например, что плоскости a_6 и g_6 сжаты значительно сильнее, чем a_5 и g_5, а также что g_5 и g_6 сжимаются сильнее, чем их двоичные аналоги a_5 и a_6. Эта тенденция наблюдается по всей таблице с одним лишь исключением для a_0. Код Грея дает выигрыш в степени сжатия в среднем около 1,06:1. Объединение данных сжатия всех плоскостей вместе показывает,

Таблица 8.10. Результаты JBIG2-сжатия без потерь битовых плоскостей изображения рис. 8.19(*а*), представленного двоичным кодом и кодом Грея. Приведенные размеры файлов включают также служебную информацию PDF-представления

Битовая плоскость (значение *i*)	Двоичный код	Код Грея	Коэффициент сжатия
7	6999	6999	1,00
6	12791	11024	1,16
5	40104	36914	1,09
4	55911	47415	1,18
3	78915	67787	1,16
2	101535	92630	1,10
1	107909	105286	1,03
0	99753	107909	0,92

[15] Обратим внимание читателя, что полученный код Грея $g_{m-1}...g_1g_0$ не может интерпретироваться в виде набора коэффициентов какого-то полинома, и поэтому значения его битов не могут быть подставлены в формулу (8.2-8) вместо коэффициентов $a_{m-1}...a_1a_0$. — *Прим. перев.*

Рис. 8.19. (*а*) 256-битовое монохромное изображение. (*б*)—(*з*) Четыре старших битовых плоскости изображения (*а*): (*б*), (*в*), (*д*), (*ж*) — двоичный код, (*б*), (*г*), (*е*), (*з*) — код Грея

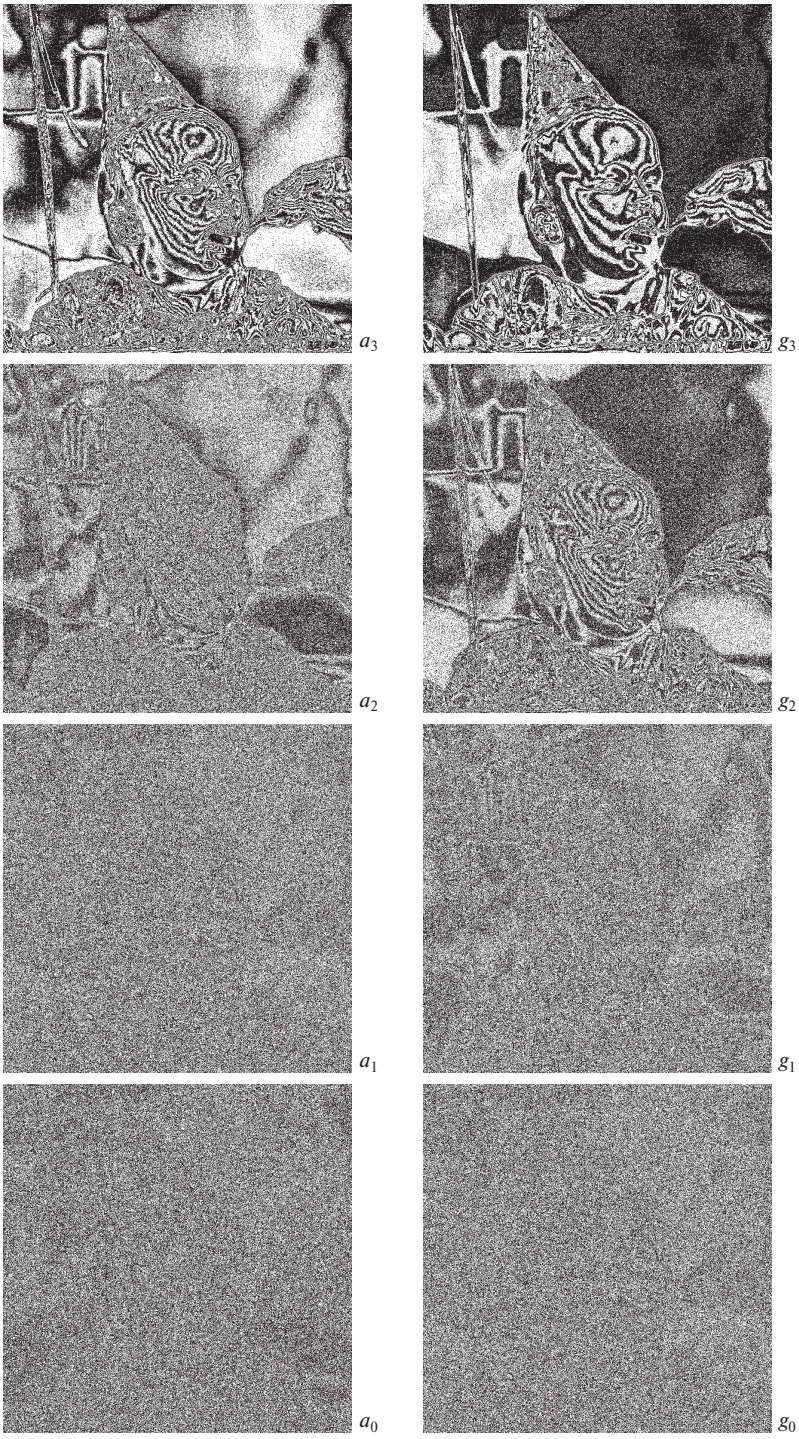


Рис. 8.20. (*а*)—(*з*) Четыре младших битовых плоскости изображения рис. 8.19(*а*): левый столбец — двоичный код, правый столбец — код Грея

что сжатие изображения, записанного кодом Грея, дает коэффициент сжатия 678676/475964 = 1,43; тогда как сжатие изображения в обычном двоичном коде дает 678676/503916 = 1,35.

Заметим в заключение, что в изображениях двух младших битов на рис. 8.20 структура практически неразличима. Поскольку это обычно для 8-битовых монохромных изображений, при кодировании битовых плоскостей часто ограничивают изображения до 6 бит/пиксель или даже меньше. Такое ограничение присутствует, в частности, в стандарте JBIG1 — предшественнике JBIG2. ■

8.2.8. Блочное трансформационное кодирование

В данном разделе будут рассмотрены методы сжатия, основанные на разбиении изображения на небольшие неперекрывающиеся блоки одинаковых размеров (скажем, 8×8) и последующей независимой обработке блоков при помощи двумерных преобразований. В *блочном трансформационном кодировании* обратимое линейное преобразование (например преобразование Фурье) используется для отображения каждого *блока* или *подызображения* в набор коэффициентов преобразования, которые затем квантуются и кодируются. Для большинства реальных изображений значительное число коэффициентов имеют малую величину и могут быть достаточно грубо квантованы (или полностью удалены) ценой небольшого искажения изображения. Для преобразования данных изображения могут использоваться различные преобразования, включая дискретное преобразование Фурье (ДПФ), рассмотренное в главе 4.

Согласно таблицам 8.3 и 8.4 блочное трансформационное кодирование используется в следующих стандартах сжатия и файловых форматах:

- JPEG,
- M-JPEG,
- MPEG-1,2,4,
- H.261, H.262, H.263 и H.264,
- DV и HDV,
- VC-1,

а также в других стандартах сжатия.

На рис. 8.21 схематически показана типичная система блочного трансформационного кодирования. Кодер выполняет четыре достаточно понятные операции: разбиение изображения на блоки, преобразование, квантование

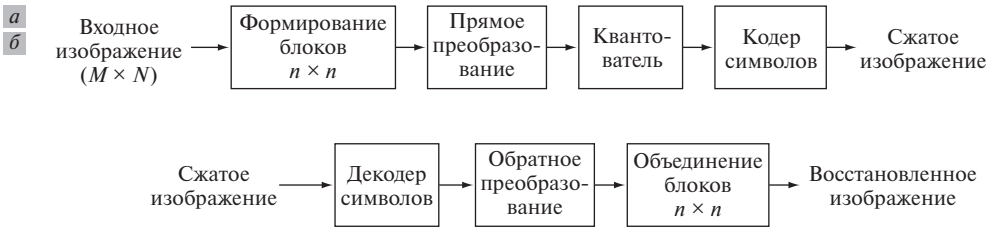


Рис. 8.21. Система блочного трансформационного кодирования: (*а*) кодер; (*б*) декодер

и кодирование. Декодер выполняет обратную последовательность операций (за исключением квантования). Первоначально изображение размерами $M \times N$ разбивается на MN/n^2 блоков размерами $n \times n$, которые затем и подвергаются преобразованиям. Целью процесса преобразования является декорреляция значений элементов в каждом блоке или уплотнение как можно большего количества информации в наименьшее число коэффициентов преобразования. На этапе квантования грубо квантуются или же удаляются те коэффициенты, которые несут малое количество информации в заранее заданном смысле (несколько методов рассматривается ниже в данном разделе). Эти коэффициенты дают наименьший вклад в качество восстанавливаемого блока. На заключительном этапе осуществляется кодирование квантованных коэффициентов, как правило, с помощью неравномерных кодов. Все или некоторые из указанных этапов могут быть адаптированы к содержимому блока, т. е. к локальным характеристикам изображения; такой вариант называют *адаптивным трансформационным кодированием*. В противном случае говорят о *неадаптивном трансформационном кодировании*.

> В настоящем разделе мы ограничим наше внимание рассмотрением блоков квадратной формы (наиболее часто используемый вариант) размерами $n \times n$. При этом предполагается, что M и N кратны n, т. е. изображение расширено, если это необходимо.

Выбор преобразования

Системы блочного трансформационного кодирования, основанные на различных дискретных двумерных преобразованиях, достаточно хорошо исследованы и изучены. Выбор наилучшего преобразования для конкретного приложения зависит от величины допустимой ошибки восстановления и от имеющихся вычислительных ресурсов. Сжатие же возникает не во время преобразования, а на этапе квантования полученных коэффициентов.

Со ссылкой на обсуждение вопроса в разделе 2.6.7 рассмотрим блок (подызображение) $g(x, y)$ размерами $n \times n$, прямое дискретное преобразование $T(u, v)$ которого может быть выражено в следующем общем виде:

> Здесь используется нотация $g(x, y)$, чтобы различать подызображение и исходное изображение $f(x, y)$. Тем самым пределом суммирования становится n, а не M и N.

$$T(u,v) = \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} g(x,y) r(x,y,u,v) \qquad (8.2\text{-}10)$$

для $u, v = 0, 1, 2, \ldots, n-1$. Аналогичным образом изображение $g(x, y)$ может быть получено по заданному $T(u, v)$ при помощи обобщенного обратного дискретного преобразования

$$g(x,y) = \sum_{u=0}^{n-1} \sum_{v=0}^{n-1} T(u,v) s(x,y,u,v) \qquad (8.2\text{-}11)$$

для $x, y = 0, 1, 2, \ldots, n-1$. Функции $r(x, y, u, v)$ и $s(x, y, u, v)$ в данных уравнениях называются соответственно *ядром прямого* и *ядром обратного преобразования*.

По причинам, которые будут ясны ниже, их также называют *базисными функциями* или *базисными изображениями*. Набор $T(u, v)$ для $u, v = 0, 1, 2,..., n - 1$ в уравнении (8.2-10) называют *коэффициентами преобразования*; они могут являться коэффициентами разложения (см. раздел 7.2.1) $g(x, y)$ по базисным функциям $s(x, y, u, v)$.

Как объяснялось в разделе 2.6.7, ядро преобразования в (8.2-10) называется разделимым, если

$$r(x,y,u,v) = r_1(x,u)r_2(y,v). \tag{8.2-12}$$

Кроме того, ядро является симметричным, если r_1 и r_2 — одна и та же функция. В этом случае (8.2-12) может быть записано в виде

$$r(x,y,u,v) = r_1(x,u)r_1(y,v). \tag{8.2-13}$$

Аналогичные комментарии могут быть сделаны по отношению к ядру обратного преобразования; для этого достаточно заменить $g(x, y, u, v)$ на $h(x, y, u, v)$ в (8.2-12) и (8.2-13). Нетрудно показать, что двумерное разделимое преобразование может быть вычислено с помощью соответствующих одномерных преобразований, выполняемых последовательными проходами сначала по строкам, затем по столбцам или же в обратном порядке (см. раздел 4.11.1).

Ядра прямого и обратного преобразований в (8.2-10) и (8.2-11) определяют само преобразование, общую вычислительную сложность, а также ошибки восстановления системы блочного трансформационного кодирования, в которой это преобразование используется. Наиболее известной парой ядер преобразования являются

$$r(x,y,u,v) = e^{-i2\pi(ux+vy)/n} \tag{8.2-14}$$

и

$$s(x,y,u,v) = \frac{1}{n^2} e^{i2\pi(ux+vy)/n}, \tag{8.2-15}$$

где $i = \sqrt{-1}$. Эти функции являются ядрами преобразований, определенных выражениями (2.6-34) и (2.6-35) главы 2 с $M = N = n$. Подставляя эти ядра в (8.2-10) и (8.2-11), получим упрощенный вариант прямого и обратного дискретного преобразования Фурье, введенного в разделе 4.5.5.

Вычислительно более простое преобразование, также широко применяемое в трансформационном кодировании и называемое *преобразованием Уолша — Адамара* (ПУА), получается с помощью функционально идентичных ядер:

$$r(x,y,u,v) = s(x,y,u,v) = \frac{1}{n}(-1)^{\sum_{i=0}^{m-1}\left[b_i(x)p_i(u)+b_i(y)p_i(v)\right]}, \tag{8.2-16}$$

> Чтобы вычислить ПУА по изображению $f(x, y)$ размерами $N \times N$, а не по блоку, следует в (8.2-16) заменить n на N.

где $n = 2^m$. Суммирование в показателе степени выполняется по модулю 2, и $b_k(z)$ означает k-й бит (справа налево) в двоичном представлении z. Если, например, $m = 3$ и $z = 6$ (110_2), то $b_0(z) = 0$, $b_1(z) = 1$ и $b_2(z) = 1$. Значения $p_i(u)$ в (8.2-16) вычисляются следующим образом:

$$
\begin{aligned}
p_0(u) &= b_{m-1}(u),\\
p_1(u) &= b_{m-1}(u) + b_{m-2}(u),\\
p_2(u) &= b_{m-2}(u) + b_{m-3}(u),\\
&\vdots\\
p_{m-1}(u) &= b_1(u) + b_0(u),
\end{aligned}
\qquad (8.2\text{-}17)
$$

где суммирования, как отмечалось ранее, производятся по модулю 2. Для вычисления $p_i(v)$ используются аналогичные выражения.

В отличие от ядер ДПФ, которые являются суммами синусов и косинусов (см. (8.2-14) и (8.2-15)), ядра преобразования Уолша — Адамара состоят из чередующихся +1 и –1, расположенных в шахматном порядке. На рис. 8.22 показано ядро для $n = 4$. Каждый блок состоит из 4×4 = 16 элементов («подквадратов»); белый цвет означает +1, а черный означает –1. Чтобы сформировать левый верхний блок необходимо положить $u = v = 0$ и вычислить значения $r(x, y, 0, 0)$ для $x, y = 0, 1, 2, 3$. Все значения в этом случае равны +1. Второй блок в верхнем ряду есть набор значений $r(x, y, 0, 1)$ для $x, y = 0, 1, 2, 3$ и так далее. Как уже отмечалось, важность преобразования Уолша — Адамара определяется простотой реализации — значения всех элементов в его ядре равны или +1, или –1.

Чтобы вычислить ДКП по изображению $f(x, y)$ размерами $N \times N$, а не по блоку, следует в (8.2-18) и (8.2-19) заменить n на N.

Одним из наиболее часто используемых преобразований для сжатия изображений является *дискретное косинусное преобразование* (ДКП). Оно получается путем подстановки в (8.2-10) и (8.2-11) следующих (одинаковых) ядер:

$$
r(x,y,u,v) = s(x,y,u,v) = \alpha(u)\alpha(v)\cos\left[\frac{(2x+1)u\pi}{2n}\right]\cos\left[\frac{(2y+1)v\pi}{2n}\right], \qquad (8.2\text{-}18)
$$

где

$$
\alpha(u) = \begin{cases} \sqrt{1/n} & \text{для } u = 0 \\ \sqrt{2/n} & \text{для } u = 1, 2, \ldots, n-1 \end{cases} \qquad (8.2\text{-}19)
$$

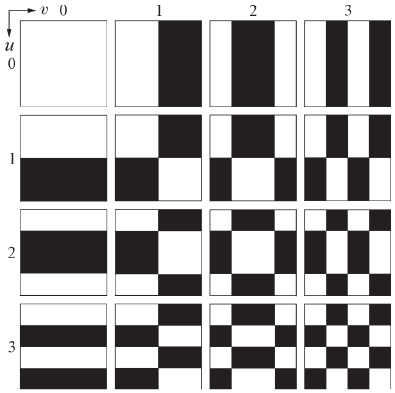


Рис. 8.22. Базисные функции Уолша — Адамара для $n = 4$. Начало координат каждого блока находится в его левом верхнем углу

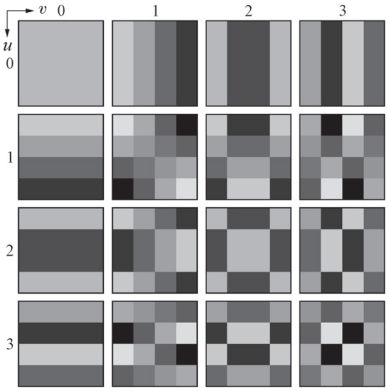


Рис. 8.23. Базисные функции дискретного косинусного преобразования для $n = 4$. Начало координат каждого блока находится в его левом верхнем углу

и аналогично для $\alpha(v)$. На рис. 8.23 показаны базисные функции $r(x, y, u, v)$ для случая $n = 4$. Результаты представлены в том же формате, что и на рис. 8.22, за исключением того, что значения r не являются целыми. Более светлые уровни яркостей на рис. 8.23 соответствуют бóльшим значениям r.

Пример 8.13. Блочное трансформационное кодирование с использованием ДПФ, ПУА и ДКП и усечением коэффициентов.

■ На рис. 8.24(*а*)—(*в*) показаны три приближения полутонового изображения рис. 8.9(*а*) размерами 512×512 элементов. Эти результаты были получены разбиением исходного изображения на блоки размерами 8×8 элементов, представлением каждого блока при помощи одного из рассмотренных преобразований (ДПФ, ПУА или ДКП), обнулением (усечением) 50 % наименьших по значениям коэффициентов и выполнением обратных преобразований над полученными массивами.

Во всех случаях 32 остающихся коэффициента выбирались как наибольшие по значению. Заметим, что 32 удаленных (младших) коэффициента во всех случаях имели весьма малое влияние на качество восстановленных изображений. Их удаление, тем не менее, привело к возникновению некоторых отклонений, которые в виде изображений представлены на рис. 8.24(*г*)—(*е*). Значения стандартных отклонений ошибок составили соответственно 2,32, 1,78 и 1,13 уровней яркости. ■

Небольшие различия в стандартных отклонениях ошибок, приведенных в предыдущем примере, прямо связаны с энергией или характеристиками уплотнения информации примененных преобразований. В соответствии с (8.2-11), изображение $g(x, y)$ размерами $n \times n$ может быть представлено как функция своего двумерного преобразования $T(u, v)$:

$$g(x,y) = \sum_{u=0}^{n-1} \sum_{v=0}^{n-1} T(u,v)s(x,y,u,v) \qquad (8.2\text{-}20)$$

для $x, y = 0, 1, 2, \ldots, n - 1$. Поскольку ядро обратного преобразования $s(x, y, u, v)$ в (8.2-20) зависит только от индексов x, y, u, v, а не от значений $g(x, y)$ или $T(u, v)$, то оно может рассматриваться как набор *базисных функций* или *базисных изо-*

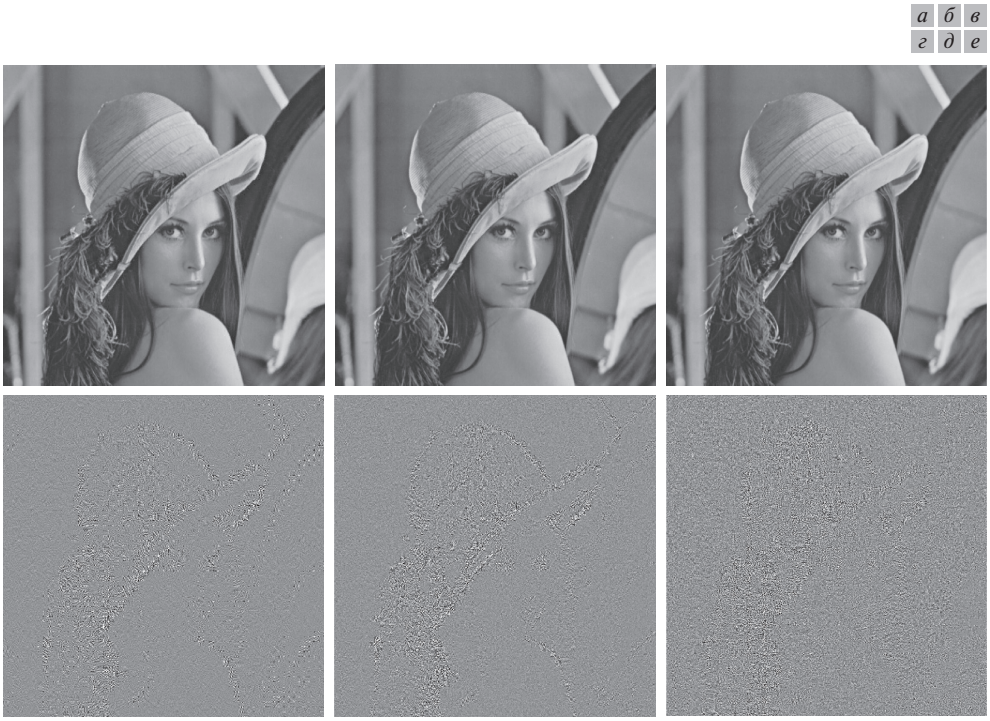


Рис. 8.24. Приближения изображения на рис. 8.9(*а*) при помощи преобразований с усечением коэффициентов: (*а*) Фурье, (*в*) Уолша—Адамара, (*д*) косинусного; (*г*)—(*е*) соответственные изображения ошибок (после градационного преобразования)

бражений для линейной комбинации (8.2-20). Эта интерпретация станет яснее, если переписать (8.2-20) в виде

$$\mathbf{G} = \sum_{u=0}^{n-1}\sum_{v=0}^{n-1} T(u,v)\mathbf{S}_{uv}, \tag{8.2-21}$$

где **G** есть матрица размерами $n \times n$, содержащая значения элементов блока $g(x,y)$,

а

$$\mathbf{S}_{uv} = \begin{bmatrix} s(0,0,u,v) & s(0,1,u,v) & \cdots & s(0,n-1,u,v) \\ s(1,0,u,v) & \vdots & \cdots & \vdots \\ \vdots & \vdots & \cdots & \vdots \\ s(n-1,0,u,v) & s(n-1,1,u,v) & \cdots & s(n-1,n-1,u,v) \end{bmatrix}. \tag{8.2-22}$$

Тогда **G** — матрица, содержащая значения элементов входного блока, явно задается как линейная комбинация n^2 матриц размерами $n \times n$, т.е. матриц $\mathbf{S}_{uv}$ для $u, v = 0, 1, 2, \ldots, n-1$ в (8.2-22). Эти матрицы фактически являются базисными изображениями (или функциями) разложения (8.2-20); соответствующие значения $T(u,v)$ являются коэффициентами разложения. Базисные изображения ПУА и ДКП для случая $n = 4$ иллюстрируются на рис. 8.22 и 8.23.

Зададим *маскирующую функцию* для коэффициентов преобразования:

$$\chi(u,v) = \begin{cases} 0 & \text{если } T(u,v) \text{ удовлетворяет заданному критерию усечения} \\ 1 & \text{в остальных случаях} \end{cases} \tag{8.2-23}$$

для $u, v = 0, 1, 2, \ldots, n-1$. Приближение для $\mathbf{G}$ получается из усеченной последовательности

$$\hat{\mathbf{G}} = \sum_{u=0}^{n-1}\sum_{v=0}^{n-1}\chi(u,v)T(u,v)\mathbf{S}_{uv}, \tag{8.2-24}$$

где $\chi(u,v)$ строится так, чтобы удалять те базисные изображения, которые дают наименьший вклад в общую сумму в (8.2-21). Тогда средний квадрат ошибки между фрагментом $\mathbf{G}$ и его приближением $\hat{\mathbf{G}}$ будет равен

$$\begin{aligned} e_{ms} &= E\left\{\left\|\mathbf{G}-\hat{\mathbf{G}}\right\|^2\right\} = E\left\{\left\|\sum_{u=0}^{n-1}\sum_{v=0}^{n-1}T(u,v)\mathbf{S}_{uv} - \sum_{u=0}^{n-1}\sum_{v=0}^{n-1}\chi(u,v)T(u,v)\mathbf{S}_{uv}\right\|^2\right\} = \\ &= E\left\{\left\|\sum_{u=0}^{n-1}\sum_{v=0}^{n-1}T(u,v)\mathbf{S}_{uv}[1-\chi(u,v)]\right\|^2\right\} = \sum_{u=0}^{n-1}\sum_{v=0}^{n-1}\sigma^2_{T(u,v)}[1-\chi(u,v)], \end{aligned} \tag{8.2-25}$$

где $E\{\cdot\}$ — математическое ожидание, $\|\mathbf{G} - \hat{\mathbf{G}}\|$ — норма матрицы $(\mathbf{G} - \hat{\mathbf{G}})$, а $\sigma^2_{T(u,v)}$ — дисперсия коэффициентов $T(u,v)$ в точке (u,v). При получении последнего выражения в (8.2-25) использовано свойство ортогональности базисных изображений, а также предположение, что элементы $\mathbf{G}$ порождаются случайным процессом с нулевым средним и известной ковариацией. Таким образом, суммарный средний квадрат ошибки приближения равен сумме дисперсий коэффициентов отброшенных членов последовательностей (т. е. тех, для которых $\chi(u,v) = 0$, а $[1 - \chi(u,v)]$ в (8.2-25) равно 1). Преобразования, которые перераспределяют или упаковывают максимальное количество информации в наименьшее число коэффициентов, обеспечивают наилучшее приближение элементов блока и, как результат, дают наименьшую ошибку восстановления. Наконец, согласно тем же предположениям, что привели к (8.2-25), средние квадраты ошибки MN/n^2 блоков на изображении размерами $M \times N$ совпадают. Следовательно, средний квадрат ошибки (являющийся мерой *средней* ошибки) для изображения размерами $M \times N$ равен среднему квадрату ошибки отдельного блока.

Предыдущий пример показал, что ДКП обладает лучшей способностью к упаковке информации по сравнению с ДПФ и ПУА. Хотя эта ситуация справедлива для большинства изображений, тем не менее оптимальным в смысле упаковки информации является преобразование Карунена—Лоэва (ПКЛ будет рассматриваться в главе 11), а не ДКП. То есть ПКЛ минимизирует средний квадрат ошибки в (8.2-25) для любого входного изображения и любого числа сохраняемых коэффициентов [Kramer, Mathews, 1956][16]. Однако поскольку ПКЛ зависит от преобразуемых данных, то получение базисных изображений для каждого блока изображения является нетривиальной вычислительной задачей. По этой причине ПКЛ для сжатия изображений используется редко. Вместо этого обычно применяются такие преобразования, как ДПФ, ПУА или ДКП, базисные изображения которых фиксированы (т. е. не зависят от входных данных). Из преобразований, не зависящих от входных данных, простейшими в реализации являются не синусоидальные, а такие, например, как ПУА. С другой стороны, преобразования, основанные на гармонических функциях (ДПФ,

[16] Оптимальность дополнительно обуславливается тем, что маскирующая функция (8.2-23) выбирает коэффициенты ПКЛ с максимальной дисперсией.

ДКП или аналогичные), лучше приближаются к оптимальной упаковке информации, достигаемой ПКЛ.

> В примере 8.13 50 % наименьших коэффициентов блочных трансформационных кодов ДПФ, ПУА и ДКП (при использовании блоков 8×8) были обнулены. После декодирования наименьшие СКО имели результаты, основанные на ДПФ, показывая тем самым, что было удалено минимальное (в смысле ошибки СКО) количество информации.

Вследствие этого большинство систем трансформационного кодирования основываются на ДКП, которое дает хороший компромисс между степенью упаковки информации и вычислительной сложностью. Доказательством того, что характеристики ДКП имеют большое практическое значение, является тот факт, что ДКП вошло в международный стандарт систем трансформационного кодирования. По сравнению с другими подобными преобразованиями, ДКП обеспечивает упаковку наибольшего количества информации в наименьшее число коэффициентов[17] (для большинства реальных изображений), а также минимизирует эффект появления блочной структуры, называемой *блоковыми искажениями*, проявляющейся в том, что на изображении становятся видны границы между соседними блоками. Последняя особенность выгодно выделяет ДКП среди других синусоидальных преобразований. Поскольку ДПФ характеризуется n-точечной периодичностью (см. раздел 4.6.3), то разрывы на границах блоков, показанные на рис. 8.25(*а*), приводят к появлению заметной высокочастотной составляющей. При усечении или квантовании коэффициентов ДПФ приграничные элементы блоков из-за явления Гиббса[18] принимают неверные значения, что приводит к возникновению блоковых искажений. Таким образом, границы между соседними блоками становятся заметными из-за того, что приграничные элементы блоков принимают искаженные значения. ДКП, представленное на рис. 8.25(*б*), уменьшает этот эффект, потому что его периодичность в $2n$ точек не приводит к разрывам на границах блока.

Выбор размеров блока

Другим важным фактором, от которого зависят ошибки трансформационного кодирования и вычислительная сложность, является размер блока. В большинстве приложений изображения разбиваются таким образом, что корреляция (избыточность) между соседними блоками уменьшается до некоторого допустимого уровня, причем размер блока n выбирается как целая степень двойки. Последнее условие позволяет упростить вычисление преобразований по блокам (см. метод последовательного удвоения в разделе 4.11.3). Вообще с увеличением

---

[17] Ахмед и др. [Ahmed et al., 1974] первыми заметили, что базисные изображения ПКЛ для марковского источника первого порядка имеют близкое сходство с базисными изображениями ДКП. Если корреляция между соседними пикселями приближается к единице, то базисные изображения зависимого от входных данных ПКЛ совпадают с базисными изображениями независимого от входных данных ДКП [Clarke, 1985].

[18] Это явление, описываемое во многих работах, посвященных анализу электрических цепей, возникает из-за того, что ряд Фурье не является равномерно сходящимся вблизи разрывов. В точке разрыва ряд Фурье принимает среднее значение.

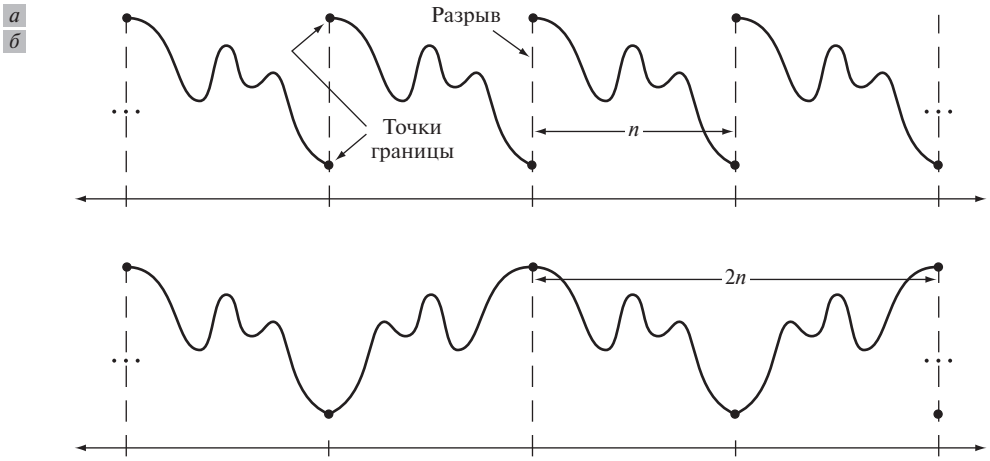


Рис. 8.25. Периодичность, присущая одномерным (*а*) ДПФ и (*б*) ДКП

размера блока возрастает как степень сжатия, так и вычислительная сложность. Наиболее часто используются размеры 8×8 и 16×16 элементов.

Пример 8.14. Влияние размера блока в трансформационном кодировании.
■ На рис. 8.26 графически иллюстрируется влияние размера блока на точность восстановления при трансформационном кодировании. Данные, приведенные на графиках, были получены разбиением полутонового изображения на рис. 8.9(*а*) на блоки размерами $n \times n$, где $n = 2, 4, 8, 16, ..., 256, 512$, вычислением преобразования по каждому блоку, усечением 75 % полученных коэффициентов и выполнением обратного преобразования. Заметим, что кривые ПУА и ДКП становятся почти горизонтальными при размерах блока, бóльших 8×8, тогда как ошибки восстановления ДПФ в этой области продолжают уменьшаться. Если n увеличивается еще больше, кривая ошибок ДПФ пересекает кривую ПУА и приближается к ДКП. Фактически эти данные совпадают с теоретическими и экспериментальными результатами, опубликованными в работах [Netravali, Limb, 1980] и [Pratt, 1991] для двумерной марковской модели источника.

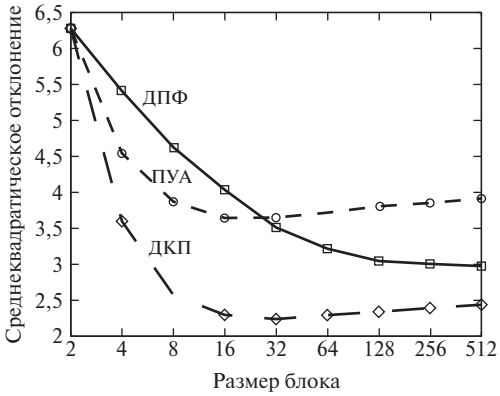


Рис. 8.26. Зависимость ошибки восстановления от размеров блока

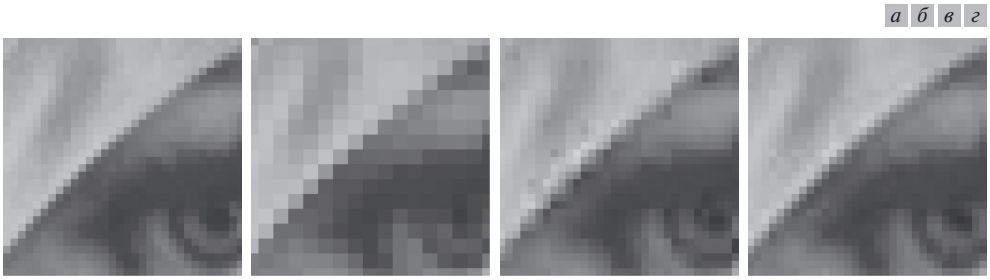


Рис. 8.27. Приближение изображения (*а*) при сохранении 25 % коэффициентов ДКП и различных размерах блока: (*а*) исходный фрагмент изображения на рис. 8.9(*а*); (*б*) размер блока 2×2; (*в*) размер блока 4×4; (*г*) размер блока 8×8

При размерах блока 2×2 все три кривые совпадают. В этом случае остается только один из четырех (25 %) коэффициентов в каждом преобразованном массиве. Этот коэффициент во всех преобразованиях является постоянной составляющей, так что обратное преобразование попросту заменяет значения всех четырех пикселей блока их средним значением (см. (4.6-21)). Этот эффект хорошо виден на рис. 8.27(*б*), где показан увеличенный фрагмент результата ДКП с блоками 2×2. Заметим, что блоковые искажения, которые максимальны на данном изображении, уменьшаются при увеличении размеров блока до 4×4 и 8×8 на рис. 8.27(*в*) и (*г*). Для сравнения на рис. 8.27(*а*) показан увеличенный фрагмент исходного изображения. ■

Двоичное представление

Ошибка восстановления, связанная с усечением разложений (8.2-24), является функцией числа и относительной важности отбрасываемых коэффициентов преобразования, а также точности, используемой для представления значений сохраняемых коэффициентов. В большинстве систем трансформационного кодирования выбор оставляемых коэффициентов, т. е. построение маскирующей функции в (8.2-23), осуществляется либо детерминированно на основе анализа дисперсии значений коэффициентов по всем блокам (*зональное кодирование*), либо адаптивно — выбором коэффициентов с максимальными значениями (*пороговое кодирование*). Весь процесс перевода в двоичную форму, включающий усечение, квантование и кодирование коэффициентов, обычно называют *двоичным представлением*.

Пример 8.15. Двоичное представление.

■ На рис. 8.28(*а*) и (*в*) показаны два приближения изображения на рис. 8.9(*а*), в которых при ДКП-преобразовании по блокам 8×8 были отброшены 87,5 % коэффициентов. Первый результат был получен применением порогового кодирования, оставляющего 8 самых больших коэффициентов каждого блока, а второй — с помощью зонального кодирования. В последнем случае каждый коэффициент ДКП рассматривался как случайная величина, распределение которой определялось по всему ансамблю блоков на изображении. Были найдены 8 распределений с максимальными дисперсиями (12,5 % из 64 коэффициентов блока 8×8), и в соответствии с их расположением была сформирована маска, аналогичная $\chi(u, v)$ в (8.2-24),

использовавшаяся для всех блоков. Обратите внимание, что при пороговом кодировании (рис. 8.28(*б*)) разностное изображение содержит значительно меньше ошибок, чем при зональном кодировании (рис. 8.28(*г*)). Оба разностных изображения были подвергнуты градационной коррекции, чтобы сделать ошибки более заметными. Значения СКО для них составили 4,5 и 6,5 соответственно. ■

Реализация зонального кодирования

Зональное кодирование основано на концепции теории информации о количестве информации как мере неопределенности. Таким образом, коэффициенты преобразования с максимальной дисперсией содержат максимум информации и, значит, должны сохраняться в процессе кодирования. Сами же дисперсии могут быть вычислены либо напрямую из ансамбля MN/n^2 массивов преобразованных блоков, также как и в предыдущем примере, либо на основании принятой модели изображения (скажем, марковской автокорреляционной функции). В любом случае, согласно (8.2-24), зональный отбор коэффициентов может рассматриваться как умножение каждого коэффициента $T(u, v)$ на соответствующие элементы *зональной маски*, которая аналогична коэффициентам $\chi(u, v)$ и содержит единицы в точках максимальной дисперсии и нули во всех остальных точках. Обычно коэффициенты с максимальной дисперсией располагаются

а *б*
в *г*

Рис. 8.28. Приближение изображения на рис. 8.9(*а*) при сохранении 12,5% коэффициентов ДКП по блокам 8×8: (*а*) и (*б*) — результаты порогового кодирования; (*в*) и (*г*) — результаты зонального кодирования. На разностных изображениях контраст увеличен в 4 раза

а *б*
в *г*

1	1	1	1	1	0	0	0
1	1	1	1	0	0	0	0
1	1	1	0	0	0	0	0
1	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

8	7	6	4	3	2	1	0
7	6	5	4	3	2	1	0
6	5	4	3	3	1	1	0
4	4	3	3	2	1	0	0
3	3	3	2	1	1	0	0
2	2	1	1	1	0	0	0
1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0

1	1	0	1	1	0	0	0
1	1	1	1	0	0	0	0
1	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

Рис. 8.29. Типичные (*а*) зональная маска, (*б*) распределение битов по зонам, (*в*) пороговая маска, (*г*) очередность взятия коэффициентов в зигзаг-упорядочивании (*Z*-упорядочивании). Затенением отмечены позиции сохраняемых коэффициентов

вблизи начала координат преобразованного блока; типичный пример зональной маски представлен ни рис. 8.29(*а*).

Коэффициенты, остающиеся в процессе зонального отбора, должны быть проквантованы и закодированы, поэтому иногда зональная маска изображается в виде массива чисел, каждое из которых означает число битов, отводимых для кодирования соответствующего коэффициента (рис. 8.29(*б*)). Коэффициентам при кодировании может отводиться как равное, так и неравное число битов. В первом случае коэффициенты, как правило, нормализуются по значению их стандартного отклонения, а затем равномерно квантуются. Во втором случае для каждого коэффициента (или группы коэффициентов) строится отдельный квантователь, подобный оптимальному квантователю Ллойда — Макса (см. оптимальный квантователь в разделе 8.2.9). При построении квантователей плотность распределения значений нулевых коэффициентов (т. е. средних значений в блоках) обычно моделируют распределением Рэлея, а плотность распределения значений оставшихся коэффициентов — распределением Лапласа или гауссовым распределением[19]. Число уровней квантования (а значит, чис-

[19] Поскольку коэффициенты являются линейными комбинациями значений элементов в блоке (см. (8.2-10)), то центральная предельная теорема утверждает, что при увеличении размеров блока распределение значений коэффициентов стремится к гауссову распределению. Этот результат, однако, неприменим к нулевому коэффициенту, поскольку неотрицательные изображения всегда имеют положительный коэффициент постоянной составляющей.

ло битов), отводимых каждому квантователю, выбирают пропорциональным $\log_2 \sigma^2_{T(u,v)}$. Таким образом, число битов, отводимое оставшимся коэффициентам в (8.2-24) (которые в данном случае выбираются согласно критерию максимальной дисперсии), должно быть пропорционально логарифму дисперсии коэффициентов[20].

Реализация порогового кодирования

При зональном кодировании для всех блоков обычно используется одна фиксированная маска. Пороговое кодирование, наоборот, является по существу адаптивным, поскольку позиции сохраняемых коэффициентов преобразования зависят от конкретного блока. Фактически пороговое кодирование является адаптивным подходом к трансформационному кодированию, которое благодаря своей вычислительной простоте чаще всего и используется на практике. В его основе лежит тот принцип, что в любом блоке коэффициенты преобразования, имеющие наибольшую амплитуду, дают самый значительный вклад в информационное содержание восстанавливаемого блока, что и было продемонстрировано на последнем примере. Поскольку положения самых больших коэффициентов от блока к блоку меняются, элементы $\chi(u, v)T(u, v)$ упорядочиваются (предварительно заданным способом) в одномерную последовательность, впоследствии кодируемую кодом длин серий[21]. На рис. 8.29(*в*) представлен пример типичной *пороговой маски* для одного блока некоторого гипотетического изображения. Эта маска позволяет проиллюстрировать процесс порогового кодирования, математически описываемого формулой (8.2-24). После маскирования двумерный массив из $n \times n$ коэффициентов переупорядочивается с помощью зигзаг-упорядочивания (называемого также Z-упорядочиванием). Упорядочивание зигзагом очевидно из рис. 8.29(*г*), где показана очередность, в которой берутся коэффициенты. Сформированный так одномерный массив из n^2 коэффициентов содержит длинные серии постоянных кодов, во второй половине — нулей, которые хорошо сжимаются кодированием длин серий. Получаемая кодовая последовательность подвергается еще одному этапу кодирования, уже с применением одного из алгоритмов неравномерного кодирования.

> N в «кодировании N-наибольших» не является размером изображения, а указывает на число сохраняемых коэффициентов.

Существует три основных способа разделения коэффициентов преобразования в блоке по порогу или, иначе говоря, построения пороговой маскирующей функции в форме, заданной формулой (8.2-23): (1) использование единого

---

[20] Такое распределение битов согласуется с теорией взаимосвязи скорости и искажения, которая гласит, что гауссова случайная переменная с дисперсией σ^2 при воспроизведении со средним квадратом ошибки меньше, чем D, не может быть представлена менее чем $\frac{1}{2}\log_2(\sigma^2/D)$ битами. Вывод таков, что информационное содержание гауссовой случайной переменной пропорционально $\log_2(\sigma^2/D)$. — *Прим. перев.*

[21] На самом деле смысл упорядочивания состоит в такой перегруппировке коэффициентов, чтобы получить максимально длинные серии из ненулевых и нулевых коэффициентов. Заметим также, что к настоящему моменту коэффициенты блоков уже являются квантованными и кодированными по значениям уровней квантования. — *Прим. перев.*

глобального порога, одинакового для всех блоков; (2) использование индивидуальных порогов для каждого блока; (3) переменный порог, который может изменяться как функция положения коэффициента в блоке. В первом случае уровень сжатия может меняться от изображения к изображению в зависимости от того, сколько коэффициентов оказываются выше или ниже порога. Второй вариант, называемый *кодированием N-наибольших*, оставляет одинаковое количество коэффициентов в каждом блоке. Как результат, скорость кода является постоянной и заранее известной. Третий метод, как и первый, приводит к коду непостоянной скорости, но зато имеет то преимущество, что позволяет объединить этапы квантования *и* разделения по порогу, заменяя $\chi(u,v)T(u,v)$ в (8.2-24) на

$$\hat{T}(u,v) = \text{round}\left[\frac{T(u,v)}{Z(u,v)}\right], \tag{8.2-26}$$

где round[·] — операция округления до ближайшего целого, $\hat{T}(u,v)$ — результат разделения по порогу и квантования значения $T(u,v)$, а $Z(u,v)$ — элемент массива коэффициентов нормировки преобразования:

$$\mathbf{Z} = \begin{bmatrix} Z(0,0) & Z(0,1) & \cdots & Z(0,n-1) \\ Z(1,0) & \vdots & \cdots & \vdots \\ \vdots & \vdots & \cdots & \vdots \\ Z(n-1,0) & Z(n-1,1) & \cdots & Z(n-1,n-1) \end{bmatrix}. \tag{8.2-27}$$

Перед тем, как нормированные (квантованные и разделенные по порогу) коэффициенты преобразования $\hat{T}(u,v)$ могут быть подвергнуты обратному преобразованию для восстановления подызображения $g(x,y)$, они должны быть умножены на $Z(u,v)$. Получаемый в результате денормированный массив $\dot{T}(u,v)$ является приближением $T(u,v)$:

$$\dot{T}(u,v) = \hat{T}(u,v)Z(u,v). \tag{8.2-28}$$

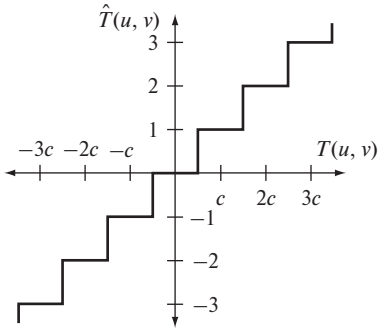
Обратное преобразование полученных значений $\dot{T}(u,v)$ дает в результате приближение восстанавливаемого блока изображения.

На рис. 8.30 графически изображена формула (8.2-26) для случая $Z(u,v) = c$. Заметим, что $\hat{T}(u,v)$ принимает целое значение k в том и только том случае, если

$$kc - \frac{c}{2} \le T(u,v) < kc + \frac{c}{2}. \tag{8.2-29}$$

Если $Z(u,v) > 2T(u,v)$, то $\hat{T}(u,v) = 0$ и коэффициент преобразования оказывается усеченным или вычеркнутым. Когда коэффициент $\hat{T}(u,v)$ представляется неравномерным кодом, длина которого возрастает с увеличением k, то, изменяя значение c, можно управлять числом битов, отводимых на представление $T(u,v)$. Т. е. элемент матрицы **Z** может масштабироваться, обеспечивая тем самым многообразие уровней сжатия. На рис. 8.30(*б*) показаны значения типичного массива коэффициентов нормировки. Этот массив значений, являющихся весовыми множителями для коэффициентов преобразования в блоке, составленный на основе оценок визуального восприятия, широко используется в стандарте JPEG, рассматриваемом в следующем разделе.

а б



16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Рис. 8.30. (*а*) График функции квантования в пороговом кодировании (см. формулу (8.2-29)). (*б*) Типичная матрица коэффициентов нормировки

Пример 8.16. Иллюстрация порогового кодирования.

■ На рис. 8.31(*а*)—(*е*) показаны шесть вариантов приближения полутонового изображения на рис. 8.9(*а*) методом порогового кодирования. Все изображения получены при использовании ДКП по блокам 8×8 и массива коэффициентов нормировки, приведенного на рис. 8.30(*б*). Первый вариант, обеспечивающий коэффициент сжатия 12:1 (т. е. $C = 12$), был получен прямым применением коэффициентов нормировки. Остальные результаты, на которых изображение сжато с коэффициентами 19, 30, 49, 85 и 182, получен с помощью масштабирова-

а б в
г д е

Рис. 8.31. Приближения изображения на рис. 8.9(*а*) при использовании ДКП и массива коэффициентов нормировки на рис. 8.30(*б*): (*а*) Z, (*б*) 2Z, (*в*) 4Z, (*г*) 8Z, (*д*) 16Z, (*е*) 32Z

ния (предварительного умножения массива коэффициентов нормировки) на 2, 4, 8, 16 и 32 соответственно. Стандартные отклонения ошибок составили соответственно 3,83, 4,93, 6,62, 9,35, 13,94 и 22,46 уровней яркостей. ■

JPEG

Одним из наиболее популярных и универсальных стандартов сжатия полутоновых неподвижных изображений является стандарт JPEG. Он определяет три различных режима кодирования: (1) режим последовательного кодирования с потерями, основанный на ДКП и подходящий для большинства применений; (2) расширенный режим кодирования, используемый для большего сжатия, для более высокой точности или для постепенного воспроизведения; и (3) режим кодирования без потерь, гарантирующий точное восстановление информации после сжатия. Чтобы быть совместимым со стандартом JPEG, продукт или система должны обеспечивать поддержку режима последовательного кодирования. При этом точно не определяются ни формат файла, ни пространственное разрешение, ни модель цветового пространства.

В системе с последовательной обработкой (кодированием), часто называемой *системой последовательной развертки*, точность входных и выходных данных ограничена 8 битами, а точность квантованных коэффициентов ДКП ограничена 11 битами. Сам процесс сжатия состоит из трех последовательных шагов: вычисление ДКП, квантование и затем кодирование неравномерным кодом. Сначала изображение разбивается на отдельные блоки размерами 8×8 элементов, которые обрабатываются последовательно слева направо и сверху вниз. Обработка каждого блока начинается со сдвига по яркости значений всех его 64 элементов, что достигается вычитанием величины 2^{k-1}, где 2^k — максимальное число уровней яркости. Затем вычисляется двумерное дискретное косинусное преобразование элементов блока. Полученные значения коэффициентов квантуются в соответствии с формулой (8.2-26), переупорядочиваются зигзаг-преобразованием согласно рис. 8.29(*д*), в результате чего формируется одномерная последовательность квантованных коэффициентов.

Одномерный массив, полученный после зигзаг-преобразования в соответствии с рис. 8.29(*д*), упорядочивается по возрастанию пространственной частоты; при этом, как правило, возникают длинные последовательности нулей, что эффективно используется процедурой JPEG-кодирования. В частности, ненулевые коэффициенты AC[22] кодируются неравномерным кодом, определяющим одновременно и значение коэффициента, и число предшествующих нулей. Текущий коэффициент DC кодируется дифференциальным кодом как разность с коэффициентом DC предыдущего блока. Табл. A-3, A-4 и A-5 в приложении A содержат составленные группой JPEG и задаваемые по умолчанию стандартные коды Хаффмана для яркостной составляющей цветного или черно-белого изображения. Рекомендованный JPEG массив квантования яркостей представлен на рис. 8.30(*б*) и может масштабироваться для получения множества уровней сжатия. Масштабирование значений в этом массиве дает пользователю возможность выбирать «качество» JPEG-сжатия. Хотя как для яркости, так и для цветности предусмотрены стандартные таблицы кодирования, а также проверенные

[22] Согласно стандарту нулевой коэффициент обозначается DC, а остальные коэффициенты — AC.

шкалы квантования, тем не менее допускается построение пользовательских таблиц и шкал, адаптированных к характеристикам сжимаемого изображения.

Пример 8.17. Последовательное кодирование и декодирование JPEG.
■ Рассмотрим сжатие и восстановление следующего блока из 8×8 элементов согласно стандарту последовательного кодирования JPEG:

52	55	61	66	70	61	64	73
63	59	66	90	109	85	69	72
62	59	68	113	144	104	66	73
63	58	71	122	154	106	70	69
67	61	68	104	126	88	68	70
79	65	60	70	77	68	58	75
85	71	64	59	55	61	65	83
87	79	69	68	65	76	78	94

Исходные значения пикселей могут иметь 256 или 2^8 возможных уровней яркости, так что процесс кодирования начинается со сдвига диапазона значений — вычитания из значений пикселей величины 2^7 или 128. В результате получится массив:

−76	−73	−67	−62	−58	−67	−64	−55
−65	−69	−62	−38	−19	−43	−59	−56
−66	−69	−60	−15	16	−24	−62	−55
−65	−70	−57	−6	26	−22	−58	−59
−61	−67	−60	−24	−2	−40	−60	−58
−49	−63	−68	−58	−51	−65	−70	−53
−43	−57	−64	−69	−73	−67	−63	−45
−41	−49	−59	−60	−63	−52	−50	−34

который, после прямого ДКП согласно (8.2-10) и (8.2-18) для $n = 8$ будет иметь вид:

−415	−29	−62	25	55	−20	−1	3
7	−21	−62	9	11	−7	−6	6
−46	8	77	−25	−30	10	7	−5
−50	13	35	−15	−9	6	0	3
11	−8	−13	−2	−1	1	−4	1
−10	1	3	−3	−1	0	2	−1
−4	−1	2	−1	2	−3	1	−2
−1	−1	−1	−2	−1	−1	0	−1

Если для квантования полученных данных используется рекомендованный JPEG нормирующий массив, приведенный на рис. 8.30(*б*), то после масштаби-

рования и усечения (то есть нормировки в соответствии с (8.2-26)) коэффициенты примут следующие значения:

$$\begin{matrix} -26 & -3 & -6 & 2 & 2 & 0 & 0 & 0 \\ 1 & -2 & -4 & 0 & 0 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

где, например, коэффициент DC вычислен следующим образом:

$$\hat{T}(0,0) = \text{round}\left[\frac{T(0,0)}{Z(0,0)}\right] = \text{round}\left[\frac{-415}{16}\right] = -26\,.$$

Заметим, что процедура преобразования и нормировки дает значительное число нулевых коэффициентов. После того, как коэффициенты переупорядочены в соответствии с зигзаг-преобразованием (см. рис. 8.29(*д*)), получится следующая одномерная последовательность коэффициентов:

[–26 –3 1 –3 –2 –6 2 –4 1 –4 1 1 5 0 2 0 0 –1 2 0 0 0 0 0 –1 –1 EOB].

Предусмотрено специальное кодовое слово EOB, означающее конец блока (см. код длины серии 0 и категории 0 в табл. А-5 кодов Хаффмана), указывающее, что все оставшиеся коэффициенты в переупорядоченной последовательности равны нулю.

Построение JPEG кода для переупорядоченной последовательности коэффициентов начинается с вычисления разности между значениями коэффициентов DC в текущем и предыдущем (уже закодированном) блоках. Предположим, что коэффициент DC соседнего левого, уже преобразованного и закодированного блока равен –17; значение разности, используемой в последующем ДИКМ-кодировании, составит (–26 – (–17)) = –9, которое попадает в категорию 4 разностей DC в табл. А-3. Согласно стандартным кодам Хаффмана для разностей из табл. А-4, правильный основной код будет 101 (3-битовый код). Однако суммарная длина полностью закодированного коэффициента категории 4 составит 7 бит — оставшиеся 4 бита должны быть взяты из младших разрядов (МР) значения разности. В общем случае для конкретной категории разностей DC (скажем, категории *K*) дополнительно требуется *K* битов, которые вычисляются или как *K* младших разрядов положительной разности, или как *K* младших разрядов отрицательной разности минус 1. Для разницы –9 соответствующие значения МР составят (0111) – 1 или 0110, и, таким образом, полное кодированное ДИКМ кодовое слово будет 1010110.

Ненулевые коэффициенты AC переупорядоченного массива кодируются аналогичным образом по табл. А-3 и А-5. Разница состоит лишь в том, что выбор кодового слова кода Хаффмана для коэффициента AC зависит как от категории амплитуды коэффициента, так и от числа предшествующих нулей (см. колонку «Длина серии/категория» в табл. А-5). Окончательный код первого ненулевого коэффици-

ента AC переупорядоченного массива (–3) будет 0100. Первые 2 бита данного кода указывают, что коэффициент был из категории 2 и что у него нет предшествующих нулевых коэффициентов (см. табл. A-3); последние 2 бита были получены процедурой добавления MP, аналогичной изложенной выше для кода разностей DC. Если продолжать подобным образом, полная кодовая последовательность переупорядоченного массива будет выглядеть

1010110 0100 001 0100 0101 100001 0110 100011 001 100011 001
001 100101 11100110 110110 0110 11110100 000 1010.

Пробелы между кодовыми словами поставлены здесь исключительно для удобства чтения. Хотя это и не потребовалось в данном примере, таблица стандартных кодов Хаффмана содержит специальное кодовое слово для серии длиной в 15 нулей, за которой опять идет 0 (см. серию F и категорию 0 в табл. A-5). Общее число битов, требуемых для кодирования переупорядоченного массива (а значит, требуемых для кодирования всех 8×8 элементов выбранного блока), составляет 92. Результирующий коэффициент сжатия равен 512 : 92, или около 5,6 : 1.

При восстановлении сжатого JPEG блока декодер в первую очередь должен из непрерывного потока битов воссоздать нормированные коэффициенты преобразования. Поскольку последовательность двоичных кодов Хаффмана является мгновенной и однозначно декодируемой (см. раздел 8.4.1), этот шаг легко реализуется при помощи табличного преобразования. Ниже приведен массив квантованных коэффициентов, восстановленный из потока битов:

–26	–3	–6	2	2	0	0	0
1	–2	–4	0	0	0	0	0
–3	1	5	–1	–1	0	0	0
–4	1	2	–1	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

После умножения на нормировочные коэффициенты согласно (8.2-28) получим массив:

–416	–33	–60	32	48	0	0	0
12	–24	–56	0	0	0	0	0
–42	13	80	–24	–40	0	0	0
–56	17	44	–29	0	0	0	0
18	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

где, например, коэффициент DC получен следующим образом:

$$\dot{T}(0,0) = \hat{T}(0,0)Z(0,0) = -26 \cdot 16 = -416.$$

Полностью восстановленный блок получается после выполнения обратного ДКП полученного массива согласно уравнениям (8.2-11) и (8.2-18), что дает

–70	–64	–61	–64	–69	–66	–58	–50
–72	–73	–61	–39	–30	–40	–54	–59
–68	–78	–58	–9	13	–12	–48	–64
–59	–77	–57	0	22	–13	–51	–60
–54	–75	–64	–23	–13	–44	–63	–56
–52	–71	–72	–54	–54	–71	–71	–54
–45	–59	–70	–68	–67	–67	–61	–50
–35	–47	–61	–66	–60	–48	–44	–44

и после обратного сдвига диапазона значений на 2^7 (т. е. +128) в результате получаем:

58	64	67	64	59	62	70	78
56	55	67	89	98	88	74	69
60	50	70	119	141	116	80	64
69	51	71	128	149	115	77	68
74	53	64	105	115	84	65	72
76	57	56	74	75	57	57	74
83	69	59	60	61	61	67	78
93	81	67	62	69	80	84	84

Все отличия значений элементов исходного и восстановленного блоков возникают вследствие самой природы сжатия с потерями, являющегося сутью процедур сжатия и восстановления методом JPEG. В данном примере ошибки восстановления находятся в диапазоне от –14 до +11 и распределены следующим образом:

–6	–9	–6	2	11	–1	–6	–5
7	4	–1	1	11	–3	–5	3
2	9	–2	–6	–3	–12	–14	9
–6	7	0	–4	–5	–9	–7	1
–7	8	4	–1	6	4	3	–2
3	8	4	–4	2	6	1	1
2	2	5	–1	–6	0	–2	5
–6	–2	2	6	–4	–4	–6	10

Среднеквадратическая ошибка, возникшая в результате всего процесса сжатия и восстановления, составляет приблизительно 5,8 уровней яркости. ■

Пример 8.18. Иллюстрация кодирования JPEG.
■ На рис. 8.32(*а*) и (*г*) приведены два варианта JPEG-приближений полутонового изображения рис. 8.9(*а*). На первом изображении достигнуто сжатие около 25:1, а на втором — 52:1. Разности между исходным изображением и восстановленными на рис. 8.32(*а*) и (*г*) показаны на рис. 8.32(*б*) и (*д*) соответственно. Соответствующие значения СКО составляют 5,4 и 10,7 уровней яркости. На рис. 8.32(*в*) и (*е*) — увеличенных фрагментах изображений (*а*) и (*г*) — прекрасно видны возникшие ошибки. Отметим, что блоковые артефакты JPEG возрастают с увеличением сжатия. ■

8.2.9. Кодирование с предсказанием

Обратимся теперь к более простому варианту, который позволяет достичь хорошего уровня сжатия при сравнительно небольших вычислительных затратах *и* притом может быть как с потерями, так и без потерь. Данный подход, называемый *кодированием с предсказанием*, основан на устранении межэлементной избыточности близко расположенных пикселей — в пространстве и/или во времени — путем выделения и кодирования только новой информации, содержащейся в каждом пикселе. Эта *новая информация*, содержащаяся в пикселе, определяется как разность между истинным и предсказанным значениями пикселя.

а *б* *в*
г *д* *е*

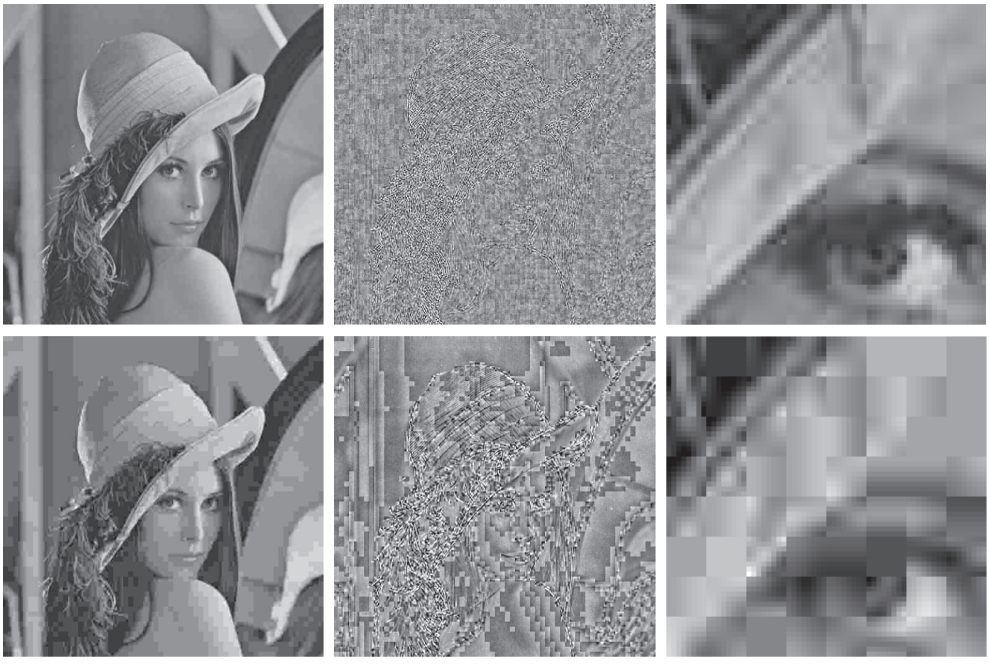


Рис. 8.32. Два JPEG-приближения изображения на рис. 8.9(*а*). В каждом ряду приведено по 3 изображения: результат после сжатия и восстановления, контрастированная разность между результатом и исходным изображением, увеличенный фрагмент восстановленного изображения

Согласно табл. 8.3 и 8.4 кодирование с предсказанием используется в следующих стандартах сжатия и файловых форматах:

- JBIG2,
- JPEG,
- JPEG-LS,
- MPEG-1,2,4,
- H.261, H.262, H.263 и H.264,
- HDV,
- VC-1,

а также в других стандартах сжатия и файловых форматах.

Кодирование с предсказанием без потерь

На рис. 8.33 представлены основные элементы системы *кодирования с предсказанием без потерь*. Система состоит из кодера и декодера, причем каждый содержит одинаковый *предсказатель*. Когда последовательные отсчеты входного сигнала дискретного времени, обозначаемые $f(n)$, поступают на вход кодера, предсказатель генерирует оценку значения каждого отсчета, основанную на значениях заданного количества предыдущих отсчетов. Затем выход предсказателя округляется до ближайшего целого, обозначаемого $\hat{f}_n$, и используется для получения разности, или *ошибки предсказания*

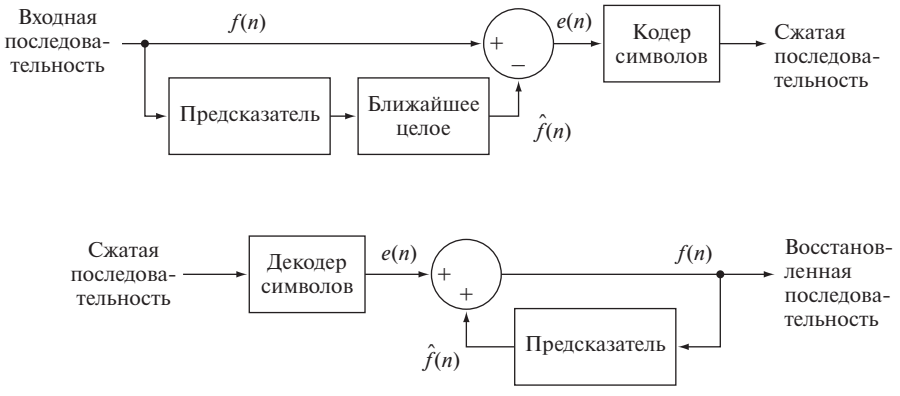
$$e(n) = f(n) - \hat{f}(n), \tag{8.2-30}$$

которая затем кодируется кодером символов при помощи неравномерного кода, и тем самым формируется очередной элемент сжатого потока данных. Декодер на рис. 8.33(*б*) восстанавливает значение $e(n)$ из полученной кодовой последовательности и выполняет обратную операцию

$$f(n) = e(n) + \hat{f}(n), \tag{8.2-31}$$

чтобы восстановить исходную входную последовательность.

Для формирования предсказываемого значения $\hat{f}_n$ могут использоваться различные локальные, глобальные или адаптивные методы (см. ниже разделы, касающиеся кодирования с потерями с предсказанием). Однако в большинстве



а
б

Рис. 8.33. Модель кодирования с предсказанием без потерь: (*а*) кодер; (*б*) декодер

случаев предсказание формируется как линейная комбинация m предыдущих отсчетов:

$$\hat{f}(n) = \text{round}\left[\sum_{i=1}^{m} \alpha_i f(n-i)\right], \tag{8.2-32}$$

где m — *порядок* линейного предсказателя, α_i — коэффициенты предсказания ($i = 1, 2,..., m$), а операция round[·] означает округление до ближайшего целого. Если отсчеты входной последовательности на рис. 8.33(*а*) рассматривать применительно к изображению, то $f(n)$ в выражениях (8.2-30)—(8.2-32) суть значения пикселей, а m отсчетов, используемых для предсказания значения очередного пикселя, могут браться из той же строки сканирования (что называется одномерным линейным кодированием с предсказанием), из текущей и предыдущих строк (двумерное линейное кодирование с предсказанием) или даже из текущего изображения и предшествующих ему изображений (трехмерное линейное кодирование с предсказанием). Для одномерного линейного кодирования с предсказанием выражение (8.2-32) может быть переписано в виде

$$\hat{f}(x,y) = \text{round}\left[\sum_{i=1}^{m} \alpha_i f(x,y-i)\right], \tag{8.2-33}$$

где каждый отсчет теперь выражается исключительно как функция пространственных координат x и y исходного изображения. Согласно (8.2-33) одномерное линейное предсказание есть функция значений предыдущих элементов одной текущей обрабатываемой строки. В случае двумерного кодирования с предсказанием величина предсказания будет, вообще говоря, являться функцией всех предыдущих элементов, полученных к настоящему моменту при последовательном сканировании слева направо и сверху вниз. В трехмерном случае будут также добавляться значения пикселей из предыдущих кадров. Для первых m элементов каждой строки выражение (8.2-33) не может быть определено, поэтому эти элементы должны кодироваться каким-то другим способом (например кодом Хаффмана) и рассматриваться как издержки процесса кодирования с предсказанием. Эти комментарии также относятся и к случаям двумерного или трехмерного кодирования.

Пример 8.19. Кодирование с предсказанием и пространственная избыточность.

■ Рассмотрим кодирование полутонового изображения на рис. 8.34(*а*) с помощью простого линейного предсказателя (8.2-33) первого порядка (т. е. $m = 1$):

$$\hat{f}(x,y) = \text{round}\left[\alpha f(x,y-1)\right]. \tag{8.2-34}$$

Данное выражение является упрощением выражения (8.2-33) для $m = 1$ и отбрасыванием ненужного индекса при единственном коэффициенте предсказания α_1. Предсказатель такого общего вида называется предсказателем *по предыдущему элементу*, и соответствующая процедура кодирования называется *дифференциальным кодированием* или *кодированием по предыдущему элементу*. На рис. 8.34(*в*) в виде изображения показано значение (сигнал) ошибки предсказания $e(x,y) = f(x,y) - \hat{f}(x,y)$, получаемое из (8.2-34) при $\alpha = 1$. На этом изо-

бражении значение яркости 128 соответствует нулевой ошибке предсказания, а отличные от нуля положительные и отрицательные ошибки предсказания усилены и изображаются соответственно более яркими или более темными оттенками. Среднее значение данного изображения составляет 128,26. Поскольку значение 128 соответствует нулевой ошибке предсказания, средняя ошибка предсказания составляет всего 0,26 бит.

На рис. 8.34(*б*) и (*г*) представлены гистограмма уровней яркостей исходного изображения (приведенного на рис. 8.34(*а*)) и гистограмма ошибок предсказания $e(x, y)$ соответственно. Заметим, что стандартное отклонение ошибок предсказания на рис. 8.34(*г*) много меньше стандартного отклонения значений яркостей исходного изображения. Более того, энтропия сигнала ошибки предсказания, оцениваемая согласно выражению (8.1-7), также значительно меньше, чем соответствующая оценка энтропии исходного изображения (3,99 бит/пиксель против 7,25 бит/пиксель). Это уменьшение энтропии отражает сокращение значительной части пространственной избыточности при помощи процесса кодирования с предсказанием, несмотря на то, что для точного представления $e(x, y)$ — последовательности ошибок предсказания k-битового изображения — требуются $(k + 1)$-битовые числа. Вообще оценка максимального сжатия любого варианта кодирования с предсказанием без потерь может быть получена делением среднего числа битов, используемого для представления значения одного пикселя, на оценку энтропии сигнала ошибки предсказания. Хотя для кодирования данной последовательности ошибок предсказания $e(x, y)$ может быть использована любая из процедур неравномерного кодирования, результирующий коэффициент сжатия будет ограничен величиной приблизительно 8/3,99 или около 2:1. ■

Заметим, что ошибка предсказания, закодированная неравномерным кодом, представляет собой сжатое изображение.

Предыдущий пример показывает, что величина сжатия, достигаемая при кодировании с предсказанием без потерь, прямо связана с уменьшением энтропии, происходящим благодаря отображению входного изображения в последовательность *ошибок предсказания*. Поскольку в процессе предсказания и вычисления разности удаляется значительная доля межэлементной избыточности, распределение вероятностей ошибок предсказания имеет резкий пик в нуле и характеризуется относительно малой дисперсией (по сравнению с распределением яркостей входного изображения). Плотность распределения вероятностей ошибки предсказателя часто моделируют распределением Лапласа с нулевым средним:

$$p_e(e) = \frac{1}{\sqrt{2}\sigma_e} e^{\frac{-\sqrt{2}|e|}{\sigma_e}}, \qquad (8.2\text{-}35)$$

где σ_e — величина стандартного отклонения e.

Пример 8.20. Кодирование с предсказанием и временна́я избыточность.
■ Изображение на рис. 8.34(*а*) является фрагментом видео NASA; на кадре Земля движется слева направо по отношению к стационарной камере, установленной на космическом корабле «Спейс шаттл». Изображение повторено на рис. 8.35(*б*),

а рядом (рис. 8.35(*а*)) показан предшествующий кадр видеопоследовательности. Используя линейный предсказатель первого порядка

$$\hat{f}(x,y,t) = \text{round}\left[\alpha f(x,y,t-1)\right] \tag{8.2-36}$$

с $\alpha = 1$, яркости пикселей на рис. 8.34(*б*) можно предсказать по значениям соответствующих пикселей изображения (*а*). Рис. 8.34(*в*) — изображение результирующей ошибки предсказания $e(x,y,t) = f(x,y,t) - \hat{f}(x,y,t)$. Рис. 8.34(*г*) — гистограмма $e(x, y, t)$. Отметим, что имеющаяся ошибка предсказания весьма невелика. Ее стандартное отклонение значительно меньше, чем в предыдущем примере — всего 3,76 бит/пиксель по сравнению с 15,58 бит/пиксель. Кроме того, энтропия ошибки предсказания (вычисленная согласно (8.1-7)) уменьшилась с 3,99 до 2,59 бит/пиксель. Используя неравномерное кодирование результирующей ошибки предсказания, изображение (*б*) можно сжать приблизительно до 8:2,59 или 3,1:1 — с улучшением на 50 % по сравнению со сжатием 2:1, полученным при использовании пространственного предсказания по предыдущему пикселю в примере 8.19. ■

Напомним еще раз, что ошибка предсказания, закодированная неравномерным кодом, как раз и является сжатым изображением.

а *б*
в *г*

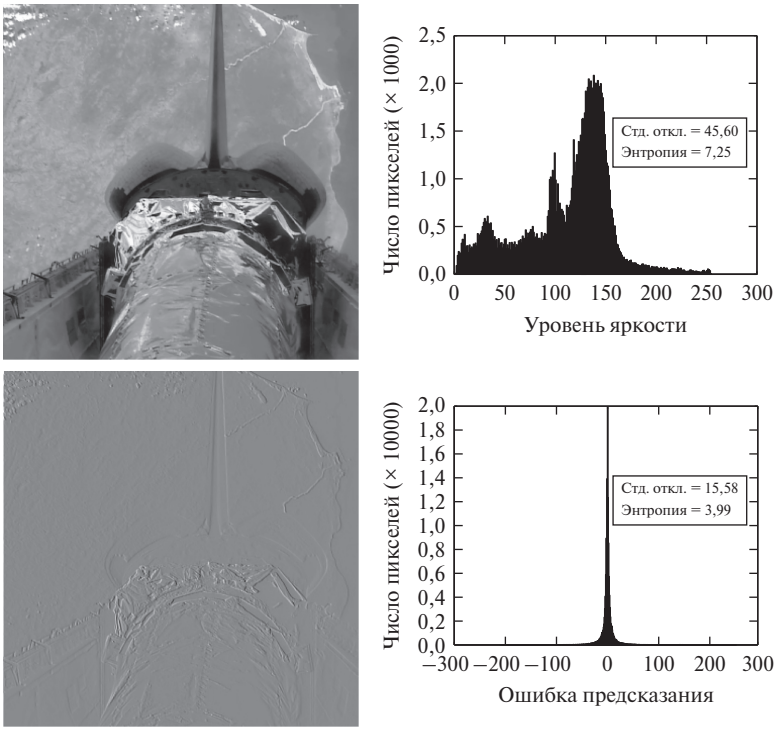


Рис. 8.34. (*а*) Вид Земли из орбитального космического корабля «Спейс шаттл». (*б*) Гистограмма значений яркости (*а*). (*в*) Изображение ошибки предсказания, полученной из (8.2-34). (*г*) Гистограмма ошибки предсказания. (Исходное изображение предоставлено NASA)

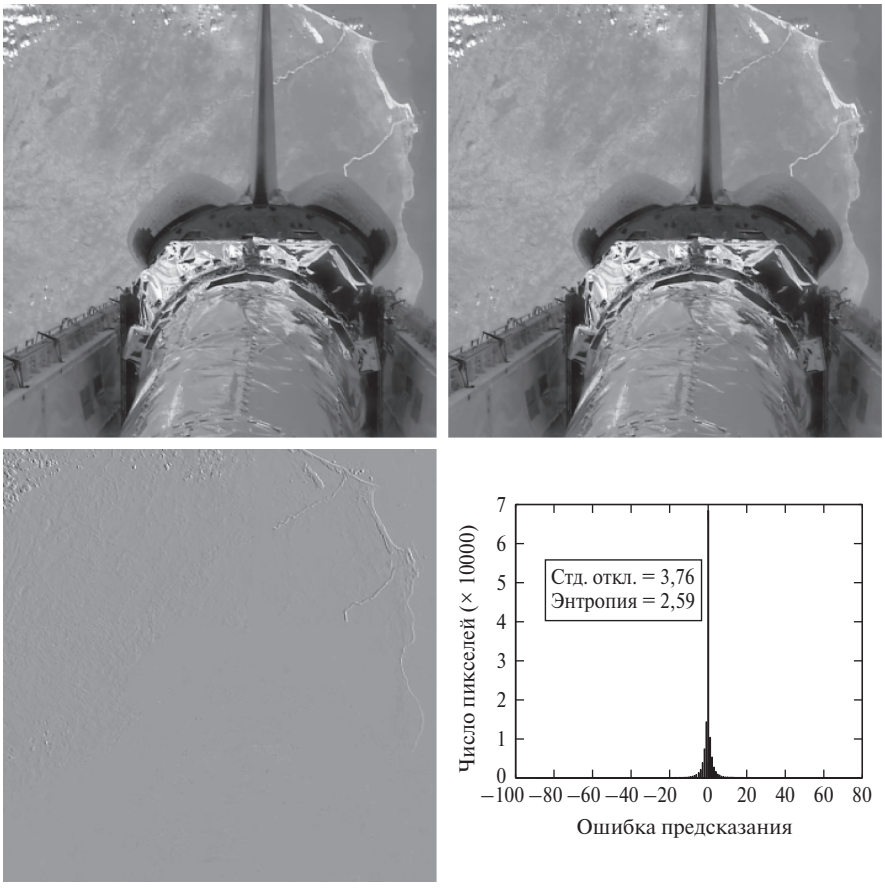


Рис. 8.35. (*а*) и (*б*) два кадра Земли из видеосъемки с космического корабля «Спейс шаттл». (*в*) Изображение ошибки предсказания, полученной согласно (8.2-36). (*г*) Гистограмма ошибки предсказания. (Исходное изображение предоставлено NASA)

Компенсация движения при кодировании с предсказанием

Как показано в примере 8.20, соседние кадры видеопоследовательности часто очень похожи. Кодирование разности между ними может уменьшить временну́ю избыточность и обеспечить существенное сжатие. Однако в ряде случаев, например при возникновении в последовательности кадров быстро движущихся объектов, при внезапных изменениях сцены, зуммировании, панорамировании, изменении яркости и др., подобие между соседними кадрами нарушается, и это отрицательно влияет на сжатие. То есть, как и большинство методов сжатия (см. пример 8.5), кодирование, основанное на временно́м предсказании, работает лучше всего с определенным типом исходных данных — а именно с последовательностью изображений, имеющей значительную временну́ю избыточность. Если же его использовать для изображений с малой временно́й избыточностью, то возможно увеличение общего объема данных. Во избежание увеличения объема данных системы сжатия видеопоследовательностей используют два подхода:

1. Прослеживание движения объекта и компенсация этого во время процессов предсказания и формирования разности.
2. Переключение на альтернативные способ кодирования, если *межкадровая* корреляция (подобие между кадрами) становится неудовлетворительной.

Первый из них — *компенсация движения* — является предметом оставшейся части данного раздела. Но прежде чем начать рассмотрение, заметим, что при недостаточной межкадровой корреляции, когда невозможно эффективное кодирование с предсказанием, обычно используется второй подход и применяется блочное двумерное трансформационное преобразование типа ДКП в JPEG-кодировании (см. раздел 8.2.8). Кадры, сжимаемые таким способом, называются *независимые кадры* (*intraframes* или *Independent frames*, *I-кадры*). Они могут декодироваться без обращения к другим кадрам видеопоследовательности, к которой они относятся. I-кадры обычно подобны изображениям, кодированным при помощи JPEG, и являются идеальной отправной точкой для начала процесса кодирования, использующего ошибки предсказания. Кроме того, наличие таких кадров в видеопоследовательности повышает возможности прямого доступа к частям видеопоследовательности, удобство ее редактирования, а также сокращает длительность воздействия ошибок передачи. По этим причинам все видеостандарты требуют периодического включения I-кадров в сжатый видеопоток.

Рис. 8.36 иллюстрирует принципы кодирования с предсказанием и компенсацией движения. Каждый видеокадр разбивается на непересекающиеся прямоугольные области — обычно размерами от 4×4 до 16×16, называемые *макроблоками*. (На рис. 8.36 показан лишь один такой макроблок.) Определяется движение (смещение) макроблока относительно предсказываемого «наиболее вероятного» его положения на предыдущем (или последующем) видеокадре, который называется *опорным кадром*, и эта величина сохраняется в кодируемом *векторе движения*. Вектор определяет движение макроблока заданием вертикального и горизонтального *смещений* относительно «наиболее вероятного» его положения. Смещение обычно задается с точностью целого пикселя или повышенной — ½ или ¼ пикселя. При использовании повышенной точности предсказания должны быть интерполированы (например, используя билинейную интерполяцию — см. раздел 2.4.4) по комбинации пикселей опорного кадра.

> «Наиболее вероятная» позиция есть такая, в которой минимизируется выбранная мера ошибки между кодируемым макроблоком и опорным изображением (в пределах допустимой области поиска). При этом, вообще говоря, макроблок и выбранный участок опорного изображения необязательно отражают один и тот же объект.

Последовательность кадров подразделяется на кадры трех типов: (1) вышеуказанные *независимые I-кадры*; (2) опорные *предсказываемые P-кадры* (*predicted frames*), идущие, как правило, с шагом в несколько (3—4) кадров видеопоследовательности и предсказываемые по предыдущему I- или P-кадру; и (3) промежуточные *двунаправленные B-кадры* (*bidirectional frames*), заполняющие промежутки между опорными P-кадрами (I-кадр здесь эквивалентен P-кадру) и предсказы-

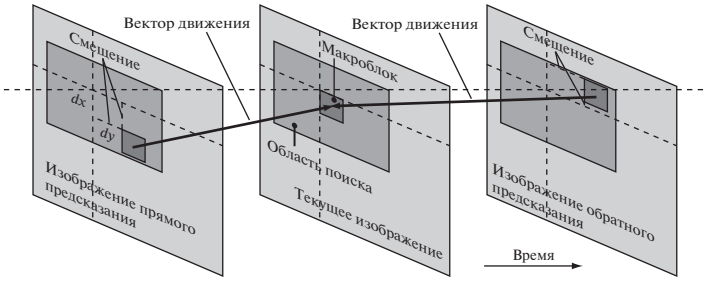


Рис. 8.36. Определение движения макроблока

ваемые по обоим этим кадрам (как вперед, так и назад по времени, см. рис. 8.36). Нумерация кадров типичной видеопоследовательности при этом будет иметь вид IBBPBBPBBPB...BPBI.... Наличие B-кадров, предусматривающих обратное по времени предсказание, требует изменения временно́го течения кодового потока, чтобы данные поступали в декодер в последовательности кодирования, а не в естественной последовательности кадров.

Как и следовало ожидать, ключевым компонентом компенсации движения является *оценка движения*. Во время оценки движения определяется смещение объектов (макроблоков), кодируется и записывается в вектор движения. Отыскание «наилучшего» вектора движения требует предварительного определения критерия оптимальности. Вектор движения может отыскиваться как точка максимума корреляции или минимума разности между пикселями макроблока и предсказываемыми пикселями (или интерполированными точками в случае смещения повышенной точности) из выбранного опорного кадра. Одной из наиболее часто используемых мер ошибки является величина *среднего абсолютного искажения* (*mean absolute distortion*, *MAD*)

$$MAD(x,y)=\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\left|f(x+i,y+j)-p(x+i+dx,y+j+dy)\right|, \qquad (8.2\text{-}37)$$

где x и y — координаты левого верхнего угла кодируемого макроблока размерами $m\times n$, dx и dy — смещения относительно опорного кадра, как показано на рис. 8.36; а $p(x,y)$ — массив значений предсказываемого кадра. Для оценки смещения повышенной точности массив p интерполируется по значениям опорного кадра. Обычно dx и dy должны попадать внутрь некоторой ограниченной области (см. рис. 8.36) вокруг макроблока. Как правило, пределы смещения выбирают от ±8 до ±64 пикселей, и горизонтальная область поиска обычно несколько больше вертикальной. Несколько более эффективная в вычислительном отношении мера ошибки, называемая *суммой аблолютных искажений* (*sum of absolute distortions*, *SAD*), выглядит так же, как (8.2-37), но без множителя $1/mn$.

Задавшись критерием выбора, подобным (8.2-37), оценка движения осуществляется отысканием dx и dy, минимизирующих $MAD(x,y)$ по допустимой области изменения вектора движения — включая значения смещения повышенной точности. Этот процесс обычно называют *привязкой блока*. Полный перебор возможных значений гарантирует наилучший результат, однако вычислительно трудоемок, поскольку каждый макроблок должен быть проверен по всему допустимому диапазону смещений. Так, для каждого макроблока раз-

мерами 16×16 и допустимого диапазона смещений ±32 по обоим направлениям (что немного для фильма с интенсивными движениями или же для спортивных мероприятий) необходимо выполнить 4225 вычислений *MAD* размерами 16×16 при одинарной точности смещения. Если же использовать повышенную точность — ½ или ¼ пикселя, объем вычислений возрастет в 4 или 16 раз соответственно. Этот объем накладных расходов могут сократить алгоритмы быстрого поиска, однако они не всегда могут давать оптимальное значение вектора движения. В литературе предлагаются и исследуются различные быстрые алгоритмы оценки движения блока (см., например, [Furht et al., 1997] или [Mitchell et al., 1997]).

Пример 8.21. Предсказание с компенсацией движения.

■ Изображения на рис. 8.37(*а*) и (*б*) были взяты из той же видеопоследовательности NASA, которая использовалась в примерах 8.19 и 8.20. Рис. 8.37(*б*) идентичен рис. 8.34(*а*) и 8.35(*б*); рис. 8.37(*а*) — аналогичный фрагмент той же видеопоследовательности, но снятый на 13 кадров раньше. Рис. 8.37(*в*) — разность между изображениями (*б*) и (*а*), откорректированная на полный диапазон яркостей. Отметим, что разность равна 0 в стационарной области (по отношению к камере) — на корабле «Спейс шаттл», но в оставшейся части изображения наблю-

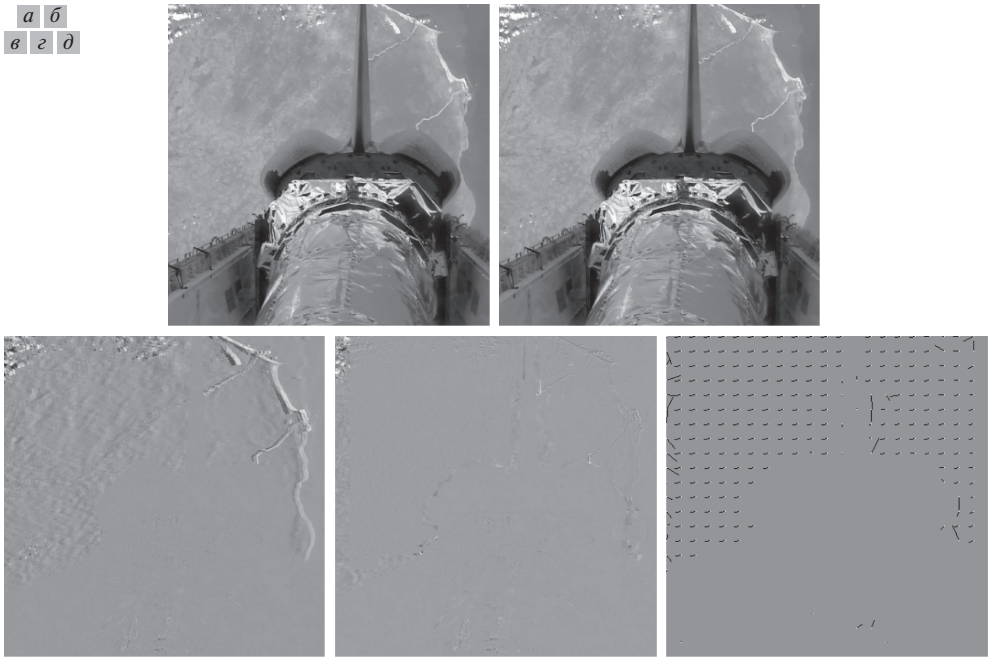


Рис. 8.37. (*а*) и (*б*) два кадра Земли из видеосъемки с космического корабля «Спейс шаттл» с промежутком 13 кадров. (*в*) Изображение ошибки предсказания без компенсации движения. (*г*) Ошибка предсказания с компенсацией движения. (*д*) Векторы движения, ассоциированные с (*г*); концы векторов движения приведены в левые верхние углы макроблоков, показанные белыми точками. (Исходное изображение предоставлено NASA)

даются значительные различия вследствие относительного движения Земли. Стандартное отклонение ошибки предсказания рис. 8.37(*в*) составило 12,73 уровня яркости, а энтропия (используя (8.1-7)) — 4,17 бит/пиксель. Максимальное сжатие ошибки предсказания, достигаемое при использовании неравномерных кодов, составляет $C = 8{:}4{,}17 = 1{,}92$.

На рис. 8.37(*г*) представлена ошибка предсказания с компенсацией движения; ее стандартное отклонение оказалось значительно меньше — 5,62 уровня яркости против 12,73 первоначальных, энтропия также несколько меньше — 3,04 против 4,17 бит/пиксель. Если использовать неравномерное кодирование для сжатия ошибки предсказания на рис. 8.37(*г*), то получим коэффициент сжатия $C = 8{:}3{,}04 = 2{,}63$. Данная ошибка предсказания была получена разбиением рис. 8.37(*б*) на неперекрывающиеся макроблоки 16×16 пикселей и отысканием оптимального фрагмента в рис. 8.37(*а*) в диапазоне смещения ±16 пикселей относительно положения макроблока в (*б*). Для нахождения наилучшей привязки осуществлялся выбор смещения (dx, dy) по формуле (8.2-37) с наименьшим MAD. Полученные значения смещений отражены в виде x и y компонентов векторов на рис. 8.37(*д*). Белые точки на рисунке расположены в левых верхних углах кодированных макроблоков; в них приведены концы показанных векторов движения. Как можно увидеть из направлений векторов, доминирующее движение изображения — слева направо. В нижней части изображения, соответствующей корпусу космического корабля, нет никакого движения, и поэтому векторы движения не показаны. В этой области макроблоки были предсказаны по соответствующим блокам опорного изображения, т. е. с теми же координатами. Поскольку векторы на рис. 8.37(*д*) сильно коррелированы, для снижения их объема и требований к каналу передачи они могут кодироваться неравномерными кодами. ■

Рис. 8.38 демонстрирует повышение точности предсказания, которое стало возможным при использовании более точной (субпиксельной) компенсации движения. Изображение на рис. 8.38(*а*) повторяет рис. 8.37(*в*) и включено сюда только для сравнения; оно демонстрирует ошибки предсказания, возникающие без компенсации движения. Изображения на рис. 8.38(*б*)—(*г*) — ошибки предсказания после компенсации движения. Они получены при использовании тех же двух кадров, что и в примере 8.21, но вычислялись при возможных точностях смещения макроблока на 1, ½ и ¼ пикселя соответственно. Использовались макроблоки размерами 8×8; диапазон смещения ограничивался ±8 пикселей.

> Видимая разница между рис. 8.37(*в*) и рис. 8.38(*а*) возникла из-за различной градационной коррекции. Изображение рис. 8.38(*а*) усилено в той же степени, как и изображения на рис. 8.38(*б*)—(*г*).

Наиболее заметными различиями между ошибками предсказания на рис. 8.38 является число и размеры яркостных пиков и впадин — наиболее светлых и темных областей яркости. Ошибка предсказания с точностью ¼ пикселя на рис. 8.38(*г*) выглядит наиболее «плоской» из всей четверки, с наименьшими отклонениями в черное и белое. Как и следовало ожидать, она имеет самую узкую гистограмму. Значения стандартных отклонений ошибки пред-

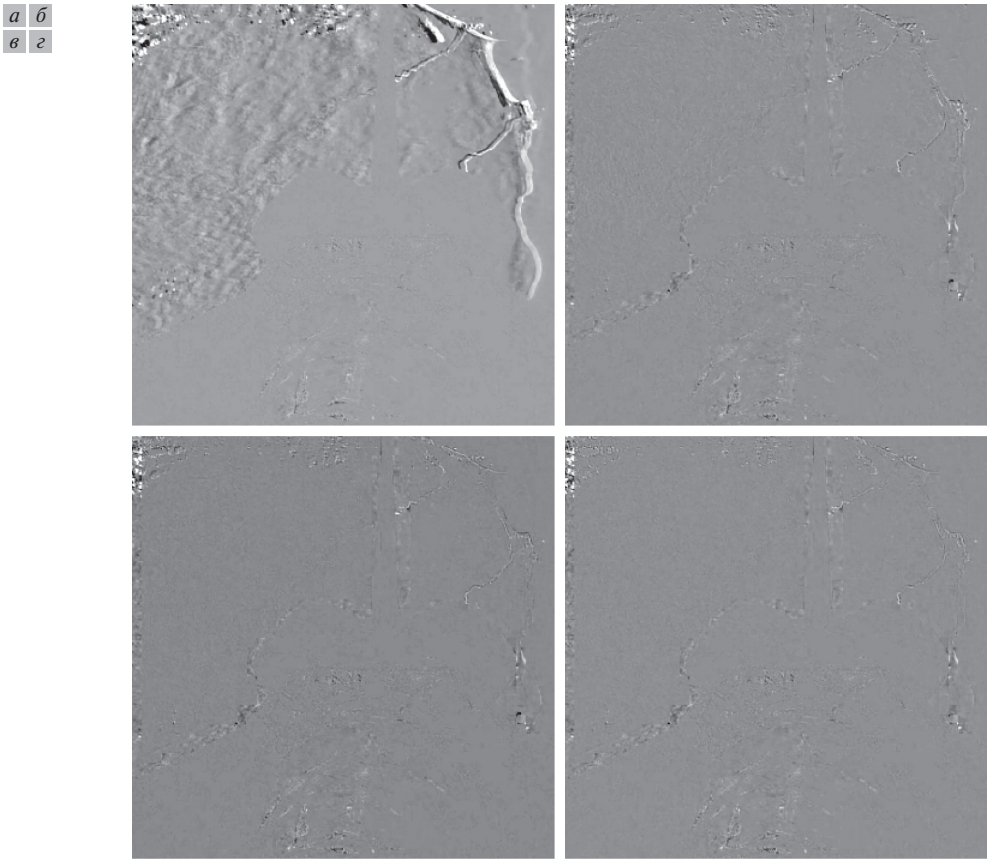


Рис. 8.38. Субпиксельная точность компенсации движения. (*а*) Без компенсации движения; (*б*) точность 1 пиксель; (*в*) точность ½ пикселя; (*г*) точность ¼ пикселя. (Все изображения ошибок предсказания подверглись градационной коррекции на весь диапазон яркостей, после чего контраст был повышен еще в 2 раза)

сказания на рис. 8.38(*а*)—(*г*) уменьшаются с повышением точности вектора предсказания: 12,7, 4,4, 4,0 и 3,8 градаций яркости соответственно. Энтропии ошибок, согласно (8.1-7), равны 4,17, 3,34, 3,35 и 3,34 бит/пиксель соответственно. Таким образом, ошибки предсказания движения содержат приблизительно одинаковое количество информации, несмотря на тот факт, что ошибки на рис. 8.38(*в*) и (*г*) используют дополнительные биты для обеспечения точности интерполяции в ½ и ¼ пикселя. Наконец, отметим, что на всех изображениях ошибок предсказания движения, слева имеется очевидная полоса больших значений. Она возникла вследствие движения Земли слева направо, что вызвало появление новых или ранее невидимых участков земной поверхности в левой части изображения. Поскольку эти области отсутствуют на предыдущем кадре, они не могут быть точно предсказаны, несмотря на точность, используемую для вычисления векторов движения.

Оценка движения является задачей, требующей существенных вычислительных затрат. К счастью, оценивать смещение макроблока приходится лишь

кодеру. Декодер, имея уже заданные векторы движения макроблоков, просто обращается к соответствующей части опорного кадра, той же, которая использовалась кодером для формирования ошибки предсказания. По этой причине оценка движения не включена в большинство стандартов сжатия видео. Стандарты сжатия ориентированы на декодер — задавая ограничения на размеры макроблоков, точность векторов движения, диапазоны вертикального и горизонтального смещения и тому подобное. Ключевые параметры кодирования с предсказанием для некоторых наиболее важных стандартов сжатия видео приведены в табл. 8.11. Отметим, что большинство из этих стандартов для кодирования I-кадров используют ДКП по блокам 8×8 пикселей, но для компенсации движения задают размер области бóльшим (макроблоки 16×16). Вдобавок, благодаря эффективности квантования коэффициентов ДКП, трансформационное кодирование используется даже для предсказания ошибок P- и B-кадров. Наконец заметим, что стандарты H.264 и MPEG-4 AVC для сокращения пространственной избыточности предусматривают кодирование I-кадров с внутрикадровым предсказанием.

На рис. 8.39 представлена блок-схема типичного видеокодера с компенсацией движения. Он использует внутрикадровые и межкадровые избыточности,

Таблица 8.11. Параметры кодирования с предсказанием в стандартах сжатия видео

Параметр	H.261	MPEG-1	H.262 MPEG-2	H.263	MPEG-4	VC-1 WMV-9	H.264 MPEG-4 AVC
Точность вектора предсказания движения	1	½	½	½	¼	¼	¼
Размер макроблока	16×16	16×16	16×16 16×8	16×16 8×8	16×16 8×8	16×16 8×8	16×16 16×8 8×16 8×8 8×4 4×8 4×4
Преобразование	8×8 ДКП	8×8 ДКП	8×8 ДКП	8×8 ДКП	8×8 ДКП	8×8 8×4 4×8 4×4 Целочисленное ДКП	8×8 4×4 Целочисленное
Межкадровые предсказания	P	P, B	P, B	P, B	P, B	P, B	P, B
Внутрикадровое предсказание в I-кадре	Нет	Нет	Нет	Нет	Нет	Нет	Да

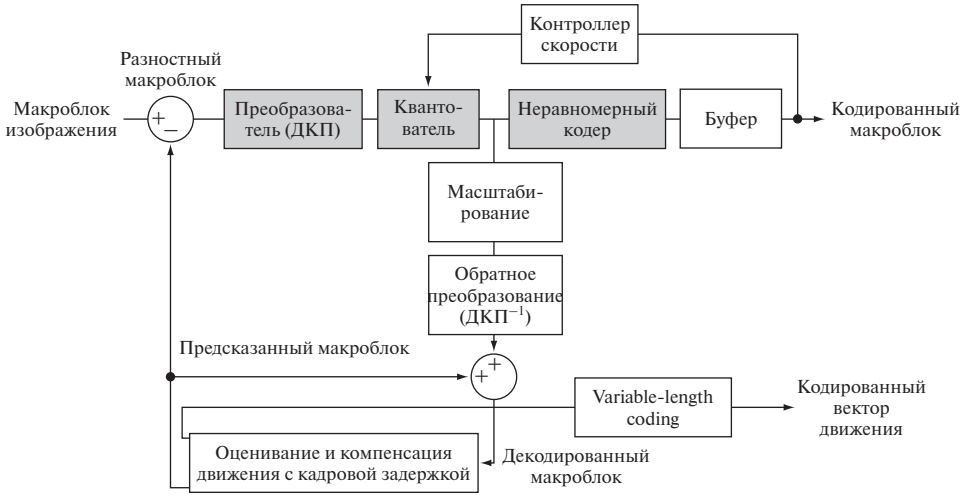


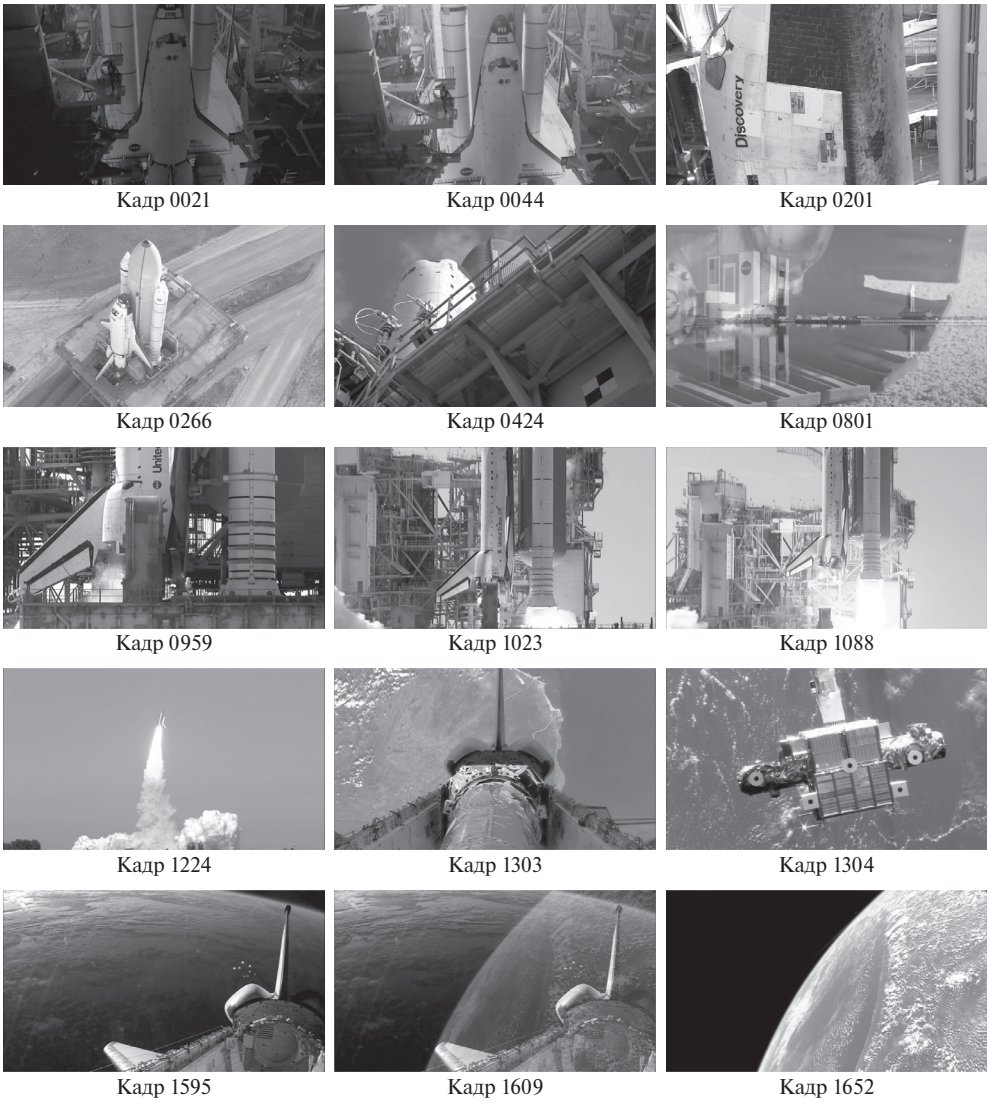
Рис. 8.39. Блок-схема типичного кодера с компенсацией движения

плавность движений между кадрами и психовизуальные свойства зрительной системы человека. Можно считать, что на вход кодера поступает последовательность макроблоков видеосигнала. В случае цветного видеосигнала каждый макроблок состоит из блока яркости и двух блоков цветности. Поскольку острота цветного зрения значительно ниже остроты черно-белого зрения, данные в блоках цветности часто дискретизуются с двукратным уменьшением разрешения по сравнению с данными в блоках яркости. Блоки, показанные серым, аналогичны операциям трансформации, квантования и неравномерного кодирования JPEG-кодера. Принципиальным отличием является то, что на их вход может подаваться как обычный макроблок изображения (в случае I-кадров), так и разность между макроблоком и его предсказанием по предыдущему и/или последующему кадру (в случае P- или B-кадров). Кодер включает блоки *масштабирования* (обратного квантования) и *обратного преобразования* (т. е. обратного ДКП), так что формируемое предсказание совпадает с предсказанием этого же блока во взаимодействующем декодере. Поток данных, формируемый на выходе кодера, не должен превышать пропускной способности предполагаемого видеоканала. Поэтому в схему кодера включен блок *контроллера скорости* как функции заполнения выходного *буфера*, который управляет параметрами квантователя. При сильном заполнении буфера квантование становится более грубым, в результате чего в буфер поступает уменьшенный битовый поток.

> Как уже говорилось ранее, квантование является необратимой операцией. Операция масштабирования («обратного квантования») не предотвращает потери информации.

Пример 8.22. Пример сжатия видеопоследовательности.

■ В завершение обсуждения вопросов кодирования с предсказанием и компенсацией движения рассмотрим пример, иллюстрирующий вариант кодирования, возможный при современных методах сжатия видео. На рис. 8.40 пока-



Кадр 0021 Кадр 0044 Кадр 0201

Кадр 0266 Кадр 0424 Кадр 0801

Кадр 0959 Кадр 1023 Кадр 1088

Кадр 1224 Кадр 1303 Кадр 1304

Кадр 1595 Кадр 1609 Кадр 1652

Рис. 8.40. 15 кадров из одноминутной видеопоследовательности NASA, состоящей из 1829 кадров в натуральных цветах (HD, 1280×720). (Предоставлено NASA)

заны 15 кадров одноминутной видеопоследовательности NASA (HD, 1280×720, натуральные цвета), некоторые части которой уже использовались в данном разделе. Хотя представленные изображения черно-белые, видеопоследовательность состоит из 1829 кадров в натуральных цветах. Отметим, что в ней имеется множество разных сцен, много движения и многочисленные видеоэффекты вытеснения сюжета. Так, например, видео начинается со 150-кадрового проявления сюжета из черноты, которое захватывает кадры 21 и 44, а завершается набором вытесняющих друг друга сюжетов, включающих кадры 1595, 1609 и 1652 на рис. 8.40, и полным затуханием изображения до черноты. Также там имеется

несколько случаев резкой смены сцены, как например на последовательных кадрах 1303 и 1304.

Видеопоследовательность NASA, использованную в данном разделе, можно найти на интернет-сайте книги.

Видеопоследовательность NASA в формате H.264, записанная в виде файла QuickTime (см. табл. 8.4), занимает 44,56 Мб плюс еще 1,39 Мб для сопровождающего звука. Качество видео можно назвать отличным. Для записи несжатых видеокадров в виде изображений в натуральных цветах потребовалось бы около 5 Гб. Следует заметить, что видеопоследовательность содержит также участки зуммирования и поворота (например последовательность, включающая кадры 959, 1023 и 1088 на рис. 8.40). Однако обсуждение движения в данном разделе было ограничено лишь сдвигом. ■

Кодирование с предсказанием с потерями

В данном разделе в модель кодирования с предсказанием без потерь, рассмотренную ранее, будет добавлен квантователь, а в контексте пространственного предсказания будет проведен поиск компромисса между точностью восстановления и качеством сжатия. Как видно из рис. 8.41, между кодером символов и точкой, в которой формируется ошибка предсказания, помещается квантователь, который подменяет функцию определения ближайшего целого от величины, получаемой на выходе кодера без ошибок. Он отображает ошибку предсказания в ограниченный набор (квантованных) значений сигнала на выходе $\dot{e}(n)$, величина разности между которыми (т. е. точность квантования) определяет степень сжатия и величину искажения, возникающего в результате такого кодирования.

Для адаптации модели к введению блока квантователя безошибочный кодер на рис. 8.33(*а*) должен быть изменен так, чтобы предсказания, генерируемые кодером и декодером, были идентичными. Как видно на рис. 8.41(*а*), это дости-

а
б

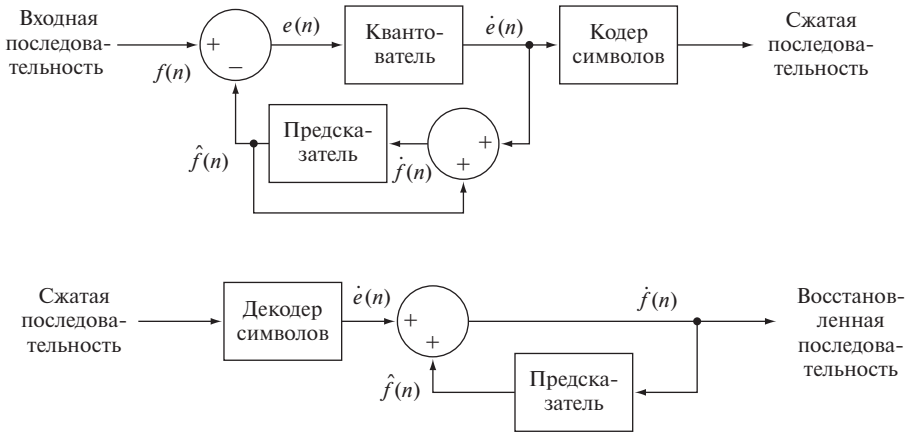


Рис. 8.41. Модель кодирования с предсказанием с потерями: (*а*) блок-схема кодера; (*б*) блок-схема декодера

гается помещением кодера с потерями в цепь обратной связи предсказателя, где его вход, обозначаемый $\dot{f}(n)$, формируется как функция от предыдущего предсказания и текущей ошибки квантования. Таким образом,

$$\dot{f}(n) = \dot{e}(n) + \hat{f}(n), \tag{8.2-38}$$

где $\hat{f}(n)$ та же, что была определена ранее. Схема с обратной связью предотвращает накопление ошибки на выходе кодера. Как видно из рис. 8.41(*б*), выход декодера также задается формулой (8.2-38).

Пример 8.23. Дельта-модуляция.

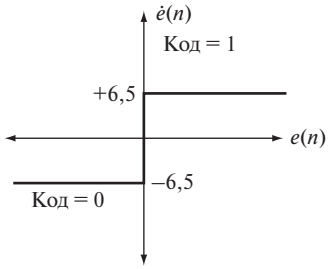
■ Дельта-модуляция (ДМ) является простым, но хорошо известным способом кодирования с потерями, в котором предсказатель и квантователь определяются следующим образом:

$$\hat{f}(n) = \alpha \dot{f}(n-1) \tag{8.2-39}$$

и

$$\dot{e}(n) = \begin{cases} +\zeta & \text{для } e(n) > 0, \\ -\zeta & \text{в остальных случаях,} \end{cases} \tag{8.2-40}$$

где α — коэффициент предсказания (обычно меньше 1), а ζ — положительная константа. Выход квантователя $\dot{e}(n)$ может быть представлен единственным битом (см. рис. 8.42(*а*)), так что кодер символов на рис. 8.41(*а*) может использо-



а б
в

Вход		Кодер				Декодер		Ошибка
n	$f(n)$	$\hat{f}(n)$	$e(n)$	$\dot{e}(n)$	$\dot{f}(n)$	$\hat{f}(n)$	$\dot{f}(n)$	$f(n) - \dot{f}(n)$
0	14	—	—	—	14,0	—	14,0	0,0
1	15	14,0	1,0	6,5	20,5	14,0	20,5	–5,5
2	14	20,5	–6,5	–6,5	14,0	20,5	14,0	0,0
3	15	14,0	1,0	6,5	20,5	14,0	20,5	–5,5
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
14	29	20,5	8,5	6,5	27,0	20,5	27,0	2,0
15	37	27,0	10,0	6,5	33,5	27,0	33,5	3,5
16	47	33,5	13,5	6,5	40,0	33,5	40,0	7,0
17	62	40,0	22,0	6,5	46,5	40,0	46,5	15,5
18	75	46,5	28,5	6,5	53,0	46,5	53,0	22,0
19	77	53,0	24,0	6,5	59,6	53,0	59,6	17,5
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.

Рис. 8.42. Пример дельта-модуляции

вать 1-битовый равномерный код. Результирующая скорость ДМ-кода составит 1 бит/пиксель.

На рис. 8.42(*в*) иллюстрируется механизм дельта-модуляции; в таблице приведены значения сигналов при сжатии и восстановлении следующей входной последовательности: {14, 15, 14, 15, 13, 15, 15, 14, 20, 26, 27, 28, 27, 27, 29, 37, 47, 62, 75, 77, 78, 79, 80, 81, 81, 82, 82} при $\alpha = 1$ и $\zeta = 6{,}5$. Процесс начинается с передачи неискаженного значения первого входного отсчета декодеру. При начальных условиях $\dot{f}_0 = f_0 = 14$, установленных как на стороне кодера, так и на стороне декодера, остальные значения на выходе могут быть определены повторными вычислениями по формулам (8.2-39), (8.2-30), (8.2-40) и (8.2-38). Так, при $n = 1$ получим: $f(1) = 1 \cdot 14 = 14$, $e(1) = 15 - 14 = 1$, $\dot{e}(1) = +6{,}5$ (поскольку $e(1) > 0$), $\dot{f}(1) = 6{,}5 + 14 = 20{,}5$, и результирующая ошибка восстановления составит $(15 - 20{,}5)$ или $-5{,}5$ уровней яркости.

На рис. 8.42(*б*) графически показаны данные, представленные в таблице на рис. 8.42(*в*). Здесь показан входной сигнал $f(n)$ и выход декодера $\dot{f}(n)$. Заметим, что на участке быстрого изменения входного сигнала от $n = 14$ до 19, где ζ оказывается слишком малым для отслеживания больших изменений на входе, возникает искажение, называемое перегрузкой по крутизне. Кроме того, на участке от $n = 0$ до 7, где ζ оказывается слишком велико для отражения малых изменений на входе или относительно плавных участков, возникает шум квантования. На большинстве изображений эти два явления приводят к размыванию контуров и появлению областей с высокой зернистостью или шумом (т. е. к искажению гладких объектов). ■

Искажения, отмеченные в предыдущем примере, являются общими для всех видов кодирования с предсказанием с потерями. Насколько велики эти искажения, зависит от выбранных методов квантования и предсказания в целом. Несмотря на взаимосвязи между ними, обычно предсказатель рассчитывается в предположении отсутствия ошибок квантования, а квантователь — исходя из условия минимизации своих собственных ошибок. Таким образом, предсказатель и квантователь рассчитываются независимо друг от друга.

Оптимальные предсказатели

В большинстве применяемых кодеров с предсказанием предсказатель выбирается так, чтобы минимизировать средний квадрат ошибки предсказания кодера:

$$E\{e^2(n)\} = E\{[f(n) - \hat{f}(n)]^2\}, \tag{8.2-41}$$

где $E\{\cdot\}$ — математическое ожидание, при условии, что

$$\dot{f}(n) = \dot{e}(n) + \hat{f}(n) \approx e(n) + \hat{f}(n) = f(n) \tag{8.2-42}$$

и

$$\hat{f}(n) = \sum_{i=1}^{m} \alpha_i f(n-i). \tag{8.2-43}$$

Таким образом, критерий оптимизации выбирается так, чтобы минимизировать средний квадрат ошибки предсказания, ошибка квантования считается пренебрежимо малой ($\dot{e}(n) \approx e(n)$), а предсказание представляет собой линейную

комбинацию значений m предыдущих отсчетов[23]. Эти ограничения не являются строго необходимыми, но они значительно упрощают анализ и в то же время позволяют уменьшить вычислительную сложность предсказателя. Получаемый в результате метод кодирования с предсказанием известен как *дифференциальная импульсно-кодовая модуляция* (ДИКМ).

При таких условиях проблема построения оптимального предсказателя сводится к относительно простой задаче выбора m коэффициентов предсказателя, которые будут минимизировать следующее выражение:

$$E\{e^2(n)\} = E\left\{\left[f(n) - \sum_{i=1}^{m} \alpha_i f(n-i)\right]^2\right\}. \tag{8.2-44}$$

Дифференцируя уравнение (8.2-44) по каждому из коэффициентов, приравнивая значения производных к нулю и решая получающуюся систему уравнений при условии, что $f(n)$ имеет нулевое среднее и дисперсию σ^2, получим

$$\boldsymbol{\alpha} = \mathbf{R}^{-1}\mathbf{r}, \tag{8.2-45}$$

где $\mathbf{R}^{-1}$ является обратной матрицей следующей матрицы автокорреляции размерами $m \times m$:

$$\mathbf{R} = \begin{bmatrix} E\{f(n-1)f(n-1)\} & E\{f(n-1)f(n-2)\} & \dots & E\{f(n-1)f(n-m)\} \\ E\{f(n-2)f(n-1)\} & \vdots & \dots & \vdots \\ \vdots & \vdots & \dots & \vdots \\ E\{f(n-m)f(n-1)\} & E\{f(n-m)f(n-2)\} & \dots & E\{f(n-m)f(n-m)\} \end{bmatrix}, \tag{8.2-46}$$

а $\mathbf{r}$ и $\boldsymbol{\alpha}$ являются m-элементными векторами:

$$\mathbf{r} = \begin{bmatrix} E\{f(n)f(n-1)\} \\ E\{f(n)f(n-2)\} \\ \vdots \\ E\{f(n)f(n-m)\} \end{bmatrix} \quad \text{и} \quad \boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{bmatrix}. \tag{8.2-47}$$

Таким образом, для любой входной последовательности коэффициенты, которые минимизируют (8.2-44), могут быть определены с помощью последовательности элементарных матричных операций. Более того, коэффициенты зависят лишь от значений автокорреляций отсчетов исходной последовательности. Дисперсия ошибки предсказания, возникающая при использовании этих оптимальных коэффициентов, будет равна

$$\sigma_e^2 = \sigma^2 - \boldsymbol{\alpha}^T\mathbf{r} = \sigma^2 - \sum_{i=1}^{m} E\{f(n)f(n-i)\}\alpha_i. \tag{8.2-48}$$

Хотя основанный на уравнении (8.2-45) способ вычислений достаточно прост, практическое вычисление значений автокорреляций, необходимых для формирования $\mathbf{R}$ и $\mathbf{r}$, настолько затруднительно, что *частные* предсказания (те, в которых коэффициенты предсказания вычисляются индивидуально для каждой входной последовательности) почти никогда не применяются. В большинстве случаев выбирается набор *общих* коэффициентов, вычисляемый пу-

[23] Вообще говоря, для не гауссова сигнала оптимальный предсказатель является нелинейной функцией значений элементов, участвующих в формировании оценки.

тем оценивания некоторой простой модели входного сигнала и подстановки соответствующих значений автокорреляции в (8.2-46) и (8.2-47). Так, в случае если предполагаются двумерный марковский источник изображения (см. раздел 8.1.4) с разделимой автокорреляционной функцией

$$E\{f(x,y)f(x-i,y-j)\} = \sigma^2 \rho_v^i \rho_h^j \tag{8.2-49}$$

и обобщенный линейный предсказатель четвертого порядка

$$\hat{f}(x,y) = \alpha_1 f(x,y-1) + \alpha_2 f(x-1,y-1) + \alpha_3 f(x-1,y) + \alpha_4 f(x-1,y+1), \tag{8.2-50}$$

то результирующие оптимальные коэффициенты [Jain, 1989] будут равны

$$\alpha_1 = \rho_h \quad \alpha_2 = -\rho_v \rho_h \quad \alpha_3 = \rho_v \quad \alpha_4 = 0, \tag{8.2-51}$$

где ρ_h — горизонтальный, а ρ_v — вертикальный коэффициенты корреляции рассматриваемого изображения.

Наконец, обычно требуется, чтобы сумма коэффициентов предсказания в (8.2-43) была меньше или равна единице, т. е.

$$\sum_{i=1}^{m} \alpha_i \le 1. \tag{8.2-52}$$

Это ограничение накладывается для того, чтобы гарантировать, что значение на выходе предсказателя будет оставаться внутри допустимого диапазона входных значений, а также чтобы уменьшить влияние шумов передачи, воздействие которых обычно проявляется на восстановленном изображении в виде горизонтальных полос, если входом блок-схемы на рис. 8.41 является изображение. Важно также уменьшить чувствительность ДИКМ-декодера по отношению к входному шуму, потому что одиночная ошибка (при неблагоприятных условиях) может распространяться на весь последующий выход. Это означает, что выход декодера может оказаться неустойчивым. Введение дополнительного ограничения в (8.2-52) — что сумма должна быть строго меньше единицы — позволяет уменьшить продолжительность влияния шума на входе декодера до нескольких выходных значений.

Пример 8.24. Сравнение методов предсказания.

■ Рассмотрим ошибки предсказания, возникающие при ДИКМ-кодировании полутонового изображения на рис. 8.9(*а*), в предположении нулевой ошибки квантования и при использовании каждого из следующих четырех предсказателей:

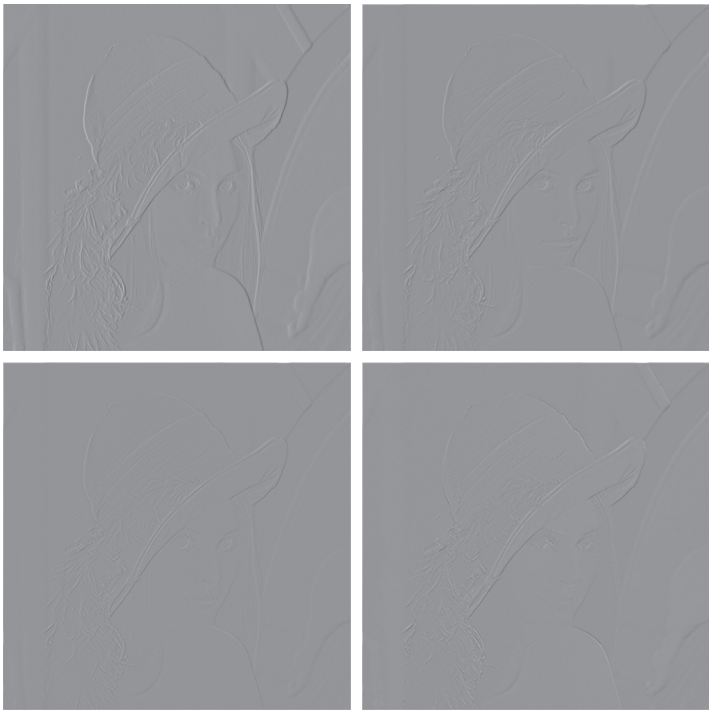
$$\hat{f}(x,y) = 0{,}97 f(x,y-1); \tag{8.2-53}$$

$$\hat{f}(x,y) = 0{,}5 f(x,y-1) + 0{,}5 f(x-1,y); \tag{8.2-54}$$

$$\hat{f}(x,y) = 0{,}75 f(x,y-1) + 0{,}75 f(x-1,y) - 0{,}5 f(x-1,y-1); \tag{8.2-55}$$

$$\hat{f}(x,y) = \begin{cases} 0{,}97 f(x,y-1) & \text{если } \Delta h \le \Delta v; \\ 0{,}97 f(x-1,y) & \text{в остальных случаях,} \end{cases} \tag{8.2-56}$$

где $\Delta h = |f(x-1, y) - f(x-1, y-1)|$ и $\Delta v = |f(x, y-1) - f(x-1, y-1)|$ означают вертикальный и горизонтальный градиенты в точках предсказания. В формулах



а б
в г

Рис. 8.43. Сравнение четырех методов линейного предсказания

(8.2-53)—(8.2-56) задан достаточно устойчивый набор коэффициентов α_i, обеспечивающий удовлетворительные характеристики в широком диапазоне изображений. Адаптивный предсказатель в (8.2-56) предназначен для улучшения передачи контуров. Он измеряет локальные характеристики изображения по направлениям (Δh и Δv) и выбирает предсказатель, соответствующий измеренной оценке.

На рис. 8.43(*а*)—(*г*) в виде изображений показаны ошибки предсказаний, возникающие при использовании формул (8.2-53)—(8.2-56). Как видно, заметность ошибки уменьшается с увеличением порядка предсказания[24]. Стандартные отклонения ошибок предсказания дают близкие результаты: они равны соответственно 11,1, 9,8, 9,1 и 9,7 уровня яркости. ■

Оптимальное квантование

Ступенчатая функция квантования $t = q(s)$, показанная на рис. 8.44, является нечетной функцией s, т. е. $q(-s) = -q(s)$; таким образом, она полностью задается набором из $L/2$ пар значений s_i и t_i для первого квадранта на графике. Эти точки разрывов задают скачки функции и называются *пороговыми уровнями* (s_i) и *уровнями квантования* (t_i) квантователя. Обычно полагают, что входное значение s отображается в уровень квантования t_i, если s находится в полуинтервале $[s_i, s_{i+1})$ при $s_i \geq 0$, и, соответственно, $(s_i, s_{i+1}]$ при $s_i < 0$.

[24] Использование в предсказателях более трех или четырех предыдущих пикселей при существенном усложнении предсказателя дает незначительное улучшение характеристик сжатия [Habibi, 1971].

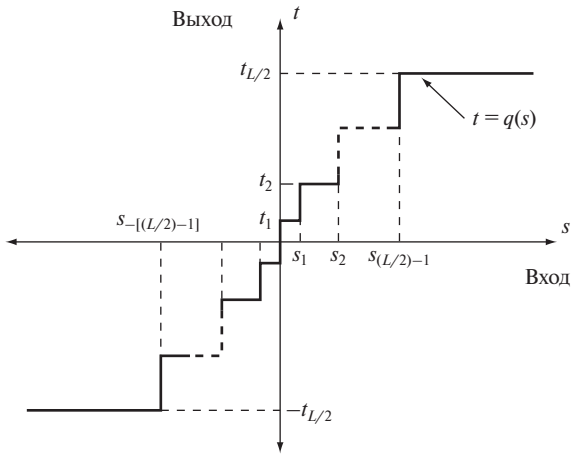


Рис. 8.44. Типичная функция квантования

Проблема построения квантователя заключается в выборе наилучших значений s_i и t_i для конкретного критерия оптимизации и плотности распределения вероятностей $p(s)$. Если критерием оптимизации, который может быть как статистической, так и визуальной[25] мерой, является минимизация среднего квадрата ошибки квантования (т. е. $E\{(s-t_i)^2\}$), а также если $p(s)$ является четной функцией, условиями минимальной ошибки [Max, 1960] являются:

$$\int_{s_{i-1}}^{s_i} (s-t_i)\,p(s)ds = 0 \qquad (i = 1, 2, \ldots, L/2), \tag{8.2-57}$$

$$s_i = \begin{cases} 0 & i = 0; \\ \dfrac{t_i + t_{i+1}}{2} & i = 1, 2, \ldots, \dfrac{L}{2} - 1; \\ \infty & i = L/2, \end{cases} \tag{8.2-58}$$

и

$$s_{-i} = -s_i \qquad t_{-i} = -t_i. \tag{8.2-59}$$

Уравнение (8.2-57) показывает, что оптимальные уровни квантования являются точками центров тяжестей областей под $p(s)$ по каждому из интервалов квантования, разделенных пороговыми уровнями, а уравнение (8.2-58) — что пороговые уровни должны располагаться посередине между уровнями квантования. Условия (8.2-59) являются следствием того, что q — нечетная функция. Таким образом, для любого L и $p(s)$ такие пары s_i и t_i, для которых выполняются уравнения (8.2-57)—(8.2-59), являются оптимальными в смысле среднеквадратической ошибки. Соответствующий квантователь называется L-уровневым *квантователем Ллойда — Макса*.

В табл. 8.12 приведены пороговые уровни и уровни квантования 2-, 4-, и 8-уровневого квантователя Ллойда — Макса для функции плотности распределения вероятностей Лапласа с единичной дисперсией (8.2.35). Эти значения были получены численным методом [Paez, Glisson, 1972], поскольку получение

[25] Касательно *визуальных* (в оригинале *психовизуальных* — *прим. перев.*) мер см. [Netravali, 1977] и [Limb, Rubinstein, 1978].

точного или явного решения уравнений (8.2-57)—(8.2-59) для большинства нетривиальных $p(s)$ достаточно затруднительно. Три представленных квантователя обеспечивают фиксированные скорости кода, равные соответственно 1, 2 и 3 битам/пиксель. Поскольку табл. 8.12 была построена для распределения с единичной дисперсией, то значения пороговых уровней и уровней квантования для случаев $\sigma \neq 1$ получаются простым умножением табулированных значений на величину стандартного отклонения имеющегося распределения плотности вероятностей. В последнем ряду таблицы приведены размеры шага θ *оптимального равномерного квантователя*, который одновременно удовлетворяет уравнениям (8.2-57)—(8.2-59), *а также* дополнительным ограничениям

$$t_i - t_{i-1} = s_i - s_{i-1} = \theta. \qquad (8.2\text{-}60)$$

Если в кодере с предсказанием с потерями (рис. 8.41(*а*)) используется кодер символов, порождающий неравномерный код, то при одной и той же выходной точности оптимальный равномерный квантователь с шагом размера θ обеспечит более низкую скорость кода (для плотности распределения вероятностей Лапласа), чем равномерно кодированный выход квантователя Ллойда — Макса [O'Neil, 1971].

Как квантователь Ллойда — Макса, так и оптимальный равномерный квантователь не являются адаптивными, и гораздо лучший результат можно получить, адаптируя уровни квантования в соответствии с изменениями локальных характеристик изображения. Теоретически области с плавными изменениями яркости могут квантоваться на более мелкие уровни, тогда как области быстрых изменений — на более грубые. Такой подход одновременно сокращает как шум квантования, так и перегрузку по крутизне, требуя при этом минимального увеличения кодовой скорости. Однако такой компромисс значительно увеличивает сложность квантователя.

8.2.10. Вейвлет-кодирование

Как и методы трансформационного кодирования, рассмотренные в разделе 8.2.8, вейвлет-кодирование основано на той же идее, что коэффициенты преобразования, осуществляющие декорреляцию значений элементов на изображении, могут быть сжаты более эффективно, чем исходные значения пикселей. Если базисные функции преобразования — в данном случае вейвлеты — упа-

Таблица 8.12. Квантователь Ллойда — Макса для плотности распределения вероятностей Лапласа с единичной дисперсией

Уровни	2		4		8	
i	s_i	t_i	s_i	t_i	s_i	t_i
1	∞	0,707	1,102	0,395	0,504	0,222
2			∞	1,810	1,181	0,785
3					2,285	1,576
4					∞	2,994
θ	1,414		1,087		0,731	

ковывают бóльшую часть зрительно важной информации в небольшое число коэффициентов, то оставшиеся коэффициенты могут быть грубо квантованы или обнулены с минимальными искажениями изображения.

Согласно табл. 8.3 и 8.4 вейвлет-кодирование используется в стандарте сжатия JPEG-2000.

На рис. 8.45 показана типичная система вейвлет-кодирования. Чтобы закодировать изображение размерами $2^J \times 2^J$, выбираются анализирующий вейвлет ψ и минимальный уровень разложения ($J - P$), которые используются для вычисления дискретного вейвлет-преобразования. Если вейвлет имеет подходящую масштабирующую функцию φ, то может применяться быстрое вейвлет-преобразование (см. разделы 7.4 и 7.5). В любом случае вычисляемое преобразование трансформирует значительную часть исходного изображения в горизонтальные, вертикальные и диагональные коэффициенты разложения с нулевым средним и распределением, близким распределению Лапласа. Вспомним изображение на рис. 7.1 и чрезвычайно простые статистики его вейвлет-преобразования на рис. 7.10(*a*). Поскольку многие из вычисленных коэффициентов несут малую зрительную информацию, то они могут быть квантованы и сжаты для уменьшения межкоэффициентной и кодовой избыточности. Более того, при квантовании может учитываться корреляция местоположения на P уровнях разложения (межуровневые связи). На последнем шаге кодирования символов может использоваться один или несколько методов кодирования без потерь, таких как кодирование длин серий, коды Хаффмана, арифметическое кодирование или кодирование битовых плоскостей. Декодирование осуществляется инвертированием операций кодирования — за исключением квантования, которое не является обратимой операцией.

Принципиальное отличие системы вейвлет-кодирования на рис. 8.45 от системы трансформационного кодирования на рис. 8.21 состоит в отсутствии этапа формирования блоков подызображения в кодере. Поскольку вейвлет-преобразование эффективно с точки зрения вычислений и одновременно с этим по существу локально (т. е. его базисные функции являются пространственно ограниченными), то не требуется дополнительного разбиения исходного изображения. Как будет видно позже в этом разделе, отсутствие такого шага позволяет избавиться от блоковых искажений, характерных для методов, основанных на ДКП, при высоких коэффициентах сжатия.

Выбор вейвлета

Вейвлеты, выбранные в качестве базиса прямого и обратного преобразований, влияют на все аспекты системы вейвлет-кодирования, включая как структурную схему, так и производительность. От них прямо зависит вычислительная сложность преобразований и косвенным образом — возможности системы в отношении сжатия и восстановления изображений при приемлемом уровне искажений. Если вейвлет-функция, задающая преобразование, имеет сопутствующую масштабирующую функцию, то преобразование может быть реализовано как последовательность операций цифровой фильтрации. При этом длины фильтров равны длинам уточняющих последовательностей для вейвлет-

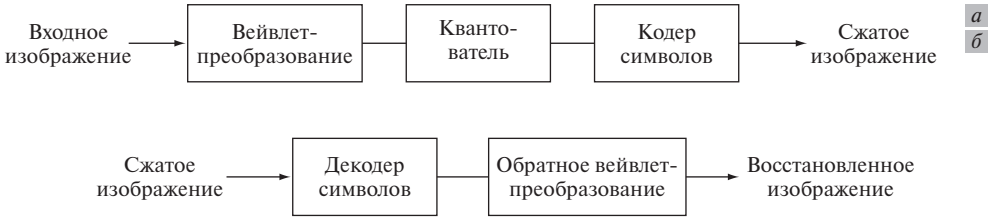


Рис. 8.45. Система вейвлет-кодирования: (*а*) кодер, (*б*) декодер

функции и масштабирующей функции. Характеристики сжатия и восстановления изображений с помощью вейвлет-преобразования определяются возможностями последнего упаковывать информацию в малое число коэффициентов преобразования.

> В цифровой фильтрации каждый из ненулевых коэффициентов фильтров означает умножение его значения на взятый с задержкой фильтруемый сигнал.

Наиболее широко используемыми системами функций разложения, предназначенными для сжатия на основе вейвлет-преобразования, являются системы вейвлетов Добеши и системы биортогональных вейвлетов. Цифровые фильтры, отвечающие последней системе, обладают рядом полезных свойств, такими как гладкость, относительно малая длина и некоторое число нулевых моментов (см. раздел 7.5), которые важны для разложения и последующего восстановления.

Пример 8.25. Вейвлет-базисы в вейвлет-кодировании.

■ Рис. 8.46 содержит четыре дискретных вейвлет-преобразования изображения на рис. 8.9(*а*). Изображение на рис. 8.42(*а*) получено с использованием вейвлетов Хаара в качестве функций разложения (базисных функций), которые являются простейшими и единственными разрывными вейвлетами из рассматриваемых здесь. Изображение на рис. 8.46(*б*) получено с использованием вейвлетов Добеши, относящихся к числу вейвлетов, наиболее популярных в обработке изображений. Изображение на рис. 8.46(*в*) — с использованием симлетов (симметричных вейвлетов), являющихся расширением вейвлетов Добеши с дополнительной симметрией. Для изображения на рис. 8.46(*г*) применялись вейвлеты Коэна — Добеши — Фово, включенные сюда для иллюстрации возможностей биортогональных вейвлетов. Как и ранее на аналогичных примерах, чтобы повысить наглядность возникающих структур, все изображения коэффициентов были подвергнуты масштабирующему градационному преобразованию, при котором значению 0 соответствует уровень яркости 128.

> Детальные коэффициенты ДВП рассматриваются в разделе 7.3.2.

Как можно видеть из табл. 8.13, для преобразований, представленных на рис. 8.46(*а*)—(*г*), число требуемых операций умножения и сложения увеличивается с 4 до 28 на коэффициент (для каждого уровня разложения). Все четыре преобразования выполнялись с использованием быстрого вейвлет-преобразования (т. е. банка фильтров). Заметим, что способность к упаковке

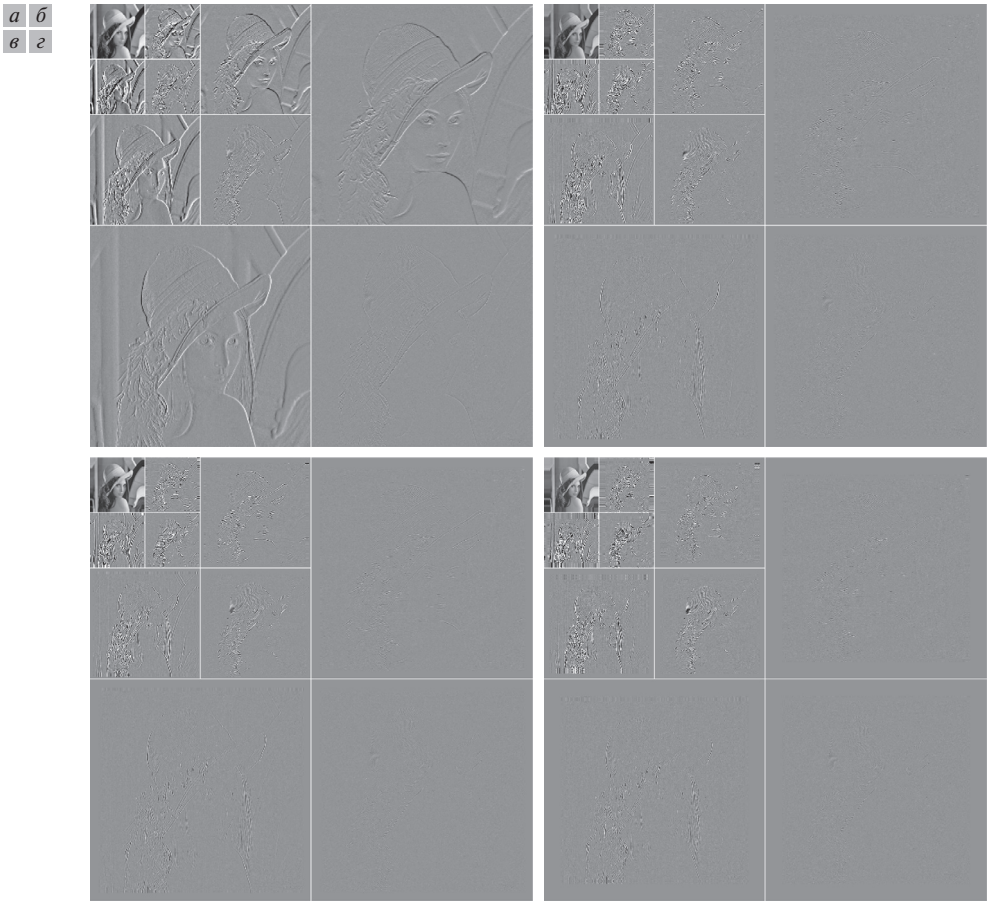


Рис. 8.46. Трехмасштабное вейвлет-преобразование изображения рис. 8.9(*а*) с использованием: (*а*) вейвлетов Хаара, (*б*) вейвлетов Добеши, (*в*) симлетов и (*г*) биортогональных вейвлетов Коэна — Добеши — Фово

информации возрастает с увеличением вычислительной сложности (т. е. числа ненулевых коэффициентов фильтров). При использовании вейвлетов Хаара и усечении коэффициентов деталей на уровне значения 1,5 обнуляются 33,8 % коэффициентов преобразования. При использовании более сложных биортогональных вейвлетов число обнуляемых коэффициентов вырастает до 42,1 %, увеличивая при этом потенциал сжатия на 10 %. ■

Выбор уровня разложения

Другим важным фактором, влияющим на вычислительную сложность и уровень ошибок восстановления вейвлет-кодирования, является число уровней разложения преобразования. Поскольку P-масштабное быстрое вейвлет-преобразование требует P итераций банка фильтров, число операций при вычислении прямого и обратного преобразований возрастает с увеличением числа уровней разложения. Более того, квантование коэффициентов уменьшающихся масштабов, требующее увеличения числа уровней разложения, влияет на все

Таблица 8.13. Длины (количество ненулевых коэффициентов) фильтров вейвлет-преобразования и доля обнуляемых коэффициентов при усечении преобразований на рис. 8.46 на уровне 1.5

Система вейвлетов	Длины фильтров (масштабирующий + вейвлет)	Обнуляемые коэффициенты
Хаара (см. пример 7.10)	2+2	33,8 %
Добеши (см. рис. 7.8)	8+8	40,9 %
Симлет (см. рис. 7.26)	8+8	41,2 %
Биортогональные (см. рис. 7.39)	17+11	42,1 %

бóльшую область восстанавливаемого изображения. Во многих приложениях, таких как поиск по базе данных изображений или передача изображений для постепенного восстановления, число уровней преобразования определяется разрешением хранимых или передаваемых изображений, а также масштабом наименьшей из используемых копий.

Пример 8.26. Уровни разложения в вейвлет-кодировании.

■ Табл. 8.14 иллюстрирует влияние выбираемого уровня разложения на результат кодирования изображения рис. 8.9(*a*) с использованием биортогональных вейвлетов и фиксированного глобального порога, равного 25. Как и в предыдущих примерах вейвлет-кодирования, усечению подвергались лишь детальные коэффициенты. В таблице приведены как доли обнуляемых коэффициентов, так и результирующие СКО ошибок восстановления (согласно выражению (8.1-10)). Заметим, что основное сжатие происходит при начальных разложениях. При увеличении уровней разложения больше трех число обнуляемых коэффициентов меняется мало. ■

Расчет квантователя

Важнейшим фактором, влияющим на коэффициент сжатия и точность восстановления изображения в вейвлет-кодировании, является квантование коэффициентов. Хотя чаще всего для разных уровней применяются одинаковые квантователи, эффективность квантования может быть заметно повышена одним из следующих способов: (1) введением увеличенного интервала квантова-

Таблица 8.14. Влияние уровней разложения на вейвлет-кодирование изображения размерами 512×512, представленного на рис. 8.9(*a*)

Уровень разложения (масштабы и итерации банка фильтров)	Размеры изображения коэффициентов приближения	Доля обнуляемых коэффициентов (%)	СКО ошибки восстановления
1	256×256	74,7 %	3,27
2	128×128	91,7 %	4,23
3	64×64	95,1 %	4,54
4	32×32	95,6 %	4,61
5	16×16	95,5 %	4,63

ния вокруг нуля, называемого *«мертвой зоной»*, или (2) согласованием размеров интервалов квантования с масштабом. В любом случае выбранные интервалы квантования должны быть переданы декодеру вместе с кодированным потоком данных. Сами интервалы могут определяться как эвристически, так и автоматически на основе анализа сжимаемого изображения. Например, глобальный порог для коэффициентов может вычисляться как медиана модулей детальных коэффициентов первого уровня или же как функция числа отбрасываемых нулей и количества энергии, которое остается в восстанавливаемом изображении.

> В качестве одной из мер энергии дискретного сигнала может служить сумма квадратов его отсчетов.

Пример 8.27. Выбор интервала «мертвой зоны» в вейвлет-кодировании.

■ На рис. 8.47 иллюстрируется влияние размера интервала «мертвой зоны» на долю усеченных детальных коэффициентов для трехуровневого вейвлет-кодирования изображения на рис. 8.9(*a*) с использованием биортогональных вейвлетов. С увеличением размера «мертвой зоны» число обнуляемых коэффициентов также возрастает. Выше излома кривой (т. е. больше 5) прирост мал; это является следствием того, что гистограмма детальных коэффициентов имеет ярко выраженный пик вблизи нуля (см., например, рис. 7.10).

Стандартные отклонения ошибок восстановления, соответствующие порогу «мертвой зоны» на рис. 8.47, возрастают от 0 (при нулевом пороге) до 1,94

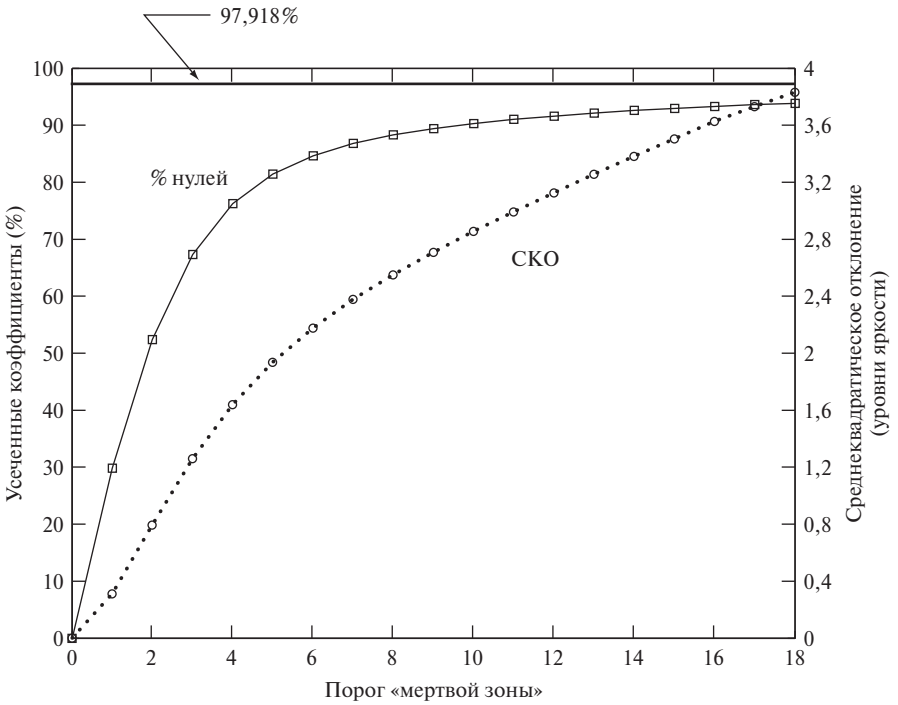


Рис. 8.47. Влияние размера интервала «мертвой зоны» в вейвлет-кодировании

уровня яркости при пороге 5 и до 3,83 при пороге 18, где число нулей достигает уже 93,85 %. Если удалить все детальные коэффициенты, это увеличит долю нулей приблизительно до 97,92 % (около 4 %), но ошибка восстановления возрастет до 12,3 уровней яркостей. ■

JPEG-2000

Стандарт JPEG-2000 расширяет популярный стандарт JPEG, предоставляя бóльшую гибкость, как при сжатии полутоновых неподвижных изображений, так и при доступе к самим сжатым данным. Так, например, отдельные части изображения, сжатого по стандарту JPEG-2000, могут быть извлечены для передачи, хранения, воспроизведения или редактирования. Сжатие по стандарту JPEG-2000 основано на только что рассмотренных методах вейвлет-кодирования. Квантование коэффициентов осуществляется по-разному в разных масштабах и диапазонах (субполосах), а сами квантованные коэффициенты кодируются арифметическим кодом как битовые плоскости (см. разделы 8.2.3 и 8.2.7). Согласно определениям стандарта [ISO/IEC, 2000] процедура кодирования изображения состоит в следующем.

> Необратимое преобразование компонент используется в сжатии с потерями. Само по себе преобразование компонент не является необратимым. Для обратимого сжатия используется другое преобразование компонент.

Первым шагом процедуры кодирования является сдвиг значения среднего уровня яркости, осуществляемый вычитанием из неотрицательных значений отсчетов кодируемого изображения величины 2^{n-1}, где n — число битов в элементах изображения (данное значение n используется в стандарте под названием *Ssiz* для определения разрешения по яркости). Если изображение имеет более одной *компоненты* — как например красная, зеленая и синяя составляющие в случае цветного изображения — каждая компонента сдвигается независимо. Если компонент в точности три, то они дополнительно могут быть декоррелированы с помощью линейного преобразования компонент. Согласно стандарту, преобразование компонент может быть *обратимым* (преобразование целых значений в целые, используемое в *обратимом вейвлет-преобразовании* 5-3 *без потерь*)[26] или *необратимым* (преобразование действительных значений в действительные, используемое в *необратимом вейвлет-преобразовании* 9-7 *с потерями*). Так, *необратимое преобразование компонент* состоит в следующем:

$$\begin{aligned} Y_0(x,y) &= 0,299I_0(x,y)+0,587I_1(x,y)+0,144I_2(x,y);\\ Y_1(x,y) &= -0,16875I_0(x,y)-0,33126I_1(x,y)+0,5I_2(x,y);\\ Y_2(x,y) &= 0,5I_0(x,y)-0,41869I_1(x,y)-0,08131I_2(x,y), \end{aligned} \qquad (8.2\text{-}61)$$

где I_0, I_1 и I_2 — сдвинутые влево по оси яркости значения входных компонент, а Y_0, Y_1 и Y_2 — соответственные значения декоррелированных компонент. Если входными компонентами являются красная, зеленая и синяя составляющие цвет-

[26] В обозначении вейвлет-преобразования *a*-*b* число (*a*) означает количество коэффициентов низкочастотного, а (*b*) — высокочастотного фильтров анализа. В стандарте приняты вейвлет-преобразования 5-3 и 9-7. — *Прим. перев.*

ного изображения, то формулы (8.2-61) соответствуют цветовому преобразованию из пространства $R'G'B'$ в пространство $Y'C_bC_r$ (см. [Poynton, 1996])[27]. Цель преобразования состоит в улучшении эффективности сжатия; компоненты Y_1 и Y_2 являются разностными изображениями, гистограммы которых имеют ярко выраженные пики вблизи нуля.

После того, как изображение сдвинуто по уровням яркости и, возможно, декоррелировано, оно может быть разбито на непересекающиеся *блоки — тайлы*[28]. Тайлы представляют собой прямоугольные массивы пикселей и содержат одинаковую относительную долю всех компонент изображения. Тем самым, в процессе разбиения на области создаются *компоненты тайла* (*тайл-компоненты*), которые могут выделяться и восстанавливаться независимо, при условии существования простого механизма для доступа и/или управления этими небольшими областями закодированного изображения. Например, изображение, имеющее соотношение сторон 16:9, может быть разбито на тайлы таким образом, что одним из его тайлов будет подызображение с соотношением сторон 4:3. Этот тайл может потом быть восстановлен без обращения к остальным тайлам сжатого изображения. Если же изображение не разбивается на тайлы, оно представляет собой единственный тайл.

Затем вычисляется одномерное дискретное вейвлет-преобразование по строкам и по столбцам каждой компоненты тайла. Сжатие без потерь (обратимое вейвлет-преобразование 5-3) основано на использовании коэффициентов уточняющих последовательностей для масштабирующей функции и вейвлет-функции системы биортогональных вейвлетов [Le Gall, Tabatabai, 1988]. Для нецелых значений коэффициентов преобразования задается процедура округления. В системах сжатия с потерями (необратимое вейвлет-преобразование 9-7) применяют коэффициенты уточняющих последовательностей для масштабиру-

Таблица 8.15. Импульсные характеристики низкочастотного и высокочастотного фильтров анализа для необратимого вейвлет-преобразования 9-7, применяемого в случае сжатия с потерями

Номера позиций фильтра	Коэффициенты высокочастотного фильтра анализа	Коэффициенты низкочастотного фильтра анализа
0	−1,115087052456994	0,6029490182363579
±1	0,5912717631142470	0,2668641184428723
±2	0,05754352622849957	−0,07822326652898785
±3	−0,09127176311424948	−0,01686411844287495
±4	0	0,02674875741080976

[27] Цветовое пространство $R'G'B'$ есть вариант нелинейной градационной коррекции линейного цветового пространства *RGB*, рекомендованного МКО — Международной комиссией по освещению (Commission Internationale de L'Eclairage, CIE). Y' есть яркость, а C_b и C_r — две цветоразности (т. е. масштабированные значения $B' - Y'$ и $R' - Y'$).

[28] В оригинале — *tiles* («плитка, фрагмент изображения»). Данный раздел изобилует терминами, почерпнутыми из материалов, касающихся стандарта кодирования JPEG-2000. Не для всех из них удалось подобрать удачный русскоязычный термин. По тем или иным причинам часть терминов было решено оставить в англоязычном произношении. — *Прим. ред. перев.*

ющей функции и вейвлет-функции системы вейвлетов, описанной в [Antonini, Barlaud, Mathieu, Daubechies, 1992]. В каждом из случаев преобразование вычисляется посредством быстрого вейвлет-преобразования, рассмотренного в разделе 7.4, или с помощью дополнительной так называемой *лифтинг*[29]*-схемы* [Mallat, 1999]. Коэффициенты, необходимые для построения банка фильтров анализа необратимого быстрого вейвлет-преобразования (БВП) 9-7, приведены в табл. 8.15. Реализация альтернативной лифтинг-схемы требует шести последовательных операций «лифтинга» и «масштабирования»:

$$\begin{aligned}
&Y(2n+1) = X(2n+1) + \alpha[X(2n) + X(2n+2)], && i_0 - 3 \le 2n+1 < i_1 + 3;\\
&Y(2n) = X(2n) + \beta[Y(2n-1) + Y(2n+1)], && i_0 - 2 \le 2n < i_1 + 2;\\
&Y(2n+1) = Y(2n+1) + \gamma[Y(2n) + Y(2n+2)], && i_0 - 1 \le 2n+1 < i_1 + 1;\\
&Y(2n) = Y(2n) + \delta[Y(2n-1) + Y(2n+1)], && i_0 \le 2n < i_1;\\
&Y(2n+1) = -K \cdot Y(2n+1), && i_0 \le 2n+1 < i_1;\\
&Y(2n) = Y(2n) / K, && i_0 \le 2n < i_1 .
\end{aligned} \qquad (8.2\text{-}62)$$

Реализация при помощи лифтинг-схемы является альтернативным способом вычисления вейвлет-преобразований. Коэффициенты, используемые при таком подходе, напрямую связаны с коэффициентами банка фильтров БВП.

Здесь X есть преобразуемая тайл-компонента, Y — результат преобразования, а i_0 и i_1 задают положение тайл-компоненты внутри компоненты полного изображения. То есть они являются индексами первого отсчета преобразуемой строки или столбца тайл-компоненты (i_0) и того отсчета, который следует непосредственно за последним отсчетом (i_1). Переменная n принимает значения в зависимости от значений i_0 и i_1, а также от того, какая из шести операций выполняется. Для $n < i_0$ или $n \ge i_1$ $X(n)$ получается симметричным продолжением X; например, $X(i_0 - 1) = X(i_0 + 1)$, $X(i_0 - 2) = X(i_0 + 2)$, $X(i_1) = X(i_1 - 2)$, $X(i_1 + 1) = X(i_1 - 3)$. По окончании операций лифтинга и масштабирования значения Y с четными индексами будут совпадать с результатами на выходе низкочастотного БВП фильтра анализа, а значения Y с нечетными индексами — с результатами на выходе высокочастотного БВП фильтра анализа. Параметры лифтинга составляют: $\alpha = -1{,}586134342$, $\beta = -0{,}052980118$, $\gamma = 0{,}882911075$, $\delta = 0{,}433506852$, а коэффициент $K = 1{,}230174105$.

Приведенные параметры лифтинг-схемы определены в стандарте.

Только что описанное преобразование дает в результате четыре субполосы — низкочастотное приближение тайл-компоненты, а также ее вертикальные, горизонтальные и диагональные высокочастотные детали. Повторное применение преобразования N_L раз к полученным на предыдущей итерации коэффициентам низкочастотного приближения дает в результате N_L-масштабное вейвлет-преобразование. Пространственное разрешение соседних масштабов различается вдвое, причем самый крупный масштаб содержит наиболее точное приближение исходной тайл-компоненты. Как можно

[29] В оригинале — *lifting*. — *Прим. перев.*

предположить из рис. 8.48, где обозначения стандарта JPEG-2000 приведены для случая $N_L = 2$, N_L-масштабное преобразование общего вида содержит $3N_L + 1$ субполосы, коэффициенты которых обозначаются a_b, где $b = N_L LL$, $N_L HL$,..., $1HL$, $1LH$, $1HH$. Стандарт не определяет число масштабов, которые должны быть вычислены.

> Напомним (согласно главе 7), что ДВП осуществляет разложение изображения на набор узкополосных компонент, называемых субполосами.

Когда компонента тайла обработана, общее число коэффициентов преобразования равно числу отсчетов в исходной тайл-компоненте, однако важная визуальная информация сосредоточена только в небольшом числе коэффициентов. Для уменьшения числа битов, необходимых для представления преобразования, коэффициент $a_b(u, v)$ субполосы b разложения квантуется в величину $q_b(u, v)$ при помощи преобразования

$$q_b(u,v) = \text{sign}(a_b(u,v)) \cdot \left\lfloor \frac{|a_b(u,v)|}{\Delta_b} \right\rfloor, \tag{8.2-63}$$

где sign(·) — знак числа, [·] — целая часть числа, а *шаг квантования* Δ_b составляет

$$\Delta_b = 2^{R_b - \varepsilon_b} \left(1 + \frac{\mu_b}{2^{11}}\right). \tag{8.2-64}$$

Здесь R_b — *номинальный динамический диапазон* субполосы b, а ε_b и μ_b — число битов, отводимых на значения порядка и мантиссы ее коэффициентов. Номинальный динамический диапазон субполосы b есть сумма числа битов, используемых для представления исходного изображения, и числа *дополнительных битов разложения* субполосы b. Число дополнительных битов приведено цифрами в кружочках на рис. 8.48. Так, для субполосы $b = 1HH$ требуется 2 дополнительных бита разложения.

> Не следует путать определенный в стандарте номинальный динамический диапазон с близко связанным определением в главе 2.

В случае сжатия без потерь $\mu_b = 0$, $R_b = \varepsilon_b$, а $\Delta_b = 1$. Для необратимого сжатия в стандарте не указывается никакого конкретного шага квантования. Вместо этого число битов порядка и мантиссы должно передаваться декодеру или вместе с каждой субполосой, и это называется *явным квантованием*, или же только с субполосой $N_L LL$, что называется *неявным квантованием*. В последнем случае остальные субполосы квантуются с использованием значений параметров, вычисленных из параметров субполосы $N_L LL$. Полагая, что ε_0 и μ_0 — число битов, отводимых для субполосы $N_L LL$, параметры для субполосы b вычисляются следующим образом:

$$\begin{aligned} \mu_b &= \mu_0; \\ \varepsilon_b &= \varepsilon_0 + n_b - N_L, \end{aligned} \tag{8.2-65}$$

где n_b обозначает число уровней разложения субполосы от исходной тайл-компоненты изображения до субполосы b.

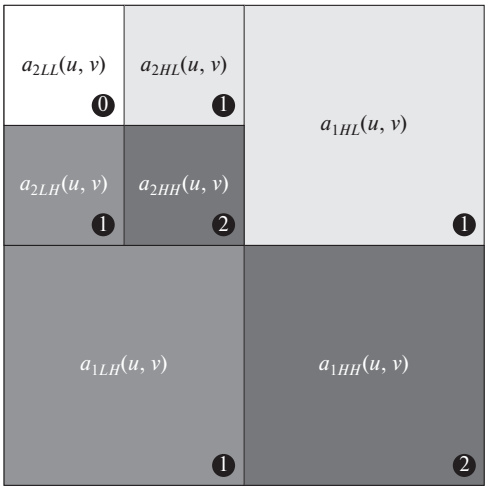


Рис. 8.48. Система обозначений JPEG-2000 для коэффициентов двухмасштабного вейвлет-преобразования тайл-компоненты и число дополнительных битов разложения

На финальных этапах процесса кодирования коэффициенты каждой трансформированной субполосы тайл-компоненты размещаются в прямоугольных блоках, называемых *кодовыми блоками*, которые кодируются независимо по битовым плоскостям. Начиная с самой старшей битовой плоскости с ненулевым элементом, каждая битовая плоскость обрабатывается за три прохода. Каждый бит из битовой плоскости кодируется только на одном из трех проходах, называемых *распространением значащего разряда*, *уточнением значения* и *подчисткой*[30]. Полученные результаты затем арифметически кодируются и группируются вместе с аналогичными проходами других кодовых блоков того же тайла, формируя *слои*[31]. Слой — объединение кодовых блоков тайла одного и того же уровня разложения. Разделение на слои позволяет по отдельности кодировать уровни разложения вейвлет-преобразования, обеспечивая тем самым при декодировании необходимую масштабируемость по пространственному разрешению. Полученные слои в конце концов делятся на *пакеты*, предоставляя дополнительную возможность выделения пространственных областей интереса из общего кодового потока. Пакеты являются основными единицами закодированного потока данных.

В декодере JPEG-2000 описанные выше операции выполняются в обратном порядке. На основании указываемого пользователем интересующего фрагмента изображения и точности его воспроизведения осуществляется восстановление нужных субполос тайл-компонентов из арифметически кодированных JPEG-2000 пакетов. Для этого из общего потока данных выбираются необходимые пакеты нужных слоев, восстанавливается битовый поток, осуществляется арифметическое декодирование и распаковываются биты коэффициентов. Хотя кодер мог закодировать M_b битовых плоскостей конкретной субполосы, пользователь, благодаря особенностям вложенности кодового потока, может выбрать

[30] В оригинале — *significance propagation, magnitude refinement и cleanup. — Прим. перев.*

[31] В оригинале — *layers. — Прим. перев.*

восстановление лишь части N_b битовых плоскостей. Это эквивалентно квантованию коэффициентов кодового блока с шагом размера $2^{M_b-N_b}\Delta_b$. Все нераспакованные биты обнуляются, и результирующие коэффициенты, обозначаемые $\overline{q}_b(u,v)$, восстанавливаются (при помощи масштабирования — операции, обратной квантованию) следующим образом:

> Как уже указывалось ранее, квантование является необратимой операцией. Термин «обратное квантование» не означает, что не происходит потери информации. Информация при этом теряется, за исключением обратимого JPEG-2000 сжатия, при котором $\mu_b = 0$, $R_b = \varepsilon_b$ и $\Delta_b = 1$.

$$R_{q_b}(u,v)=\begin{cases}(\overline{q}_b(u,v)+\boldsymbol{r}\cdot 2^{M_b-N_b(u,v)})\cdot\Delta_b & \text{при } \overline{q}_b(u,v)>0;\\ (\overline{q}_b(u,v)-\boldsymbol{r}\cdot 2^{M_b-N_b(u,v)})\cdot\Delta_b & \text{при } \overline{q}_b(u,v)<0;\\ 0 & \text{при } \overline{q}_b(u,v)=0,\end{cases} \tag{8.2-66}$$

где $R_{q_b}(u,v)$ означает восстановленное значение коэффициента, а $N_b(u,v)$ — число декодируемых битовых плоскостей для $\overline{q}_b(u,v)$. Параметр восстановления r выбирается при декодировании исходя из визуального или объективного качества восстановления. Обычно $0 \le r \le 1$ при общепринятом значении $r = ½$. Восстановленные значения коэффициентов затем подвергаются обратным преобразованиям по столбцам и строкам, используя банк фильтров обратного БВП, коэффициенты которого берутся из табл. 8.15 и уравнения (7.1-11), или с помощью следующей лифтинг-операции:

$$\begin{aligned}
&X(2n)=K\cdot Y(2n), && i_0-3\le 2n<i_1+3\\
&X(2n+1)=(-1/K)\cdot Y(2n+1), && i_0-2\le 2n+1<i_1+2\\
&X(2n)=X(2n)-\delta[X(2n-1)+X(2n+1)], && i_0-3\le 2n<i_1+3\\
&X(2n+1)=X(2n+1)-\gamma[X(2n)+X(2n+2)], && i_0-2\le 2n+1<i_1+2\\
&X(2n)=X(2n)-\beta[X(2n-1)+X(2n+1)], && i_0-1\le 2n<i_1+1\\
&X(2n+1)=X(2n+1)-\alpha[X(2n)+X(2n+2)], && i_0\le 2n+1<i_1.
\end{aligned} \tag{8.2-67}$$

Параметры α, β, γ, δ и K здесь те же, что использовались для уравнений (8.2-62). Если необходимо, осуществляется симметричное продолжение восстановленных значений коэффициентов $Y(n)$ по строкам и столбцам. Финальными операциями декодирования являются сбор тайл-компонент, обратное преобразование компонент (если требуется) и обратный сдвиг значения среднего уровня яркости. В случае необратимого вейвлет-преобразования 9-7 обратное преобразование компонент вычисляется по формулам

$$\begin{aligned}
&I_0(x,y)=Y_0(x,y)+1{,}402Y_2(x,y);\\
&I_1(x,y)=Y_0(x,y)-0{,}34413Y_1(x,y)-0{,}71414Y_2(x,y);\\
&I_2(x,y)=Y_0(x,y)+1{,}772Y_1(x,y),
\end{aligned} \tag{8.2-68}$$

после чего к полученным значениям прибавляется величина 2^{n-1}, где $n = Ssiz$ — число битов в элементах изображения.

Рис. 8.49. Четыре варианта приближения полутонового изображения на рис. 8.9(*a*) методом сжатия JPEG-2000. В каждом ряду приведены результаты сжатия и восстановления, контрастированная разность между результатом и исходным изображением и увеличенный фрагмент восстановленного изображения. (Сравните результаты в первых двух рядах с результатами JPEG на рис. 8.32)

Пример 8.28. Сравнение способов кодирования, основанных на вейвлет-преобразовании (JPEG-2000) и на ДКП (JPEG).

■ На рис. 8.49 представлены четыре варианта приближения полутонового изображения на рис. 8.9(*а*) методом сжатия JPEG-2000. С каждым рядом изображений на рис. 8.49 коэффициент сжатия увеличивается: $C = 25$, 52, 75 и 105. В столбце 1 — восстановленные результаты сжатия JPEG-2000. В столбце 2 представлены контрастированные разности (ошибки) исходного изображения (рис. 8.9(*а*)) и изображений в левом столбце. В столбце 3 — увеличенный фрагмент изображения в столбце 1. Поскольку коэффициенты сжатия в первых двух строках совпадают с коэффициентами JPEG-сжатия в примере 8.18 (рис. 8.32(*а*)—(*е*)), то приводимые результаты являются сравнением — как качественным, так и количественным — вариантов сжатия на основе ДВП и ДКП.

Визуальное сравнение изображений ошибок в первых двух рядах рис. 8.49 с соответствующими изображениями на рис. 8.32(*б*) и (*д*) показывает значительное сокращение ошибок при использовании JPEG-2000 — 3,86 и 5,77 уровней яркости по сравнению с 5,4 и 10,7 уровней яркости для результатов JPEG. Эти оценки показывают преимущество результатов, основанных на вейвлет-кодировании, для обоих уровней сжатия. Заметим отсутствие блоковых искажений, доминирующих на аналогичных фрагментах рис. 8.32(*в*) и (*е*), сжатых JPEG-кодированием. В заключение заметим, что степень сжатия, продемонстрированная в 3 и 4 рядах рис. 8.49, практически недостижима при использовании JPEG-сжатия. JPEG-2000 позволяет использовать изображения, сжатые 100:1 и более; наиболее заметным искажением при этом является увеличение размывания изображения. ■

8.3. Нанесение цифровых водяных знаков на изображение

Методы и стандарты, рассмотренные в разделе 8.2, сделали распространение изображений (как фотографий, так и видео) в цифровых средствах информации и в Интернете совершенно реальным делом. К сожалению, распространяемые таким путем изображения могут многократно и безошибочно копироваться, что чревато нарушением прав собственности. Даже если перед рассылкой изображение было зашифровано, оно становится незащищенным после расшифровки. Одним из способов помешать нелегальному копированию является внесение в потенциально уязвимые изображения одного или нескольких элементов информации, собирательно называемых *водяным знаком*, чтобы они были неотделимы от самого изображения. Будучи неотъемлемой частью *изображения с водяными знаками*, они защищают права собственника различными способами, среди которых:

1. *Идентификация авторского права*. Водяные знаки могут являться информацией, служащей доказательством собственности в случаях, когда права собственника нарушены.
2. *Идентификация пользователя*, или *нанесение «отпечатков пальцев»*. В водяных знаках могут быть закодированы идентификаторы легаль-

ных пользователей, что поможет обнаружению источника нелегальных копий.

3. *Определение подлинности.* Наличие водяного знака может гарантировать, что изображение не было изменено — в предположении, что водяной знак таков, что он будет разрушен при любой модификации изображения.
4. *Автоматический мониторинг.* Водяные знаки могут контролироваться системой, которая прослеживает, когда и где изображения использовались (например, при помощи программ поиска в Интернете изображений, помещенных на веб-страницах). Мониторинг полезен для определения размера авторского гонорара и/или обнаружения незаконного использования.
5. *Защита от копирования.* Водяные знаки могут задавать правила использования и копирования изображений (например, для DVD-проигрывателей).

В нынешнем разделе мы даем обзор вопросов, связанных с нанесением цифровых водяных знаков на изображение — процесса внесения данных в изображение таким путем, что они могут быть использованы для формулировки тех или иных утверждений касательно изображения. Хотя у рассматриваемых методов мало общего с технологиями сжатия, описанными в предыдущих разделах, тем не менее они имеют непосредственное отношение к кодированию информации. Фактически нанесение водяных знаков и сжатие в некотором смысле противоположны. Если целью сжатия является уменьшение количества данных, используемых для его представления, то задача нанесения водяных знаков состоит в добавлении в изображение информации, т. е. некоторых данных (водяных знаков). Как будет показано в оставшейся части раздела, водяные знаки сами по себе могут быть как видимыми, так и невидимыми.

Видимым водяным знаком может быть непрозрачное или полупрозрачное изображение (подызображение), помещаемое впереди другого изображения (т. е. изображения, на которое наносится водяной знак), так что оно становится видимым для наблюдателя. Телевизионные сети часто размещают видимые водяные знаки (в форме их логотипов) в углу телевизионного экрана. Как показано в следующих примерах, добавление видимых водяных знаков осуществляется, как правило, в пространственной области.

Пример 8.29. Простой видимый водяной знак.

■ На рис. 8.50(*б*) показан правый нижний квадрант изображения рис. 8.9(*а*) с уменьшенным и нанесенным на него вариантом водяного знака, представленного на рис. 8.50(*а*). Если через w обозначить изображение водяного знака, а через f — исходное изображение, то изображение с водяным знаком fw можно получить как их линейную комбинацию:

$$f_w = (1-\alpha)f + \alpha w, \tag{8.3-1}$$

где константа α задает относительную видимость водяного знака и лежащего под ним изображения. Если $\alpha = 1$, то водяной знак будет непрозрачным и лежащее под ним изображение будет полностью скрыто. Если α приближается к 0, то исходное изображение будет видно больше, а водяной знак — меньше. Вообще

$0 < \alpha \le 1$; на рис. 8.50(*б*) $\alpha = 0{,}3$. Рис. 8.50(*в*) — вычисленная разность (масштабированная по яркости) между изображением с водяным знаком (*б*) и исходным изображением рис. 8.9(*а*). Значение яркости 128 соответствует нулевой разности. Заметим, что исходное изображение достаточно хорошо видно через «полупрозрачный» водяной знак. Это видно как на рис. 8.50(*б*), так и (*в*). ■

В отличие от видимого водяного знака в предыдущем примере, *невидимые водяные знаки* нельзя увидеть невооруженным глазом. Они незаметны, но могут быть восстановлены при помощи соответствующего алгоритма декодирования. Невидимость обеспечивается тем, что они вносятся как визуально избыточная информация — т. е. информация, которую зрительная система человека игнорирует или не может воспринять (см. раздел 8.1.3). Простой пример показан на рис. 8.51(*а*). Так как самые младшие биты 8-битового изображения практически не влияют на наше восприятие изображения, то водяной знак из рис. 8.50(*а*) был добавлен (или «спрятан») в его два младших бита. Используя уже введенную нотацию, получим

$$f_w = 4\left(\frac{f}{4}\right) + \frac{w}{64} \tag{8.3-2}$$

и воспользуемся целочисленной арифметикой для выполнения вычислений. Деление и умножение на 4 приводит к обнулению двух младших значащих битов f; деление w на 64 сдвигает его на 6 разрядов, переводя два старших бита в позиции двух младших разрядов (MP). Сложение двух полученных результатов дает *изображение с водяным знаком в MP*. Отметим, что нанесенный водяной знак на изображении не виден (рис. 8.51(*а*)). Обнулив 6 старших значащих битов изображения и масштабируя (контрастируя) оставшиеся в младших разрядах

Рис. 8.50. Простой видимый водяной знак: (*а*) водяной знак; (*б*) изображение с водяным знаком; (*в*) разность между изображением с водяным знаком (*б*) и исходным изображением

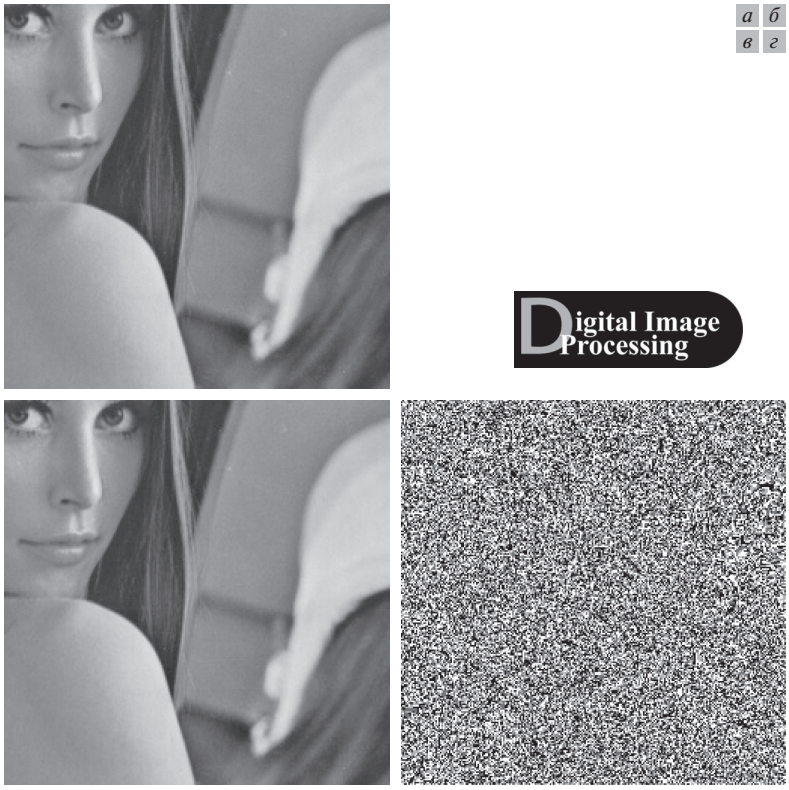


Рис. 8.51. Простой невидимый водяной знак: (*а*) изображение с водяным знаком; (*б*) выделенный водяной знак; (*в*) изображение с водяным знаком после JPEG-сжатия высокого качества и восстановления; (*г*) выделение водяного знака из (*в*)

значения до полного диапазона яркостей, водяной знак может быть выделен, как это показано на рис. 8.51(*б*).

Важным свойством невидимых водяных знаков является их стойкость к случайным или намеренным попыткам устранить их. *Нестойкие невидимые водяные знаки* разрушаются при любых модификациях изображений, в которые они встроены. Однако в некоторых ситуациях это становится желаемой характеристикой, например при необходимости подтверждения их подлинности. Как видно из рис. 8.51(*в*) и (*г*), изображение с невидимым водяным знаком в MP на рис. 8.51(*а*) содержит нестойкий водяной знак. Если изображение (*а*) сжато и восстановлено при помощи JPEG-сжатия с потерями, то водяной знак разрушается. Рис. 8.51(*в*) — результат после сжатия и восстановления рис. 8.51(*а*); СКО составляет 2,1 градации яркости. Если же попытаться восстановить водяной знак из этого изображения тем же способом, как и из изображения (*б*), результат невозможно интерпретировать (рис. 8.51(*г*)). Несмотря на то, что при сжатии с потерями (и восстановлении) в изображении сохранилась важная визуальная информация, нестойкий водяной знак был разрушен.

Стойкие невидимые водяные знаки должны сохраняться при модификациях изображения (так называемых *атаках*), будь то случайных или преднамерен-

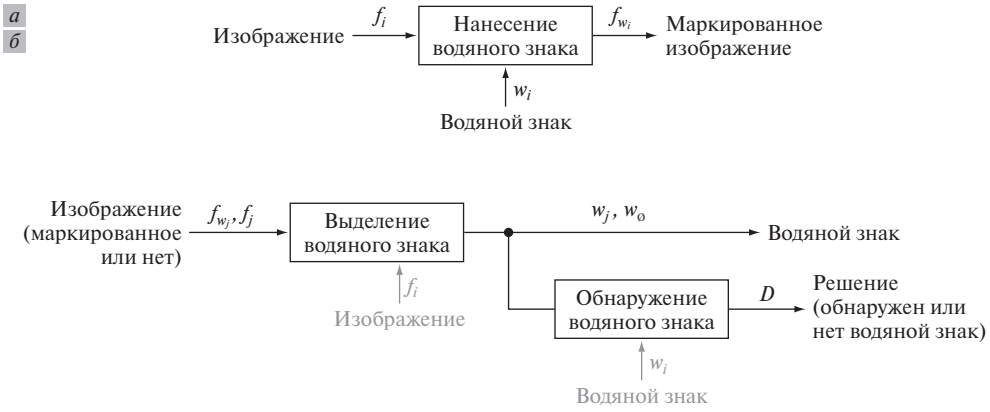


Рис. 8.52. Типичная система нанесения и проверки водяных знаков: (*а*) кодер; (*б*) декодер

ных. Обычно непреднамеренными атаками являются сжатие с потерями, линейная и нелинейная фильтрация, градационная коррекция, поворот, масштабирование и тому подобные операции. Преднамеренные атаки простираются от печати и пересканирования до нанесения дополнительных водяных знаков и/или шума. Конечно, не имеет смысла противостоять атакам, которые делают изображение как таковое бесполезным.

На рис. 8.52 представлены основные компоненты типичной системы нанесения и проверки водяных знаков. Кодер на рис. 8.52(*а*) наносит водяной знак (метку) w_i на изображение f_i, создавая изображение с водяным знаком f_{w_i}. Соответствующий ему декодер на (*б*), принимая на входе изображение f_{w_j}, либо выделяет и подтверждает наличие w_j, либо диагностирует непомеченное изображение f_j. Если водяной знак w_j является видимым, в декодере нет необходимости. Если же w_j невидим, декодеру, в зависимости от ситуации, для его действия может потребоваться копия f_i или w_i (показаны серым цветом на рис. 8.52(*б*)). Если в системе используются f_i или w_i, то такая система называется *закрытой* или системой *с ограниченным ключом*; если не используются, то она называется *открытой* или системой *с неограниченным ключом*. Так как декодер должен обрабатывать как помеченные, так и непомеченные изображения, то отсутствие водяного знака на рис. 8.52(*б*) обозначается знаком $w_\varnothing$. Наконец отметим, что для определения наличия w_i на изображении декодер должен осуществить корреляцию выделенного водяного знака w_j с w_i и сравнить результат с заранее заданным порогом. Порог устанавливает степень подобия, которая допустима для подтверждения «совпадения».

Пример 8.30. Невидимый стойкий водяной знак на основе ДКП.

■ Нанесение маркировки и ее выделение могут выполняться как в пространственной области (что было показано в предыдущем примере), так и в области преобразованных данных. На рис. 8.53(*а*) и (*в*) показаны два варианта маркировки изображения рис. 8.9(*а*), используя описанный ниже метод нанесения водяного знака на основе ДКП [Cox et al., 1997]:

Шаг 1. Вычислить двумерное ДКП изображения, подлежащего маркировке.

Шаг 2. Выделить K наибольших по амплитуде коэффициентов $c_1, c_2, ..., c_K$.

Шаг 3. Создать водяной знак генерацией псевдослучайной последовательности из K элементов $\omega_1, \omega_2, ..., \omega_K$, имеющих гауссово распределение со средним $\mu = 0$ и дисперсией $\sigma^2 = 1$.

Псевдослучайная последовательность чисел аппроксимирует свойства случайных чисел. Однако она не является действительно случайной, поскольку зависит от выбранного алгоритма генерации и заданного начального значения.

Шаг 4. Вставить водяной знак, полученный на шаге 3, в K наибольших коэффициентов ДКП, определенных на шаге 2, при помощи преобразования

$$c_i' = c_i(1 + \alpha\omega_i) \qquad 1 \le i \le K \tag{8.3-3}$$

при заданной константе $\alpha > 0$ (которая управляет степенью влияния ω_i на c_i). Заменить исходные значения c_i на c_i', вычисленные согласно (8.3-3).

Для изображений на рис. 8.53 $\alpha = 0{,}1$ и $K = 1000$.

Шаг 5. Вычислить обратное ДКП полученного на шаге 4 результата.

При использовании водяных знаков, полученных из псевдослучайных чисел и примененных к наиболее существенным для восприятия частотным компонентам изображения, α может быть выбран достаточно малым, уменьшая заметность водяного знака. С другой стороны, секретность самого водяного знака сохраняется высокой, поскольку (1) водяные знаки формируются при помощи случайных чисел без очевидной структуры; (2) водяные знаки внедряются во многие частотные компоненты с пространственным влиянием на все двумерное изображение (так что их локализация неочевидна); и (3) атаки против них могут привести к значительному ухудшению изображения как такового (т.к. чтобы воздействовать на водяные знаки, необходимо изменять наиболее существенные частотные компоненты изображения).

На рис. 8.53(*б*) и (*г*) демонстрируются два варианта видимых изменений яркостей изображения в результате добавления псевдослучайных чисел к коэффициентам ДКП. Очевидно, что псевдослучайные изменения коэффициентов ДКП должны оказывать влияние на изображение — пусть и слишком малое, чтобы его легко заметить. Чтобы показать эффект, вычислялись разности изображений на рис. 8.53(*а*) и (*в*) с исходным изображением на рис. 8.9(*а*), после чего эти разности масштабировались до полного диапазона яркостей [0, 255]. Полученные изображения на рис. 8.53(*б*) и (*г*) показывают двумерный пространственный вклад псевдослучайных значений. Поскольку разности масштабированы, то нельзя просто добавить их к изображению рис. 8.9(*а*) и получить рис. 8.53(*а*) и (*в*). Как видно по изображениям рис. 8.53(*а*) и (*в*), действительные изменения яркостей настолько малы, что практически незаметны.

Чтобы определить, действительно ли полученное изображение является копией исходного изображения с коэффициентами $c_1, c_2, ..., c_K$, на которое был

а *б*
в *г*

Рис. 8.53. (*а*) и (*в*) Изображение рис. 8.9(*а*) с двумя вариантами нанесенных водяных знаков; (*б*) и (*г*) разности (масштабированные по яркости) между изображениями с водяными знаками и исходным изображением рис. 8.9(*а*); эти разности показывают, как выглядит добавка псевдослучайного водяного знака к исходному изображению

нанесен водяной знак в виде последовательности $\omega_1, \omega_2, ..., \omega_K$, используется следующая процедура:

Шаг 1. Вычислить двумерное ДКП интересующего изображения.

Шаг 2. Выделить K коэффициентов ДКП (в позициях, соответствующих $c_1, c_2, ..., c_K$ шага 2 процедуры нанесения водяного знака) и обозначить их как $\hat{c}_1, \hat{c}_2, ..., \hat{c}_K$. Если интересующее изображение совпадает с первоначальным, на которое были нанесены водяные знаки (т. е. не модифицировалось), то $\hat{c}_i = c_i'$ для $1 \le i \le K$. Если оно было модифицировано (т. е. подверглось некому виду атаки), то $\hat{c}_i \approx c_i'$ для

$1 \le i \le K$ (т.е. $\hat{c}_i$ будет лишь приближением c_i'). Если же они далеки друг от друга по значениям, т.е. $\hat{c}_i$ не похожи на c_i', то полученное изображение либо несет другой водяной знак, либо вообще не содержит водяного знака.

Шаг 3. Вычислить водяной знак $\hat{\omega}_1, \hat{\omega}_2, ..., \hat{\omega}_K$, используя

$$\hat{\omega}_i = \frac{\hat{c}_i - c_i}{\alpha c_i} \qquad \text{для } 1 \le i \le K. \tag{8.3-4}$$

Напомним, что водяным знаком в данном случае является последовательность псевдослучайных чисел.

Шаг 4. Измерить сходство $\hat{\omega}_1, \hat{\omega}_2, ..., \hat{\omega}_K$, полученных на шаге 3, и $\omega_1, \omega_2, ..., \omega_K$, сформированных на шаге 3 процедуры нанесения водяного знака, с помощью какой-либо метрики, например коэффициента корреляции

$$\gamma = \frac{\sum_{i=1}^{K} (\hat{\omega}_i - \bar{\hat{\omega}})(\omega_i - \bar{\omega})}{\sqrt{\sum_{i=1}^{K} (\hat{\omega}_i - \bar{\hat{\omega}})^2 \sum_{i=1}^{K} (\omega_i - \bar{\omega})^2}} \quad 1 \le i \le K, \tag{8.3-5}$$

где $\bar{\omega}$ и $\bar{\hat{\omega}}$ — средние значения первоначальной и полученной случайных последовательностей соответственно.

Коэффициент корреляции подробно рассматривается в разделе 12.2.1.

Шаг 5. Сравнить величину измеренного сходства γ с предварительно заданным порогом T и сделать вывод на основании следующего решения:

$$D = \begin{cases} 1 & \text{если } \gamma \ge T, \\ 0 & \text{в остальных случаях.} \end{cases} \tag{8.3-6}$$

Другими словами, если $D = 1$, то это показывает, что водяной знак $\omega_1, \omega_2, ..., \omega_K$ присутствует (согласно выбранному порогу T); если же $D = 0$, то искомого водяного знака нет.

Используя описанную процедуру, изображение с водяными знаками на рис. 8.53(*а*) при сравнении со значениями своего же водяного знака дало коэффициент корреляции $\gamma = 0{,}9999$, что показывает безошибочное совпадение. Аналогичное сравнение изображения рис. 8.53(*в*) с тем же водяным знаком дало коэффициент корреляции $\gamma = 0{,}0417$; таким образом, поскольку коэффициент корреляции γ столь низкий, то изображение (*в*) не может быть спутано с изображением (*а*). ■

В завершение раздела отметим, что подход к нанесению водяных знаков, основанный на ДКП и рассмотренный в предыдущем примере, достаточно устойчив к атакам отчасти потому, что он является закрытым, т.е. методом с ограниченным ключом. Методы с ограниченным ключом всегда более устойчивы, чем их аналоги с неограниченным ключом. Взяв за основу изображение на рис. 8.53(*а*) с нанесенным водяным знаком, на рис. 8.54 показаны способности данного мето-

да противостоять атакам разного вида. Как видно по цифрам, обнаружение водяных знаков дает хорошие показатели при многих из применявшихся атак; цифры (значения коэффициентов корреляции, приведенные под каждым из изображений на рис. 8.54) варьируются от 0,3113 до 0,9945. При высококачественном JPEG-сжатии и восстановлении (сжатие с потерями, СКО около 7 градаций яркости) $\gamma = 0{,}9945$. Даже когда сжатие и восстановление привели к увеличению СКО до 10 градаций яркости и визуальное качество данного изображения сильно пострадало, корреляция уменьшилась до 0,7395. Значительное сглаживание при помощи пространственной фильтрации или нанесение гауссова шума не понижают коэффициент корреляции ниже 0,8230. Однако эквализация гистограммы заметно снизила γ до 0,5210; а незначительный поворот изображения привел к наихудшему результату, уменьшив γ до 0,3313. Все перечисленные виды атак, за исключением варианта (*а*) JPEG-сжатия и восстановления, существенно снизили возможности использования исходного изображения, на которое наносился водяной знак.

а *б* *в*
г *д* *е*

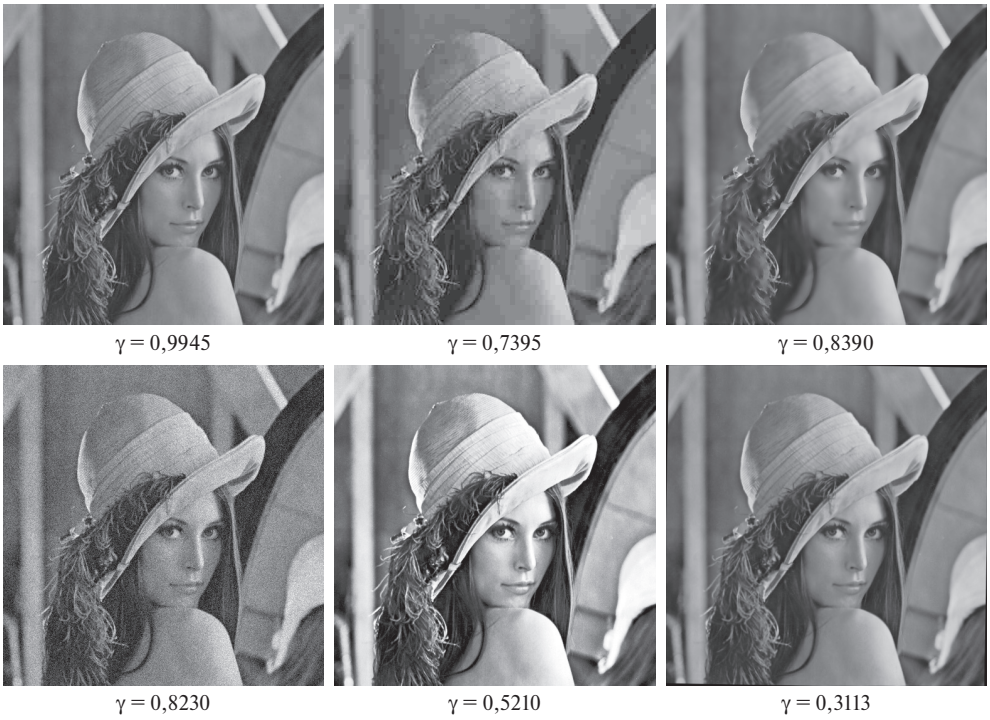


Рис. 8.54. Различные виды атак на изображение рис. 8.53(*а*): (*а*) JPEG-сжатие с потерями и восстановление, СКО 7 уровней яркости; (*б*) JPEG-сжатие с потерями и восстановление, СКО 10 уровней яркости (заметны блоковые артефакты); (*в*) сглаживание пространственным фильтром; (*г*) добавление гауссова шума; (*д*) эквализация гистограммы; (*е*) незначительный поворот изображения. Каждое изображение получено модификацией изображения рис. 8.53(*а*). После модификации они в разной степени сохранили водяной знак, что показано значениями коэффициентов корреляции, приведенными под каждым из изображений

Заключение

Главные цели настоящей главы заключались в том, чтобы изложить теоретические основы сжатия цифровых изображений, описать наиболее распространенные методы сжатия, а также ознакомить с близкой областью нанесения цифровых водяных знаков на изображение. Хотя материал представлен на вводном уровне, приводимые ссылки дают отправные точки для знакомства с огромным объемом литературы, посвященным сжатию изображений и смежным вопросам. Как становится очевидным из списка международных стандартов сжатия, приведенных в табл. 8.3 и 8.4, сжатие играет ключевую роль в хранении и передаче изображений, в интернет-технологиях, а также в коммерческом распространении видео (например DVD). Сжатие является одной из небольшого числа областей обработки изображений, которые приобрели достаточно большую коммерческую привлекательность, что гарантирует дальнейшее развитие имеющихся широко распространенных стандартов. Вопросы нанесения на изображение водяных знаков также становятся весьма важными, поскольку все больше и больше изображений распространяются в сжатом цифровом виде.

Ссылки и литература для дальнейшего изучения

Вводный материал настоящей главы, изложенный в разделе 8.1, представляет теоретические основы сжатия изображений и может быть найден в том или ином виде в большинстве книг по обработке изображений, приведенных в конце главы 1. Для получения дополнительной информации по вопросам зрительной системы человека см. [Netravali, Limb, 1980], [Huang, 1966], [Schreiber, Knapp, 1958], а также ссылки в конце главы 2. Для более глубокого изучения теории информации см. интернет-сайт книги, а также [Abramson, 1963], [Blahut, 1987] и [Berger, 1971]. Фундаментальные основы всего этого раздела были заложены в классической работе К. Шеннона «Математическая теория связи» [Shannon, 1948], которую мы настоятельно рекомендуем читателю. Субъективные критерии качества обсуждаются в [Frendendall, Behrend, 1960].

На протяжении главы многие стандарты сжатия использованы в примерах. Реализация большинства из них осуществлялась посредством Adobe Photoshop (при помощи свободно распространяемых дополнительных программных модулей) либо MATLAB, как описано в [Gonzalez et al., 2004]. Описания стандартов сжатия, как правило, многословны и сложны; мы даже не пытались раскрыть какой-то из них во всей полноте. Для получения более подробной информации по каким-то стандартам см. опубликованные документы соответствующих организаций по стандартам[32] — Международной организации по стандартизации (International Standards Organization, ISO), Международной электротехнической комиссии, МЭК (International Electrotechnical

[32] В настоящее время большинство из необходимой документации по стандартам сжатия можно найти в Интернете. — *Прим. перев.*

Commission, IEC) и/или Сектора стандартизации телекоммуникаций Международного союза электросвязи (International Telecommunications Union — Telecommunications, ITU-T) — организации, ранее называвшейся Международный консультационный комитет по телефонии и телеграфии, MKKTT (Consultative Committee of the International Telephone and Telegraph, CCITT). В качестве дополнительных ссылок по стандартам сжатия можно указать [Hunter, Robinson, 1980], [Ang et al., 1991], [Fox, 1991], [Pennebaker, Mitchell, 1992], [Bhatt et al., 1997], [Sikora, 1997], [Bhaskaran, Konstantinos, 1997], [Ngan et al., 1999], [Weinberger et al., 2000], [Symes, 2001], [Mitchell et al., 1997] и [Manjunath et al., 2001].

Описание методов сжатия с потерями и без потерь, приведенные в разделе 8.2, а также методы нанесения водяных знаков по большей части основаны на оригинальных статьях, цитированных в тексте, а также на указанных ниже работах. Рассматриваемые в них алгоритмы отражают положение дел в данной области, но никоим образом ее не исчерпывают. Результаты, касающиеся LZW-кодирования, восходят к работам [Ziv, Lempel, 1977, 1978]. Материалы по арифметическому кодированию следуют разработкам [Witten, Neal, Cleary, 1987]. Одно из наиболее важных применений арифметического кодирования приведено в работе [Pennebaker et al., 1988]. Хорошее обсуждение кодирования с предсказанием без потерь можно найти в руководстве [Rabbani, Jones, 1991]. Адаптивный предсказатель (8.2-56) взят из [Graham, 1958]. По поводу компенсации движения см. [S.Solari, 1997], где также содержится введение в общие вопросы и стандарты сжатия видео, а также см. [Mitchell et al., 1997]. Методы нанесения водяных знаков на основе ДКП раздела 8.3 базируются на статье [Cox et al., 1997]. Также касательно нанесения водяных знаков см. [Cox et al., 2001], [Parhi, Nishitani, 1999] и статью [S. Mohanty, 1999].

Проблеме сжатия изображений посвящено много обзорных статей. Особого внимания заслуживают обзоры [Netravali, Limb, 1980], [A. K. Jain, 1981], специальный выпуск по системам передачи изображений в журнале [*IEEE Transactions on Communications*, 1981], специальный выпуск по кодированию графики в сборнике [*Proceedings of IEEE*, 1980], специальный выпуск по системам коммуникации в сборнике [*Proceedings of IEEE*, 1985], специальный выпуск по сжатию видеопоследовательностей в журнале [*IEEE Transactions on Image Processing*, 1994], а также специальный выпуск по векторному квантованию в журнале [*IEEE Transactions on Image Processing*, 1996]. Кроме того, многие выпуски *IEEE Transactions on Image Processing*, *IEEE Transactions on Circuits and Systems for Video Technology* и *IEEE Transactions on Multimedia* содержат статьи, касающиеся сжатия как видео, так и неподвижных изображений, компенсации движения и нанесения водяных знаков. В качестве отправной точки для дальнейшего чтения и ссылок см., например, [Robinson, 2006], [Chandler, Hemami, 2005], [Yan, Cosman, 2003], [Boulgouris et al., 2001], [Martin, Bell, 2001], [Chen, Wilson, 2000], [Hartenstein et al., 2000], [Yang, Ramchandran, 2000], [Meyer et al., 2000], [S. Mitra et al., 1998], [Mukherjee, Mitra, 2003], [Xu et al., 2005], [Rane, Sapiro, 2001], [Hu et al., 2006], [Pi et al., 2006], [Dugelay et al., 2006] и [Kamstra, Heijmans, 2005].

Задачи

8.1 **(а)** Может ли процедура неравномерного кодирования сжать изображение с эквализованной гистограммой, имеющее 2^n уровней яркости? Поясните.

(б) Может ли такое изображение иметь пространственную или временную избыточность, которую можно было бы использовать для сжатия данных?

8.2 Одна из процедур кодирования длин серий включает (1) кодирование только серий из нулей или единиц (не обеих) и (2) присвоение специального кода началу каждой строки с целью снижения эффекта ошибок при передаче. Возможной кодовой парой является (x_k, r_k), где x_k и r_k представляют соответственно координату начала и длину k-й серии. Код (0, 0) используется для начала каждой новой строки.

(а) Выведите общее выражение для наибольшего среднего числа серий на строке изображения, требуемого для обеспечения эффекта сжатия двоичного изображения размерами $2^n \times 2^n$.

(б) Вычислите максимально допустимое значение для $n = 8$.

8.3 Рассмотрите 8-элементную строку полутонового изображения, содержащую значения {255, 118, 127, 182, 18, 178, 82, 55}. Если они равномерно квантованы с 4-битовой точностью, вычислите СКО ошибки квантования и среднеквадратическое отношение сигнал—шум для квантованных данных.

★**8.4** Хотя квантование и приводит к потере информации, иногда его можно сделать незаметным для глаз. Например, при квантовании 8-битового изображения на меньшее число бит часто появляются ложные контуры. Этот эффект можно уменьшить или удалить при помощи так называемого *модифицированного квантования яркости*, МКЯ (improved gray-scale quantization, IGS). Метод заключается в следующем. Вводится понятие *остатка квантования* как разности между квантуемым значением и результатом квантования. Этот остаток переносится и прибавляется к значению соседнего (в строке) пикселя, для которого будет квантоваться уже полученная сумма; новый остаток квантования переносится следующему пикселю и т. д. Для первого пикселя строки начальный остаток равен нулю. Пусть квантование происходит на 16 градаций, т. е. из 8-битового кода остаются лишь старшие 4 разряда.

(а) Постройте МКЯ-код для значений яркости {108, 139, 135, 244, 172, 178, 56, 97}.

(б) Вычислите СКО ошибки квантования и среднеквадратическое отношение сигнал—шум для декодированных МКЯ данных.

8.5 8-битовое изображение размерами 1024×1024 с энтропией 4,2 бит/пиксель (вычисленной по его гистограмме с помощью выражения (8.1-7)) кодируется кодом Хаффмана.

(а) Какое максимальное сжатие можно ожидать?

(б) Будет ли оно достигнуто?

(в) Если требуется более высокая степень сжатия, что еще можно сделать?

★**8.6** Единицу информации по основанию e обычно называют *натом*, а по основанию 10 — *хартли*. Вычислите коэффициенты преобразования, необходимые для связи этих единиц с единицей информации по основанию 2 (битом).

★**8.7** Докажите, что для источника без памяти с q символами максимальное значение энтропии составляет $\log_2 q$, которое достигается тогда и только тогда, когда все символы источника равновероятны. (*Совет*: рассмотрите величину $\log_2 q - H(\mathbf{z})$ и воспользуйтесь неравенством $\log_2 x \le x - 1$.)

8.8 **(а)** Сколько различных кодов Хаффмана существует для четырехсимвольного источника?

(б) Постройте их.

8.9 Рассмотрите простое 8-битовое изображение размерами 4×8:

21	21	95	95	169	169	243	243
21	21	95	95	169	169	243	243
21	21	95	95	169	169	243	243
21	21	95	95	169	169	243	243

(а) Вычислите энтропию изображения.

(б) Сожмите изображение, используя код Хаффмана.

(в) Вычислите полученное сжатие и эффективность хаффмановского кодирования.

★ **(г)** Рассмотрите код Хаффмана *пар пикселей*, а не одиночных пикселей. Т. е. рассмотрите изображение, создаваемое вторым расширением источника без памяти, который создавал исходное изображение. Какова энтропия полученного изображения, состоящего из пар пикселей?

(д) Рассмотрите кодирование разности между двумя соседними пикселями. Какова будет энтропия нового изображения разностей? Что нам это говорит о сжатии изображения?

(е) Объясните различия значений энтропии в случаях (а), (г) и (д).

8.10 Используя код Хаффмана на рис. 8.8, декодируйте закодированную последовательность: 01010010000011101011.

8.11 Вычислите код Голомба $G_5(n)$ для $0 \le n \le 20$.

8.12 Напишите общую процедуру для декодирования кода Голомба $G_m(n)$.

8.13 Почему невозможно вычислить код Хаффмана для неотрицательных целых $n \ge 0$ с геометрическим распределением вероятностей, заданных выражением (8.2-2)?

8.14 Вычислите экспоненциальный код Голомба $G^1_{\exp}(n)$ для $0 \le n \le 15$.

★**8.15** Напишите общую процедуру декодирования экспоненциального кода Голомба $G^k_{\exp}(n)$.

8.16 Нарисуйте график оптимального кодового параметра m кода Голомба как функцию ρ для $0 < \rho < 1$ в (8.2-3).

8.17 Пусть задан четырехсимвольный источник $\{a, b, c, d\}$ с вероятностями {0,1, 0,4, 0,3, 0,2}. Закодируйте арифметическим кодом последовательность *abcda*.

★**8.18** Процесс арифметического декодирования является обратным по отношению к процессу кодирования. Декодируйте сообщение 0,32256, сформированное моделью кода

Символ	Вероятность
a	0,2
e	0,3
i	0,1
o	0,2
u	0,1
!	0,1

8.19 С помощью алгоритма кодирования LZW из раздела 8.2.4 закодируйте строку 7-битовых ASCII-символов «AAAAAAAAAAA» (в коде ASCII символ A имеет битовое представление 1000001).

8.20 Разработайте алгоритм для декодирования LZW-кодированного выхода в примере 8.7. Поскольку словарь, который использовался при кодировании, недоступен, кодовая книга должна формироваться по мере декодирования.

8.21 Декодируйте BMP-кодированную последовательность {127, 0, 5, 25, 29, 40, 103, 52, 75, 82}.

8.22 **(а)** Постройте полный 5-битовый код Грэя.

(б) Создайте общую процедуру для преобразования числа, представленного кодом Грэя, в его двоичный эквивалент и используйте его для декодирования последовательности 0111010100111.

8.23 Используйте алгоритм сжатия MKKTT группы 4 для кодирования второй строки следующего сегмента из двух строк:

0 1 1 0 0 1 1 1 0 0 1 1 1 1 1 1 1 1 0 0 0 0 1
1 1 1 1 1 0 0 0 1 1 1 0 0 0 1 1 1 1 1 1 0 0 0.

Предположите, что начальный опорный элемент a_0 расположен на первом элементе второй строки сегмента.

8.24 **(а)** Выпишите все члены JPEG-коэффициентов DC категории разностей 3.

(б) На основании табл. A.4 вычислите их коды Хаффмана, используемые по умолчанию.

8.25 Какое количество вычислений требуется для поиска оптимального вектора движения макроблока размерами 8×8 элементов, используя в качестве критерия оптимальности величину среднего абсолютного искажения (8.2-37), точность в один пиксель и максимальное допустимое смещение в 16 пикселей? Как это будет выглядеть в случае точности ½ пикселя?

8.26 В чем состоит выгода использования B-кадров для компенсации движения?

8.27 Нарисуйте блок-схему видеодекодера с компенсацией движения, соответствующего кодеру на рис. 8.39.

8.28 Изображение, автокорреляционная функция которого задана в форме (8.2-49) с $\rho_h = 0$, кодируется при помощи ДИКМ-кодера с использованием предсказателя второго порядка.

(а) Сформируйте автокорреляционную матрицу **R** и вектор **r**.

(б) Найдите оптимальные коэффициенты предсказания.

(в) Определите дисперсию ошибки предсказания, которая получится в результате использования оптимальных коэффициентов.

8.29 Найдите пороговые уровни и уровни квантования квантователя Ллойда — Макса для $L = 8$ и равномерной функции распределения вероятностей

$$p(s) = \begin{cases} \dfrac{1}{2A} & -A \le s \le A; \\ 0 & \text{в остальных случаях.} \end{cases}$$

8.30 Рентгенолог из известного исследовательского центра одного из госпиталей недавно посетил медицинскую конференцию, где была представлена система, которая может передавать оцифрованные рентгеновские изображения размерами 4096×4096, 12-бит, по стандартному каналу T1 (1,544 Мбит/с). Система передает изображения в сжатом виде с использованием методики последовательных приближений. При этом на приемной стороне сначала восстанавливается некоторое приближение рентгеновского снимка, которое затем постепенно улучшается до точного воспроизведения. Передача данных, необходимых для построения первого приближения, требует от 5 до 6 с. времени. Улучшения происходят (в среднем) каждые 5 или 6 с. в течение следующей 1 минуты при том, что первое улучшение является наиболее значительным, а последнее — наименее заметным. Данная система произвела сильное впечатление на врача, поскольку он может начать диагностику уже при воспроизведении первого приближения и закончить ее к моменту полного безошибочного воспроизведения рентгеновского изображения. Вернувшись в госпиталь, он подал в администрацию госпиталя запрос на покупку. К сожалению, бюджет госпиталя был невелик, поскольку недавно был принят на работу молодой перспективный и целеустремленный выпускник электротехнического вуза. Чтобы удовлетворить пожелания рентгенолога, администратор дал молодому инженеру задачу разработать такую систему. Он полагал, что разработка и создание подобной системы собственными силами окажется дешевле. Госпиталь располагал некоторыми из элементов такой системы, но передача несжатого рентгеновского изображения занимала более 2 мин. Администратор попросил инженера подготовить исходную блок-схему возможной системы к собранию персонала во второй половине дня. Имея лишь немного времени и экземпляр данной книги, оставшийся после недавнего обучения, инженер смог разработать концепцию системы, которая удовлетворяла условиям на скорость передачи и связанным с этим требованиям к сжатию. Составьте концептуальную блок-схему такой системы, точно определив, какие именно методы сжатия Вы бы рекомендовали.

8.31 Покажите, что вейвлет-преобразование, основанное на применении лифтинг-схемы, определенной уравнениями (8.2-62), эквивалентно применению обычного банка фильтров БВП с коэффициентами, заданными в табл. 8.20. Найдите значения коэффициентов фильтров в терминах α, β, γ, δ и K.

8.32 Вычислите величины шагов квантования отдельных субполос для изображения, кодированного методом JPEG-2000, в котором используется неявное квантование, а на мантиссу и порядок отведено 8 битов субполосы $2LL$.

8.33 Как можно добавить к изображению видимый водяной знак в частотной области?

★**8.34** Постройте систему нанесения невидимых водяных знаков на основе дискретного преобразования Фурье.

8.35 Постройте систему нанесения невидимых водяных знаков на основе дискретного вейвлет-преобразования.

ГЛАВА 9

МОРФОЛОГИЧЕСКАЯ ОБРАБОТКА ИЗОБРАЖЕНИЙ

Я рос столь похожим на брата чертами и телосложением,
что люди принимали меня за него, а его — за меня.

Генри Сэмбрук Лей,
«Песни Кокейна. Близнецы»

Введение

Словом *морфология* обычно обозначают ту область биологии, которая занимается формой и строением животных и растений. Мы будем использовать здесь то же слово в контексте *математической морфологии* — инструмента для извлечения некоторых компонент изображения, полезных для его представления и описания, например, границ, остовов и выпуклых оболочек. Интерес также представляют морфологические методы, применяемые на этапах предварительной и заключительной обработки, например морфологическая фильтрация, утончение и усечение.

В следующих разделах вводятся в рассмотрение и иллюстрируются некоторые важные понятия математической морфологии. Многие из представленных идей допускают формулировку в терминах n-мерного евклидова пространства E^n, однако мы прежде всего интересуемся двоичными изображениями, состоящими из элементов Z^2 (см. раздел 2.4.2). Обобщение описанных концепций для полутоновых изображений рассматривается в разделе 9.6.

С материала данной главы начинается переход от методов, сконцентрированных исключительно на обработке изображений (по смыслу определения в разделе 1.1.1, когда и на входе, и на выходе процесса присутствуют изображения), к таким процессам, на вход которых по-прежнему поступают изображения, но выходными результатами являются некоторые атрибуты, извлекаемые из этих изображений. Такие инструменты, как морфология и сопутствующие ей концепции, играют роль краеугольного камня, составленного из математических основ и служащего для извлечения «смысла» изображения. Развитию и применению других подходов к такой проблеме посвящены оставшиеся главы книги.

9.1. Начальные сведения

В математической морфологии используется язык теории множеств. Морфология как таковая предлагает единый мощный подход для многочисленных

задач обработки изображений. Множества в математической морфологии соответствуют объектам на изображении. Например, множество всех белых пикселей двоичного (двухградационного, т. е. содержащего только элементы со значениями 0 или 1) изображения является одним из вариантов его полного морфологического описания. В двоичных изображениях обсуждаемые множества являются подмножествами двумерного целочисленного пространства Z^2 (см. раздел 2.4.2) с элементами в виде пар чисел, т. е. двумерных векторов (x, y), координаты которых указывают на белый (или черный, в зависимости от соглашения) пиксель изображения. Полутоновые цифровые изображения, рассматривавшиеся в предыдущих главах, могут быть описаны множествами, состоящими из элементов пространства Z^3. В этом случае две координаты элемента множества указывают положение пикселя, а третья соответствует значению его яркости. Множества в пространствах более высокой размерности могут с помощью дополнительных координат описывать и другие характеристики изображения, например его цветовые компоненты или закон изменения во времени.

> Перед тем, как двигаться дальше, читателю будет полезно освежить в памяти материал разделов 2.4.2 и 2.6.4.

Помимо основных операций над множествами, которые были введены в разделе 2.6.4, в морфологии широко применяются понятия центрального отражения и параллельного переноса множеств. *Центральное отражение* множества B, обозначаемое $\hat{B}$, определяется следующим образом:

> Операция центрального отражения множества эквивалентна повороту множества B на 180° относительно начала координат, что есть один из вариантов пространственной свертки изображения (см. раздел 3.4.2).

$$\hat{B} = \{w \mid w = -b,\ b \in B\}. \tag{9.1-1}$$

Если B — множество пикселей (точек плоскости), представляющее объект на изображении, то $\hat{B}$ состоит из точек B, у которых координаты (x, y) заменены на $(-x, -y)$. На рис. 9.1(*а*) и (*б*) представлены множество простой формы и его центральное отражение.[1]

Параллельный перенос (или *сдвиг*) множества B в точку $z = (z_1, z_2)$ обозначается $(B)_z$ и определяется по следующему правилу:

$$(B)_z = \{c \mid c = b + z,\ b \in B\}. \tag{9.1-2}$$

Если B — множество пикселей (точек плоскости), представляющее объект на изображении, то $(B)_z$ состоит из точек B, у которых координаты (x, y) замене-

[1] При схематическом изображении множеств на рисунках, подобных рис. 9.1, точки рассматриваемого множества затемнены. При обработке двоичных изображений интерес представляют пиксели, которые соответствуют объектам на изображении. Эти пиксели отображаются белым цветом, а все прочие — черным. Для обозначения множеств пикселей, которые определяют объекты и части изображения, не являющиеся объектами, часто соответственно используют термины *передний план* и *фон*.

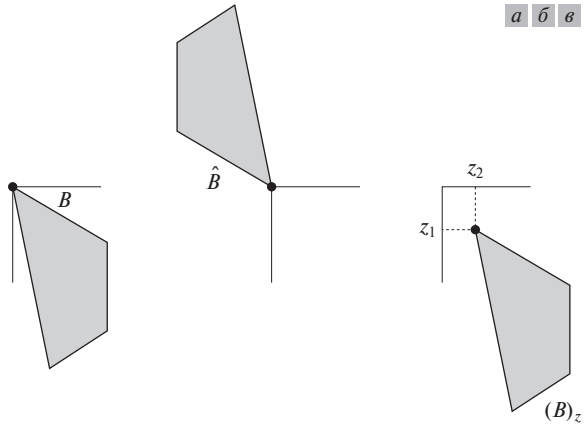


Рис. 9.1. (*а*) Множество, (*б*) его центральное отражение и (*в*) его сдвиг в точку z

ны на $(x + z_1, y + z_2)$. Рис. 9.1(*в*) иллюстрирует это определение для множества B из рис. 9.1(*а*).

Сдвиг и центральное отражение множеств часто применяются для формулирования морфологических операций на основе так называемых *структурообразующих множеств* или *примитивов*[2] — небольшого набора элементарных изображений, которые используются при исследовании интересующих свойств рассматриваемого изображения. На рис. 9.2 в верхнем ряду представлены примеры структурообразующих множеств, где каждый затемненный квадратик означает элемент соответствующего примитива. В тех случаях, когда не имеет значения, принадлежит ли некоторый элемент данному примитиву, будем отмечать такой элемент символом «×», означающим «несущественно», как это далее делается в разделе 9.5.4. Помимо указания, какие элементы принадлежат некоторому примитиву, необходимо также задать положение его *центра* (начала координат). Центры структурообразующих множеств на рис. 9.2 указаны жирной точкой (хотя обычно центр совпадает с центром тяжести примитива, в общем случае выбор центра примитива зависит от решаемой задачи). Если примитив обладает центральной симметрией и точка не указана, предполагается, что центр находится в центре симметрии.

Для удобства обработки изображений нужно, чтобы структурообразующие множества были прямоугольными массивами. Это достигается путем добавления к примитиву минимально необходимого числа фоновых элементов (на рис. 9.2 это не затемненные элементы), дополняющих его до прямоугольника, как демонстрируют первый и последний примитивы в нижнем ряду на рисунке. Два другие примитива из верхнего ряда уже имеют прямоугольную форму.

В качестве вводного примера к использованию структурообразующих множеств при морфологической обработке изображений рассмотрим рис. 9.3. На рис. 9.3(*а*) и (*б*) представлены множество простой формы и примитив. Как говорилось выше, для удобства компьютерной реализации множество A представлено в виде прямоугольного массива путем добавления фоновых элементов.

---

[2] В оригинале использован термин *structuring element. — Прим. перев.*

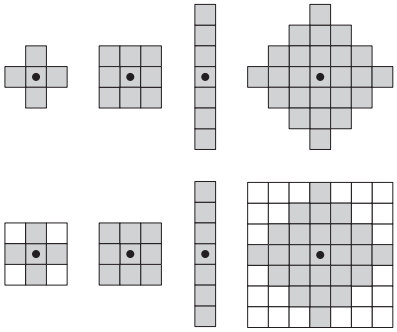


Рис. 9.2. Верхний ряд: примеры структурообразующих множеств (примитивов). Нижний ряд: примитивы, преобразованные в прямоугольные массивы. Точкой обозначен центр каждого примитива

Такая фоновая рамка делается достаточно большой, чтобы примитив целиком помещался на изображении, когда его центр находится на границе исходного множества (это аналогично дополнению изображения фоном при вычислении пространственной корреляции и свертки, как описывалось в разделе 3.4.2). В данном случае примитив имеет размеры 3×3 с центром в центральном элементе, поэтому достаточно окружить все множество фоновой рамкой шириной в один элемент, что показано на рис. 9.3(*в*). Как и на рис. 9.2, примитив тоже дополнен минимально необходимым количеством фоновых элементов, превращающих его в прямоугольный массив (рис. 9.3(*г*)).

> Для простоты последующих иллюстраций мы всюду предполагаем, что элементы фона добавлены в необходимом количестве, и не будем их показывать, если это ясно из рисунка.

Предположим, что операция над множеством A с использованием примитива B определена следующим образом. Новое множество строится путем наложения центра примитива B поочередно на каждый элемент A. Если при этом весь примитив содержится в исходном множестве, то местоположение

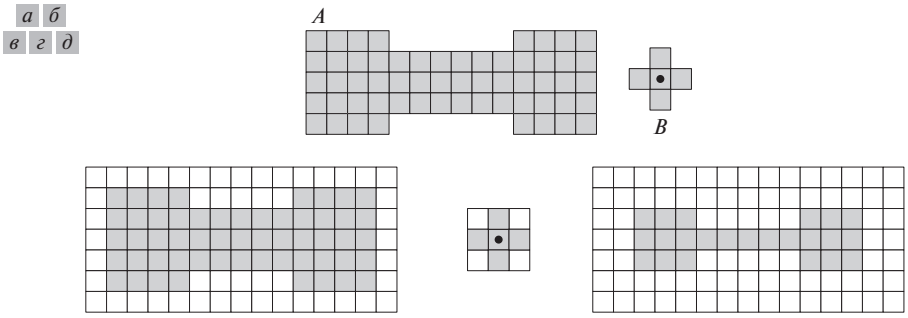


Рис. 9.3. (*а*) Множество, состоящее из затемненных элементов. (*б*) Структурообразующее множество (примитив). (*в*) Множество, дополненное элементами фона до прямоугольного массива с бордюром из фона. (*г*) Примитив с дополнением до прямоугольного массива. (*д*) Результат обработки с помощью этого примитива

центра считается элементом нового множества (будет показано серым цветом), и наоборот. Результат такой операции представлен на рис. 9.3(*д*). Видно, что когда центр примитива совпадает с граничными элементами *A*, часть элементов *B* выходят за пределы исходного множества, и тем самым такие положения центра исключаются из числа элементов нового множества. Общий результат операции состоит в том, что граница множества повсеместно разрушается, *эродирует*, как показано на рис. 9.3(*д*). Говоря, что «примитив содержится в исходном множестве», конкретно имеется в виду, что все элементы примитива *B* попадают на элементы множества *A*. Иными словами, хотя множества *A* и *B* показаны в форме прямоугольных массивов, содержащих как элементы множеств, так и элементы фона, только отмеченные серым элементы множества учитываются при определении того, содержится примитив *B* в множестве *A* или нет. Такие концепции лежат в основе материала следующего раздела, поэтому важно полностью понять рис. 9.3, прежде чем двигаться дальше.

9.2. Эрозия и дилатация

Мы начнем рассмотрение морфологических операций с детального изучения двух операций: *эрозии* и *дилатации*. Эти операции имеют основополагающее значение для морфологической обработки изображений. По существу, многие из обсуждаемых в данной главе алгоритмов морфологической обработки основаны на этих двух базовых операциях.

9.2.1. Эрозия

Для множеств *A* и *B* из пространства Z^2 *эрозия A* по *B*, обозначаемая $A \ominus B$, определяется как

$$A \ominus B = \left\{ z \,|\, (B)_z \subseteq A \right\}. \tag{9.2-1}$$

Иначе говоря, эрозия множества *A* по множеству *B* — это множество всех таких точек *z*, при сдвиге в которые множество *B* целиком содержится в *A*. В дальнейшем предполагается, что *B* есть структурообразующее множество (примитив). Выражение (9.2-1) является математической формулировкой действий в примере на рис. 9.3(*д*) в конце предыдущего раздела. Поскольку утверждение «*B* целиком содержится в *A*» эквивалентно тому, что *B* не имеет общих элементов с фоном, эрозию можно выразить в следующей эквивалентной форме:

$$A \ominus B = \left\{ z \,|\, (B)_z \cap A^c = \varnothing \right\}, \tag{9.2-2}$$

где в соответствии с определениями из раздела 2.6.4 A^c — дополнение множества *A*, а $\varnothing$ — пустое множество.

Рис. 9.4 демонстрирует пример эрозии. Элементы множеств *A* и *B* показаны затемненными, а фон белым. Сплошная линия на рис. 9.4(*в*) соответствует тем предельным значениям *z*, при сдвиге центра примитива *B* за которые перестает выполняться условие, требующее, чтобы *B* было подмножеством множества *A*. Таким образом, геометрическое место точек внутри (включи-

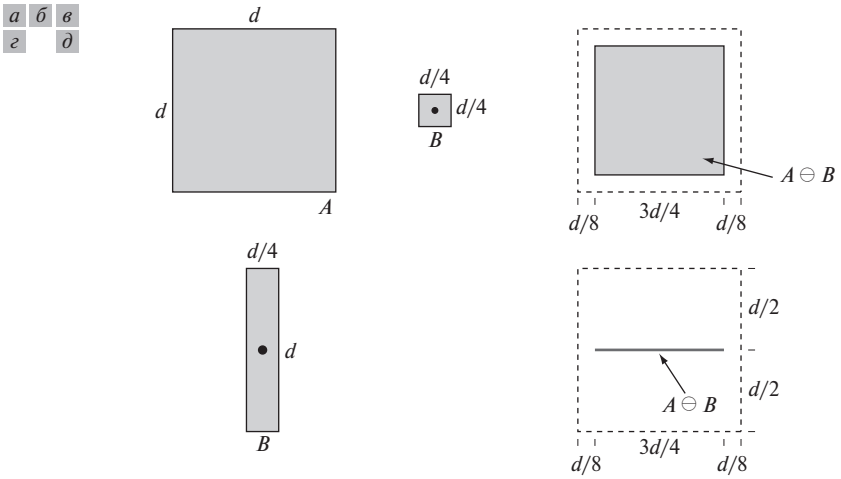


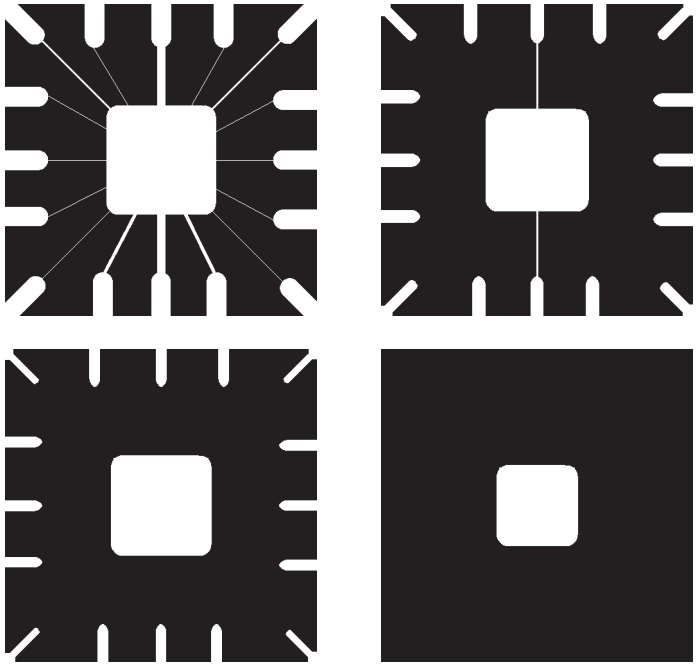
Рис. 9.4. (*а*) Множество *A*. (*б*) Примитив *B* в форме квадрата. (*в*) Результат эрозии *A* по *B* (затемненная область). (*г*) Примитив вытянутой формы. (*д*) Эрозия *A* по такому примитиву. Пунктирная рамка на (*в*) и (*д*) указывает границу множества *A* для наглядности сопоставления

тельно) сплошной границы (т. е. затемненная область) является результатом эрозии множества *A* по примитиву *B*. Подчеркнем, что этот результат есть просто множество значений *z*, удовлетворяющих (9.2-1) или (9.2-2). Граница множества *A* показана на рис. 9.4(*в*) и (*д*) пунктирной линией только для наглядности сопоставления. На рис. 9.4(*г*) изображен примитив вытянутой формы, а рис. 9.4(*д*) демонстрирует результат эрозии множества *A* по такому примитиву. Заметим, что в результате эрозии исходное множество сжалось до горизонтальной линии.

Выражения (9.2-1) и (9.2-2) являются не единственными возможными определениями эрозии (см. задачи 9.9 и 9.10, где приводятся два отличающихся, хотя и эквивалентных определения). Однако приведенные выше определения лучше других формулировок тем, что они более наглядны, если рассматривать примитив *B* в качестве пространственной маски (раздел 3.4.1).

Пример 9.1. Применение эрозии для удаления деталей изображения.

■ Предположим, что задача состоит в том, чтобы на изображении маски соединений микросхемы (рис. 9.5(*а*)) убрать все линии, соединяющие центральную область с периферией. Применяя к изображению операцию эрозии по квадратному примитиву с размерами 11×11 пикселей, все элементы которого имеют значения 1, получим результат на рис. 9.5(*б*), где большинство линий удалены. Причина, по которой две вертикальные линии в середине остались, хотя и стали тоньше, в том, что ширина этих линий была больше 11 пикселей. Если увеличить размеры примитива до 15×15, то эрозия по нему исходного изображения приведет к удалению всех линий, как показано на рис. 9.5(*в*) (альтернативой может быть повторное применение операции эрозии с тем же примитивом 11×11 к ранее полученному результату на рис. 9.5(*б*)). Дальнейшее увеличение размеров примитива приведет к удалению более крупных деталей. Например, при ис-



а б
в г

Рис. 9.5. Применение эрозии для удаления деталей изображения. (*а*) Двоичное изображение маски соединений микросхемы, имеющее размеры 486×486 пикселей. (*б*)—(*г*) Результаты эрозии изображения (*а*) по квадратному примитиву размерами соответственно 11×11, 15×15 и 45×45 пикселей, заполненному единицами

пользовании примитива размерами 45×45 пикселей полностью пропадут контактные площадки на периферии, как демонстрирует рис. 9.5(*г*).

Как показывает этот пример, эрозия приводит к тому, что объекты на двоичном изображении становятся меньше и тоньше. По существу, эрозию можно рассматривать как операцию *морфологической фильтрации*, которая отфильтровывает (удаляет) детали изображения с размерами меньше структурообразующего множества. На рис. 9.5 эрозия выполняет функции «фильтра линий». Мы вернемся к понятию морфологического фильтра в разделах 9.3 и 9.6.3. ■

9.2.2. Дилатация

Пусть A и B — множества из пространства Z^2. *Дилатация* множества A по множеству B (или относительно B) обозначается $A \oplus B$ и определяется как

$$A \oplus B = \left\{ z \middle| \left(\hat{B} \right)_z \cap A \neq \varnothing \right\}. \qquad (9.2\text{-}3)$$

В основе этого соотношения лежит получение центрального отражения множества B относительно его центра и затем сдвиг полученного множества в точку z. При этом дилатация множества A по B — это множество всех таких *смещений* z, при которых множества $\hat{B}$ и A совпадают по меньшей мере в одном элементе.

Исходя из такой интерпретации, соотношение (9.2-3) можно переписать в следующем виде[3]:

$$A \oplus B = \left\{ z \middle| \left[(\hat{B})_z \cap A \right] \subseteq A \right\}. \tag{9.2-4}$$

Как и раньше, предполагается, что B — структурообразующее множество (примитив), а A — множество (набор объектов на исходном изображении), подлежащее дилатации.

Выражения (9.2-3) и (9.2-4) представляют собой не единственные определения дилатации, встречающиеся в современной литературе (см. задачи 9.11 и 9.12, где приводятся два отличающихся, хотя и эквивалентных определения). Однако приведенные выше определения лучше других формулировок тем, что они более наглядны, если рассматривать примитив B в качестве маски свертки. Основной процесс, состоящий в «перевороте» множества B относительно его центра и затем в последовательном «скольжении» по множеству A (т.е. изображению), по сути аналогичен процессу свертки, обсуждавшемуся в разделе 3.4.2. Однако следует учитывать, что дилатация основана на операциях над множествами, т.е. не является линейной операцией, а свертка использует арифметические операции и потому линейна.

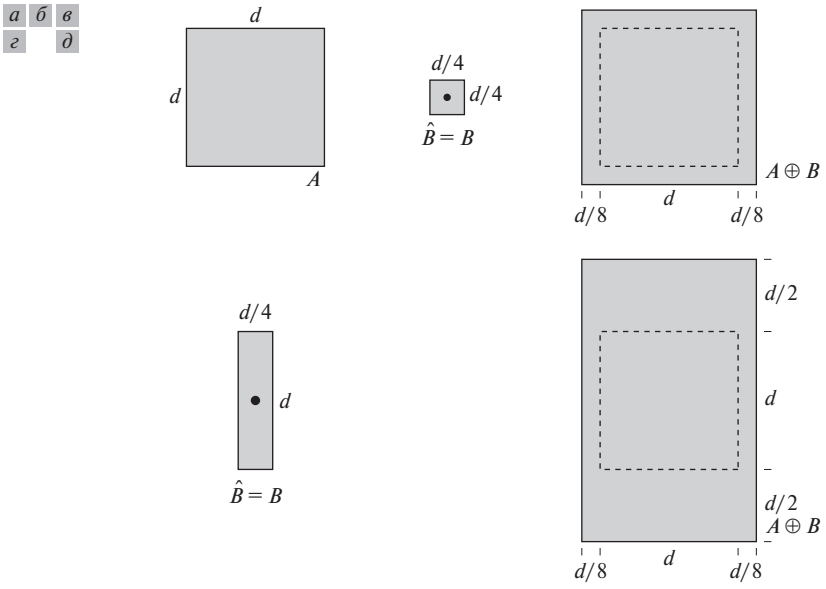


Рис. 9.6. (*а*) Множество A. (*б*) Структурообразующее множество (примитив) квадратной формы; точкой обозначен центр. (*в*) Результат дилатации A по B (затемненная область). (*г*) Примитив вытянутой формы. (*д*) Дилатация A по такому примитиву. Пунктирная рамка на (*в*) и (*д*) указывает границу множества A для наглядности сопоставления

[3] Строго говоря, выписанное соотношение не эквивалентно определению (9.2-3), а может рассматриваться лишь как его следствие, поскольку пустое множество также является подмножеством любого множества. Авторы, очевидно, исходят из противоположного. — *Прим. перев.*

В отличие от эрозии, которая приводит к сужению или утончению, дилатация «расширяет» или «утолщает» объекты на двоичном изображении. Конкретный вид и степень такого утолщения определяются формой используемого примитива. На рис. 9.6(*а*) приведено то же множество, что и на рис. 9.4, а на рис. 9.6(*б*) изображен примитив (в данном случае сам примитив и его отражение совпадают, поскольку это структурообразующее множество симметрично относительно центра). Пунктирной линией на рис. 9.6(*в*) для сравнения показаны границы исходного множества, а сплошная линия соответствует тем предельным значениям z, при выходе центра множества $\hat{B}$ за которые пересечение этого множества с множеством A оказывается пустым. Следовательно, множество всех точек, находящихся внутри обозначенной сплошной линией границы и на ней, образует дилатацию A по B. На рис. 9.6(*г*) изображен примитив, построенный с таким расчетом, чтобы добиться большей дилатации в вертикальном направлении, чем в горизонтальном. Рис. 9.6(*д*) показывает результат дилатации по такому примитиву.

Пример 9.2. Демонстрация применения дилатации.

■ Одно из простейших применений дилатации — устранение разрывов линий путем их перекрытия. На рис. 9.7(*а*) показано изображение с разорванными символами, которое уже приводилось на рис. 4.49 в связи с рассмотрением низкочастотной фильтрации. Известно, что максимальная длина разрывов составляет два пикселя. На рис. 9.7(*б*) показан простой примитив, которым можно воспользоваться для устранения разрывов (заметим, что здесь элементы примитива вместо затемнения обозначены единицами, а элементы фона — нулями, потому что примитив рассматривается как изображение, а не графический рисунок). Результат дилатации исходного изображения по такому примитиву представлен на рис. 9.7(*в*). Поверх разрывов образовались «мостики». Одно из непосредственных преимуществ морфологического подхода по сравнению с применявшимся в главе 4 методом устранения разрывов путем низкочастотной фильтрации состоит в том, что морфологический метод сразу приводит к двоичному изобра-

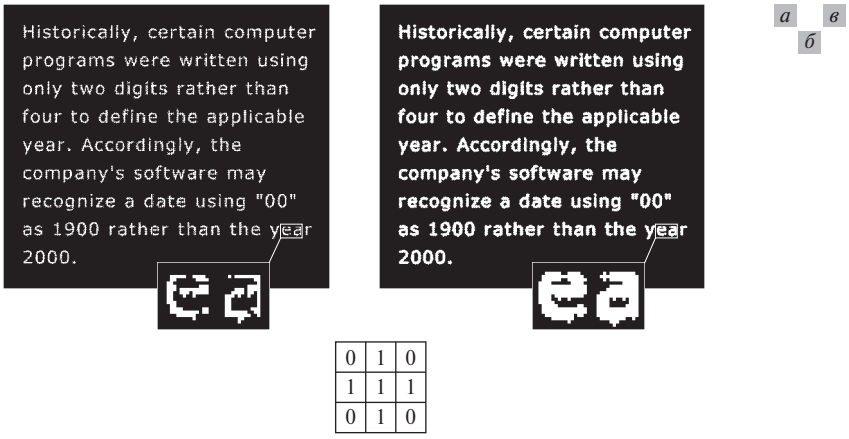


Рис. 9.7. (*а*) Пример изображения текста с недостаточным разрешением и разрывами букв (видны на увеличенном фрагменте). (*б*) Вид примитива. (*в*) Дилатация множества (*а*) по примитиву (*б*). Изолированные сегменты объединились

жению. Напротив, при низкочастотной фильтрации из исходного двоичного изображения вначале получается полутоновое, которое затем с помощью пороговой функции предстоит преобразовывать обратно в двоичное. ■

9.2.3. Двойственность

Эрозия и дилатация являются двойственными операциями по отношению к теоретико-множественным операциям дополнения и центрального отражения. Иначе говоря,

$$(A \ominus B)^c = A^c \oplus \hat{B} \tag{9.2-5}$$

и

$$(A \oplus B)^c = A^c \ominus \hat{B}. \tag{9.2-6}$$

Соотношение (9.2-5) показывает, что эрозия A по B совпадает с дополнением дилатации A^c по $\hat{B}$ и наоборот. Это свойство двойственности особенно полезно, когда примитив центрально симметричен относительно своего центра, как часто и бывает, так что $\hat{B} = B$. Таким образом, можно получить эрозию изображения по B, выполнив дилатацию его фона (т. е. дилатацию A^c) с тем же примитивом и взяв затем дополнение результата. Аналогичные комментарии справедливы для соотношения (9.2-6).

Проведем формальное доказательство соотношения (9.2-5), чтобы проиллюстрировать типичный способ проверки справедливости соотношений в морфологическом подходе. Начав с определения эрозии, имеем

$$(A \ominus B)^c = \left\{ z \,|\, (B)_z \subseteq A \right\}^c.$$

Раз множество $(B)_z$ содержится в множестве A, то $(B)_z \cap A^c = \varnothing$, и предыдущее равенство принимает вид

$$(A \ominus B)^c = \left\{ z \,|\, (B)_z \cap A^c = \varnothing \right\}^c.$$

Но *дополнением* для множества z, удовлетворяющим условию $(B)_z \cap A^c = \varnothing$, является множество таких z, что $(B)_z \cap A^c \neq \varnothing$. Поэтому

$$(A \ominus B)^c = \left\{ z \,|\, (B)_z \cap A^c \neq \varnothing \right\} = A^c \oplus \hat{B},$$

что и требовалось доказать (последний шаг следует из определения (9.2-3)). С помощью аналогичной цепочки рассуждений можно доказать (9.2-6) (см. задачу 9.13).

9.3. Размыкание и замыкание

Как мы видели, дилатация приводит к расширению деталей на изображении, а эрозия — к их сужению. В этом разделе рассматриваются еще две важные морфологические операции: размыкание и замыкание. В общем случае *размыкание*[4]

[4] В оригинале использованы термины *opening* («размыкание») и *closing* («замыкание»). — *Прим. перев.*

сглаживает контуры объекта, обрывает узкие перешейки и ликвидирует выступы небольшой ширины. *Замыкание* также приводит к сглаживанию участков контуров объектов, но, в отличие от размыкания, в общем случае «заливает» узкие промежутки и длинные углубления малой ширины, а также ликвидирует небольшие «дырки» и заполняет разрывы контура.

Размыкание множества A по примитиву B обозначается $A \circ B$ и определяется равенством

$$A \circ B = (A \ominus B) \oplus B. \tag{9.3-1}$$

Таким образом, размыкание множества A по примитиву B строится как эрозия A по B, результат которой затем подвергается дилатации по тому же примитиву B.

Аналогично замыкание множества A по примитиву B обозначается $A \bullet B$ и определяется как

$$A \bullet B = (A \oplus B) \ominus B, \tag{9.3-2}$$

то есть как дилатация множества A по B, за которой следует эрозия по тому же примитиву B.

Операция размыкания допускает простую геометрическую интерпретацию (рис. 9.8). Предположим, что примитив B имеет форму круга. Тогда *границу* множества $A \circ B$ образуют *максимально близкие* к границе A точки множества B при «обкатывании» примитивом B этой границы *изнутри*. Это геометрическое свойство *прилегания*, присущее операции размыкания, вытекает из теоретико-множественной формулировки, согласно которой размыкание множества A по B строится как объединение всех параллельных переносов B, которые укладываются в множество A. Таким образом, операция размыкания может быть выражена процессом «укладки» следующего вида:

$$A \circ B = \bigcup \left\{ (B)_z \middle| (B)_z \subseteq A \right\}, \tag{9.3-3}$$

где $\cup\{\cdot\}$ обозначает объединение всех множеств внутри фигурных скобок.

Замыкание допускает сходную геометрическую интерпретацию, но в этом случае примитив B обкатывает границу A снаружи (рис. 9.9). Ниже будет показано, что размыкание и замыкание являются двойственными операциями, так

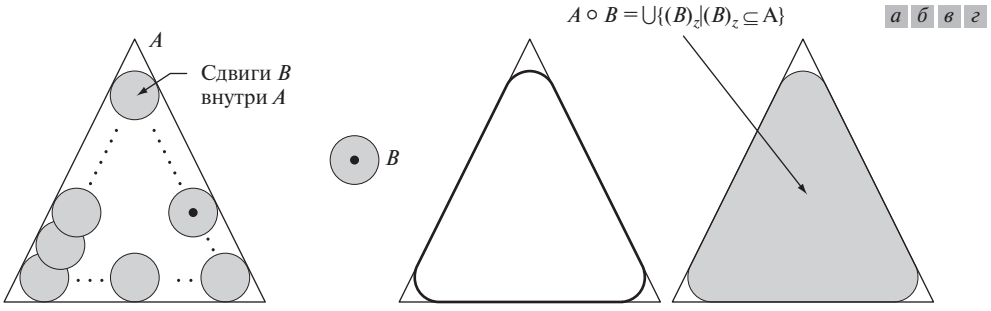


Рис. 9.8. (*а*) Примитив B, обкатывающий изнутри границу множества A. (*б*) Примитив B; положение его центра указано точкой. (*в*) Жирная линия показывает внешнюю границу размыкания. (*г*) Все множество, получаемое в результате размыкания (показано темным цветом)

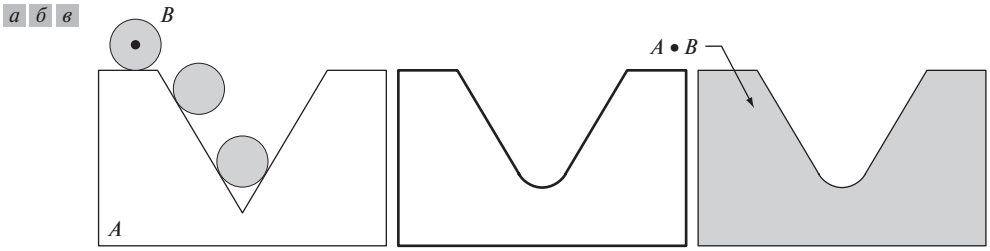


Рис. 9.9. (*а*) Примитив *B*, обкатывающий снаружи границу множества *A*. (*б*) Жирная линия показывает внешнюю границу замыкания. (*в*) Все множество, получаемое в результате замыкания (показано темным цветом). Исходное множество *A* на (*а*) не затемнено, чтобы не перегружать рисунок

что «обкатывание» с противоположной стороны границы (*снаружи*) не является чем-то удивительным. С геометрической точки зрения точка *w* является элементом множества $A \bullet B$ в том и только в том случае, если $(B)_z \cap A \neq \varnothing$ для любого сдвига $(B)_z$, накрывающего точку *w*. Рис. 9.9 иллюстрирует основные геометрические свойства замыкания.

Пример 9.3. Простая иллюстрация морфологических операций размыкания и замыкания.

■ Рис. 9.10 служит для дальнейшей иллюстрации операций замыкания и размыкания. На рис. 9.10(*а*) показано множество *A*, а на рис. 9.10(*б*) — различные положения круглого примитива в ходе операции эрозии. В результате выполнения этой операции образуются несвязные области, показанные на рис. 9.10(*в*). Обратите внимание на уничтожение мостика между двумя основными частями исходного объекта. Ширина мостика меньше диаметра примитива, т. е. данный примитив не может целиком поместиться в этой части множества *A*, и, значит, условие в (9.2-1) нарушается. То же самое справедливо для двух частей с правой стороны объекта, и выступающие элементы, куда не вмещается круг, также пропадают. Рис. 9.10(*г*) демонстрирует процесс дилатации множества, полученного в результате эрозии, а на рис. 9.10(*д*) приведен конечный результат размыкания. Видно, что обращенные наружу углы подверглись закруглению, тогда как обращенных внутрь углов это не коснулось.

Аналогично фигуры на рис. 9.10(*е*)—(*и*) иллюстрируют результаты замыкания множества *A* по тому же примитиву. Можно заметить, что в этом случае закругляются внутренние углы, а внешние углы остаются нетронутыми. Углубление, расположенное слева на границе множества *A*, значительно уменьшилось в размерах, поскольку в нем снаружи не помещается используемый круглый примитив. Обратите также внимание на сглаживание формы деталей в результате размыкания и замыкания исходного множества *A* с помощью круглого примитива. ■

Так же, как в случае дилатации и эрозии, операции размыкания и замыкания являются двойственными операциями по отношению к теоретико-множественным операциям дополнения и центрального отражения. Иначе говоря,

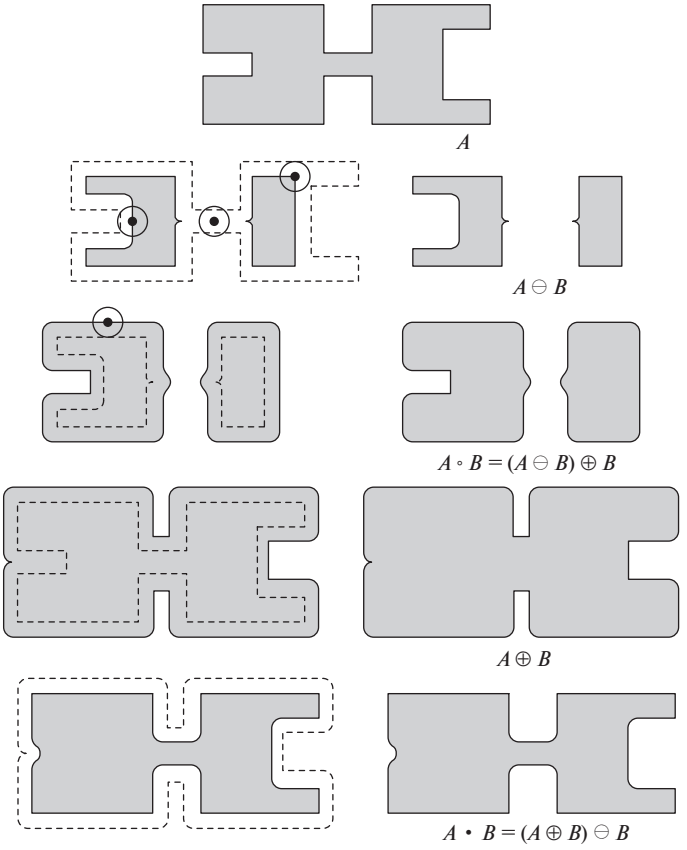


Рис. 9.10. Морфологические размыкание и замыкание. Используется примитив в форме небольшого круга, показанного в различных положениях на рисунках (*б*) и (*г*). Темная точка в центре круга указывает положение начала координат примитива (сам примитив не затемнен, чтобы не перегружать рисунок)

$$(A \bullet B)^c = A^c \circ \hat{B} \tag{9.3-4}$$

и

$$(A \circ B)^c = A^c \bullet \hat{B}. \tag{9.3-5}$$

Доказательство этих фактов оставляем в качестве самостоятельного упражнения (задача 9.14).

Операция размыкания обладает следующими свойствами:

(а) $A \circ B$ является подмножеством A (т. е. вложенным изображением).

(б) Если C есть подмножество D, то $C \circ B$ является подмножеством $D \circ B$.

(в) $(A \circ B) \circ B = A \circ B$.

Аналогично операция замыкания обладает следующими свойствами:

(а) A является подмножеством (вложенным изображением) $A \bullet B$.

(б) Если C есть подмножество D, то $C \bullet B$ является подмножеством $D \bullet B$.

(в) $(A \bullet B) \bullet B = A \bullet B$.

В обоих случаях свойства (*в*) означают, что многократные применения операций размыкания или замыкания к некоторому множеству не оказывают никакого действия после того, как соответствующая операция выполнена в первый раз.

Пример 9.4. Применение операций размыкания и замыкания для морфологической фильтрации.

■ Морфологические операции можно использовать для построения фильтров, похожих по своему принципу на пространственные фильтры, рассматривавшиеся в главе 3. На рис. 9.11(*а*) приведено двоичное изображение фрагмента отпечатка пальца, искаженное шумом, который проявляется в виде присутствующих на темном фоне светлых элементов, а также темных элементов на светлых полосах, составляющих отпечаток. Задача состоит в устранении шума при минимальном искажении формы отпечатка. Для решения этой задачи можно применить морфологический фильтр, выполняющий вначале операцию размыкания, а затем — замыкания.

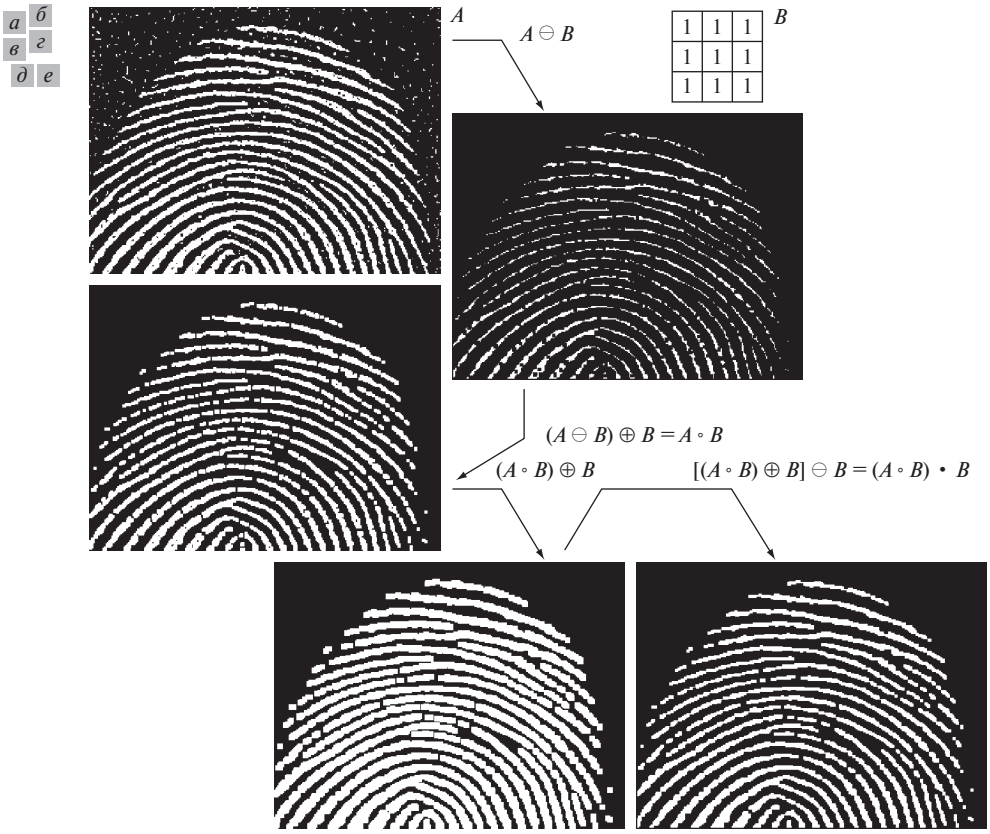


Рис. 9.11. (*а*) Зашумленное изображение. (*б*) Использующийся примитив. (*в*) Изображение после применения операции эрозии. (*г*) Результат размыкания исходного изображения. (*д*) Дилатация размыкания. (*е*) Замыкание размыкания. (Исходное изображение предоставлено Национальным институтом стандартов и технологии США)

Используется примитив, изображенный на рис. 9.11(*б*). Остальные части рисунка иллюстрируют последовательность шагов операции фильтрации. На рис. 9.11(*в*) показан результат эрозии множества *A* по указанному примитиву. Шум в фоновой области изображения полностью устраняется на этапе эрозии, входящем в состав операции размыкания, поскольку в рассматриваемом примере размеры всех составляющих шума меньше размеров примитива. Что касается шумовых компонент в виде темных пятен на отпечатке, то они увеличились в размерах. Это объясняется тем, что такие компоненты являются внутренними границами области, и они увеличиваются в размерах при выполнении операции эрозии. Такое расширение нейтрализуется путем дилатации, что приводит к результату, показанному на рис. 9.11(*г*). Видно, что шумовые составляющие на полосах отпечатка пальца уменьшились в размерах или полностью исчезли.

Две только что описанные операции вместе составляют операцию размыкания множества *A* по примитиву *B*. Из рис. 9.11(*г*) видно, что результирующий эффект от размыкания состоит в устранении практически всех шумовых составляющих как в фоновой области, так и на самом отпечатке. Однако в результате этой операции появились ранее отсутствовавшие пропуски на полосах отпечатка. Чтобы преодолеть этот нежелательный эффект, применим к результату размыкания операцию дилатации, как показано на рис. 9.11(*д*). В большинстве указанных промежутков целостность полос восстановилась, однако сами полосы стали шире. Такое увеличение можно скомпенсировать эрозией. Результат, показанный на рис. 9.11(*е*), получен путем применения операции замыкания к ранее построенному размыканию исходного множества, приведенному на рис. 9.11(*г*). Этот конечный результат примечателен отсутствием шумовых точек, хотя и имеет тот недостаток, что некоторые из полос отпечатка пальца восстановлены не полностью, а содержат разрывы. Этого можно было ожидать, поскольку в описываемую процедуру не было заложено никаких условий для сохранения связности. Мы вновь вернемся к этому вопросу в примере 9.8 и укажем пути его решения в разделе 11.1.7. ■

9.4. Преобразование «попадание/пропуск»

Морфологическое преобразование «попадание/пропуск»[5] является основным инструментом для обнаружения объектов определенных размеров и формы. Мы изложим его идею с помощью рис. 9.12, где приведено множество *A*, состоящее из трех несвязных частей (подмножеств), обозначенных *C, D* и *E*. Затемненные области на рис. 9.12(*а*)—(*в*) указывают исходные множества, тогда как темным цветом на рис. 9.12(*г*) и (*д*) показаны результаты выполнения морфологических операций. Задача состоит в том, чтобы найти местоположение одной из фигур, скажем, *D*.

Пусть начало координат (центр) каждой из фигур находится в ее центре тяжести. Опишем вокруг фигуры *D* несколько большее окно *W*. *Локальный фон* фигуры *D* по отношению к окну *W* определяется как разностное множество ($W \setminus D$),

[5] В оригинале использован термин *hit-or-miss* transform. — *Прим. перев.*

показанное на рис. 9.12(*б*). На рис. 9.12(*в*) изображено дополнение множества A, которое понадобится позже. Рис. 9.12(*г*) демонстрирует результат применения к A операции эрозии с использованием D в качестве примитива (пунктирными линиями для ясности показаны контуры исходных фигур). Напомним, что результатом эрозии A по D является множество таких положений *начала коор-*

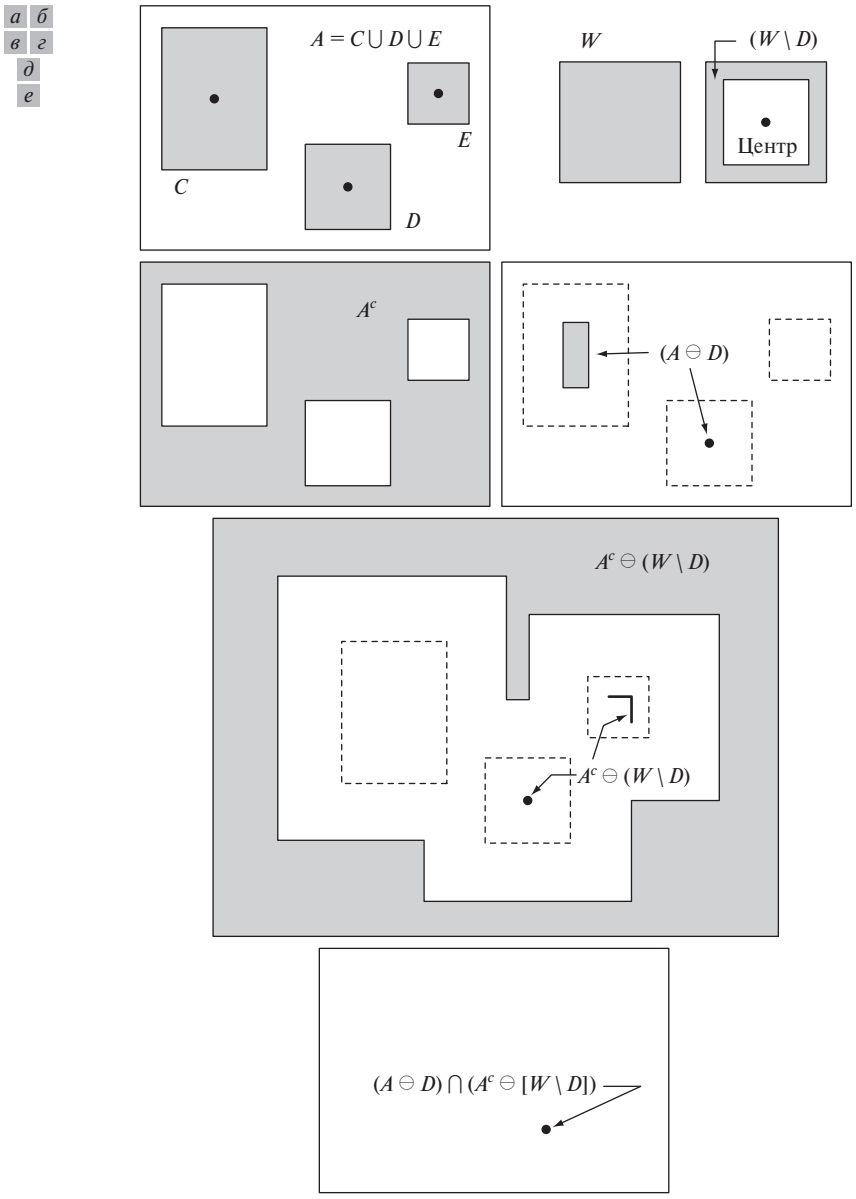


Рис. 9.12. (*а*) Множество A. (*б*) Окно W и локальный фон области D по отношению к W, $(W \setminus D)$. (*в*) Дополнение множества A. (*г*) Эрозия A по D. (*д*) Эрозия A^c по $(W \setminus D)$. (*е*) Пересечение множеств (*г*) и (*д*), которое дает искомое положение начала координат D

динат множества D, при которых D совпадает с некоторым подмножеством A. В другой интерпретации множество $A \ominus D$ можно рассматривать как геометрическое место точек положения начала координат D, для которых у D имеется эквивалент в A (т. е. D «попадает» в A). Будем помнить, что обсуждаемое на рис. 9.12 множество A состоит только из трех *непересекающихся* подмножеств C, D и E.

На рис. 9.12(*д*) показан результат применения к A^c (дополнению A) операции эрозии с использованием множества локального фона ($W \setminus D$) в качестве примитива (как уже отмечалось, предполагается, что A^c дополнено фоном и за пределами границ рисунка (*в*)). Внешняя область темного цвета на рис. 9.12(*д*) также является частью результата эрозии. Сопоставляя рис. 9.12(*г*) и (*д*), замечаем, что множество позиций *точного* местонахождения D в множестве A есть *пересечение* результата эрозии A по D и результата эрозии A^c по ($W \setminus D$), как показано на рис. 9.12(*е*). Это пересечение дает в точности искомое местоположение центра D. Другими словами, если множество, состоящее из объекта D и его окрестности, обозначить B, то вхождение (или набор вхождений) B в A, которое обозначается $A \circledast B$, описывается множеством

$$A \circledast B = (A \ominus D) \cap \left[A^c \ominus (W \setminus D) \right]. \tag{9.4-1}$$

Эта нотация допускает некоторое обобщение, если рассмотреть множество B как $B = (B_1, B_2)$, где B_1 — множество элементов B, относящихся к интересующему объекту, а B_2 есть множество элементов окружающего фона. Из предшествующего рассмотрения ясно, что $B_1 = D$ и $B_2 = (W \setminus D)$. С использованием таких обозначений равенство (9.4-1) принимает вид

$$A \circledast B = (A \ominus B_1) \setminus (A^c \ominus B_2). \tag{9.4-2}$$

Таким образом, множество $A \circledast B$ содержит все точки (положения центра), в которых одновременно для B_1 имеется эквивалент в A, а для B_2 — эквивалент в A^c (т. е. имеет место «попадание»). Применяя определение разности множеств (2.6-19) и соотношение двойственности между операциями эрозии и дилатации (9.2-5), можно переписать равенство (9.4-2) в виде

$$A \circledast B = \left(A \ominus B_1\right) \setminus \left(A \oplus \hat{B}_2\right). \tag{9.4-3}$$

Запись в форме (9.4-2) все же отличается значительно большей наглядностью. Любое из трех вышеприведенных соотношений мы будем называть *морфологическим преобразованием «попадание/пропуск»*.

Мотивировка для совместного использования примитива B_1, связанного с интересующим объектом, и примитива B_2, связанного с фоном, базируется на том соображении, что два или более объектов различимы только в том случае, если они образованы непересекающимися множествами. Это гарантируется, если потребовать, чтобы вокруг каждого объекта на изображении присутствовала область фона шириной как минимум один пиксель. В некоторых прикладных задачах интерес представляет обнаружение в исходном множестве заданных эталонов, т. е. сочетаний единиц и нулей. В подобных случаях понятие фона теряет смысл, и преобразование «попадание/пропуск» сводится к обычной операции эрозии. Как уже указывалось, эрозия дает лишь множество точек совпадения, однако в отсутствие дополнительного требования к совпадению фона это позволяет обнаруживать изолированные объекты. Такая упрощенная

схема обнаружения эталонов применяется в некоторых из алгоритмов, описываемых в следующем разделе.

9.5. Некоторые основные морфологические алгоритмы

Опираясь на результаты проведенного выше обсуждения, мы готовы теперь рассмотреть некоторые практические применения морфологических методов. При работе с двоичными изображениями морфология в основном применяется для извлечения компонент изображения, которые могут служить для представления и описания формы объектов. В частности, мы рассмотрим морфологические алгоритмы выделения границ, связных компонент, выпуклых оболочек и остовов областей. Будут также изложены некоторые методы, такие как заполнение, утончение, утолщение областей и отсечение ветвей, часто используемые в сочетании с упомянутыми алгоритмами в качестве шагов предварительной или финальной обработки. В этом разделе мы будем широко использовать небольшие модельные изображения, помогающие лучше понять механизм действия каждого описываемого морфологического процесса. Все эти изображения двоичные; темные клетки соответсуют единицам, а светлые — нулям.

9.5.1. Выделение границ

Граница множества A, которую будем обозначать $\beta(A)$, может быть выделена путем выполнения сначала операции эрозии A по B, а затем получения разностного множества между A и результатом его эрозии, т. е.

$$\beta(A) = A \setminus (A \ominus B), \tag{9.5-1}$$

где B — подходящий примитив (структурообразующее множество).

Рис. 9.13 иллюстрирует механизм выделения границы. Здесь приведены простой двоичный объект, примитив B и результат применения соотношения (9.5-1). Хотя показанный на рис. 9.13(*б*) примитив относится к числу наиболее часто используемых, он никоим образом не является единственно возможным. Например, если использовать заполненный единицами квадратный примитив размерами 5×5 пикселей, то это приведет к построению границы толщиной от 2 до 3 пикселей.

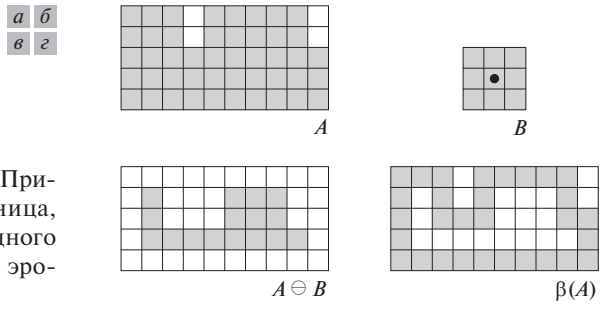


Рис. 9.13. (*а*) Множество A. (*б*) Примитив B. (*в*) Эрозия A по B. (*г*) Граница, полученная вычитанием из исходного множества A результата операции эрозии

Начиная с этого места, мы больше не показываем бордюр необходимой ширины, состоящий из фоновых элементов.

Пример 9.5. Выделение границы путем морфологической обработки.

■ Рис. 9.14 служит дальнейшей иллюстрацией использования операции (9.5-1) с примитивом размерами 3×3, заполненным единицами. Как и для всех двоичных изображений в этой главе, двоичной единице отвечает белый цвет, а двоичному нулю — черный, поэтому пиксели примитива (которые являются единичными) также считаются белыми. Благодаря размерам применяемого примитива выделяется граница толщиной в один пиксель, как показано на рис. 9.14(*б*). ■

9.5.2. Заполнение дырок

Дырку можно определить как область фона, окруженную связным бордюром из элементов переднего плана. В этом разделе мы изложим алгоритм заполнения дырок на изображении, основанный на операциях дилатации, дополнения и пересечения множеств. Пусть множество A состоит из точек 8-связных границ, внутри каждой из которых заключена фоновая область (т. е. имеется дырка). Задача состоит в том, чтобы, начав с заданной точки внутри каждой дырки, заполнить все дырки единичными значениями.

Начнем с построения массива X_0 (с такими же размерами, что и исходный массив, содержащий множество A), заполненного нулями во всех точках, кроме заданных точек в каждой дырке, которым присвоим значение 1. К заполнению всех дырок единицами приведет следующая рекуррентная процедура:

$$X_k = (X_{k-1} \oplus B) \cap A^c \qquad k = 1, 2, 3, \ldots, \tag{9.5-2}$$

а б

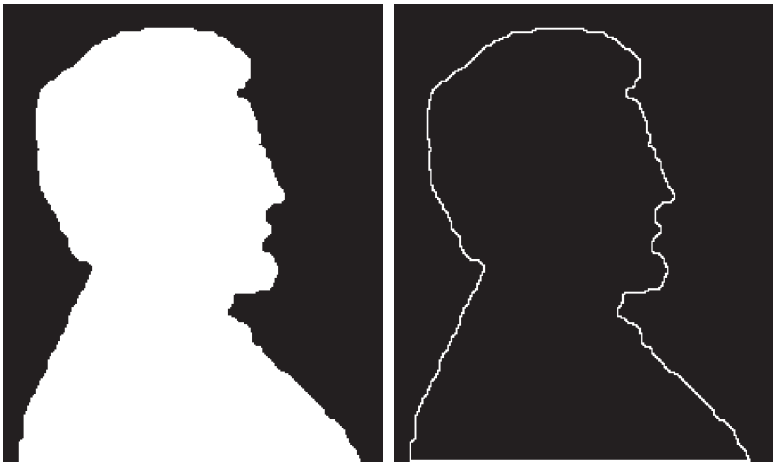


Рис. 9.14. (*а*) Простое двоичное изображение; белым цветом представлены единицы. (*б*) Результат применения операции (9.5-1) с примитивом на рис. 9.13(*б*)

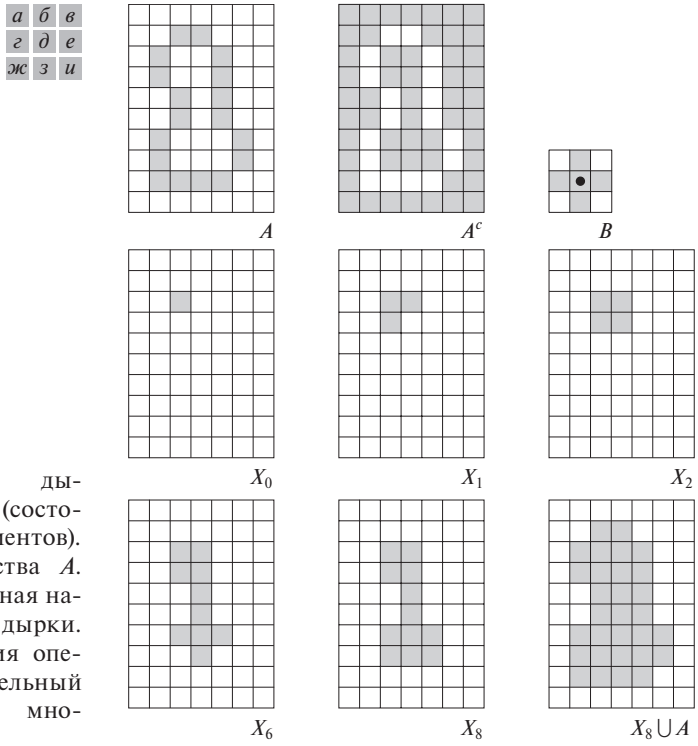


Рис. 9.15. Заполнение дырок. (*а*) Множество A (состоит из затемненных элементов). (*б*) Дополнение множества A. (*в*) Примитив B. (*г*) Заданная начальная точка внутри дырки. (*д*)—(*з*) Шаги применения операции (9.5-2). (*и*) Окончательный результат (объединение множеств (*а*) и (*з*))

где B — симметричный примитив, показанный на рис. 9.15(*в*). Данный алгоритм останавливается на k-м шаге итерации, когда $X_k = X_{k-1}$. При этом множество X_k содержит заполнение всех дырок. Объединение множеств X_k и A содержит исходные границы вместе со всеми заполненными дырками в них.

При отсутствии дополнительного контроля процесс дилатации (9.5-2) привел бы к заполнению единицами всего поля изображения. Однако взятие на каждом шаге пересечения с A^c ограничивает результат операции только внутренностью интересующей области. Это наш первый пример того, как можно ограничить морфологическую обработку для достижения некоторого желаемого свойства. В рассматриваемом приложении этот прием уместно назвать *условной дилатацией*. Остальные примитивы рис. 9.15 иллюстрируют фазы применения операции (9.5-2). Хотя в этом примере присутствует только одна дырка, ясно, что тот же принцип может применяться для любого конечного числа дырок при условии, что заданы начальные точки внутри каждой из них.

Пример 9.6. Морфологическое заполнение дырок.

■ На рис. 9.16(*а*) приведено изображение, состоящее из белых кругов с черными пятнами внутри. Подобное изображение могло быть получено в результате порогового преобразования изображения сцены, в которой присутствуют полированные сферические объекты (например подшипниковые шарики). Темные пятна внутри кругов могли возникнуть в результате бликов. Задача состоит в устранении этих бликов путем заполнения дырок. На рис. 9.16(*а*) нанесена точка, выбранная внутри одного из кругов, а на рис. 9.16(*б*) показан результат запол-

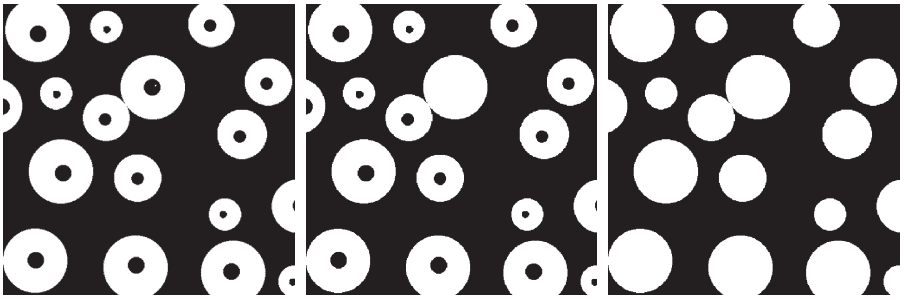


Рис. 9.16. (*а*) Двоичное изображение (белая точка внутри одной из областей — начальная точка для алгоритма заполнения дырок). (*б*) Результат заполнения этой области. (*в*) Результаты заполнения всех дырок

нения соответствующей области. Наконец, рис. 9.16(*в*) демонстрирует результат заполнения всех таких областей. Поскольку в данном случае необходимо знать, относятся ли черные точки к фону изображения или являются внутренними точками области, для полной автоматизации процедуры необходимо внести в алгоритм дополнительный «интеллект». В разделе 9.5.9 будет изложен полностью автоматический метод, основанный на морфологической реконструкции. (См. также задачу 9.23.) ■

9.5.3. Выделение связных компонент

Понятия связности и связных компонент были введены в рассмотрение в разделе 2.5.2. Выделение компонент связности на двоичном изображении занимает центральное место во многих прикладных задачах анализа изображений. Пусть A — множество, содержащее одну или более связных компонент, и предположим, что в каждой из них известна ровно одна точка. Сформируем массив X_0 (с теми же размерами, что и исходный массив, содержащий множество A), все элементы которого равны 0 (значение фона), за исключением известных точек внутри связных компонент, которым присвоим значение 1 (значение элементов компонент переднего плана). Задача состоит в том, чтобы, начав с X_0, найти все компоненты связности. Решение находится с помощью следующего рекуррентного алгоритма:

$$X_k = (X_{k-1} \oplus B) \cap A \qquad k = 1, 2, 3, \ldots, \tag{9.5-3}$$

где B — подходящий примитив, например изображенный на рис. 9.17(*б*). Алгоритм сходится, когда $X_k = X_{k-1}$; при этом X_k содержит все связные компоненты исходного изображения. Соотношение (9.5-3) по форме схоже с (9.5-2); единственное отличие состоит в использовании самого множества A вместо его дополнения A^c. Это не удивительно, потому что здесь мы отыскиваем элементы переднего плана, тогда как в разделе 9.5.2 задача состояла в поиске точек фона.

Механизм действия соотношения (9.5-3) проиллюстрирован на рис. 9.17, где сходимость достигается при $k = 6$. Заметим, что форма примитива предполагает наличие 8-связности между пикселями одной связной компоненты[6].

[6] Иначе говоря, области, соприкасающиеся по диагонали, включаются в одну компоненту связности. — *Прим. перев.*

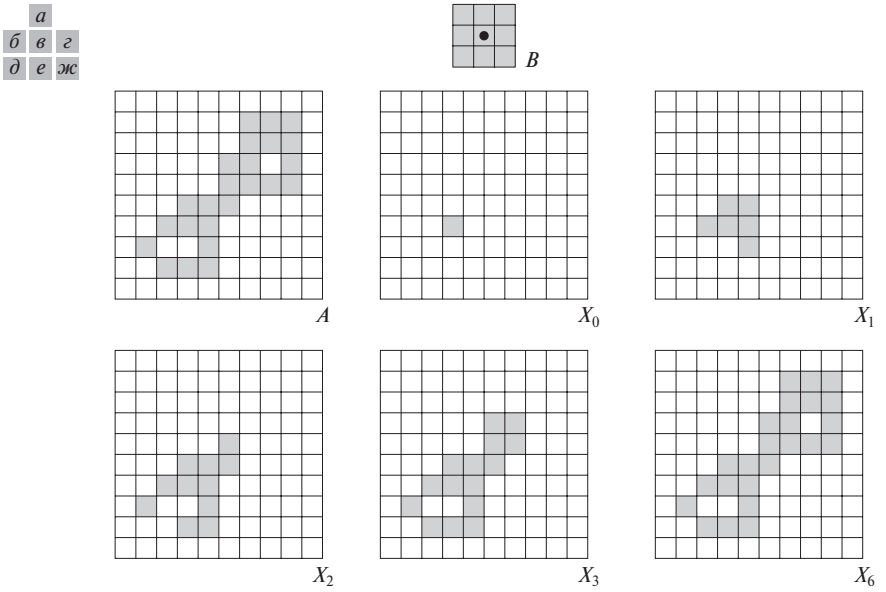


Рис. 9.17. Выделение связных компонент. (*а*) Используемый примитив. (*б*) Массив, содержащий множество с одной связной компонентой. (*в*) Начальный массив, содержащий единичный элемент в известной точке связной компоненты. (*г*)—(*ж*) Результаты последовательных шагов итерации согласно (9.5-3)

Если бы использовался примитив на рис. 9.15, опирающийся на 4-связность, то не был бы выделен самый левый элемент связной компоненты в нижней части изображения, поскольку он характеризуется 8-связностью с остальной частью фигуры. Как и для алгоритма заполнения дырок, процедура (9.5-3) применима также в случае любого конечного числа связных компонент, из которых состоит множество A, при условии, что внутри каждой компоненты связности известна хотя бы одна точка.

> См. алгоритм в задаче 9.24, для которого не требуется изначально знать точку в каждой компоненте связности.

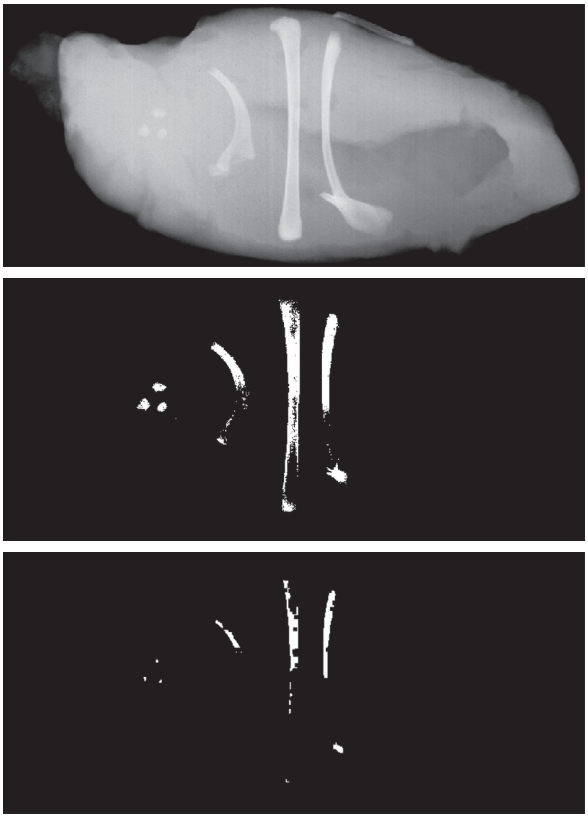
Пример 9.7. Применение алгоритма выделения связных компонент для обнаружения инородных объектов в упакованных пищевых продуктах.

■ Выделение связных компонент изображения часто применяется для автоматического контроля. На рис. 9.18(*а*) приведено рентгеновское изображение куриного филе, в котором присутствуют фрагменты костей. Значительный интерес представляет задача обнаружения подобных инородных объектов в ходе обработки пищевых продуктов, до их упаковки и/или поставки. В данном конкретном случае, благодаря слабой проницаемости костей для рентгеновских лучей, соответствующие пиксели заметно отличаются по уровню яркости от фона изображения, что делает выделение костных фрагментов из фона простой задачей,

решаемой с помощью порогового преобразования (пороговое преобразование было введено в разделе 3.1 и более подробно рассматривается в разделе 10.3). В результате его применения получается двоичное изображение, показанное на рис. 9.18(*б*).

Наиболее существенной характеристикой данного рисунка является тот факт, что точки, оставшиеся после выполнения преобразования, сконцентрированы в виде объектов (костей), а не превратились в отдельные несвязанные фрагменты. Мы можем оставить только объекты «значительных» размеров, применив к полученному двоичному изображению преобразование эрозии. В данном примере значительными признаются любые объекты, не исчезающие после эрозии по заполненному единицами квадратному примитиву размерами 5×5 пикселей. Результат применения такого преобразования показан на рис. 9.18(*в*). Следующий шаг состоит в анализе размеров оставшихся объектов, которые опознаются путем выделения связных компонент на изображении. Результаты выделения

а
б
в *г*



Номер связной компоненты	Число пикселей в связной компоненте
01	11
02	9
03	9
04	39
05	133
06	1
07	1
08	743
09	7
10	11
11	11
12	9
13	9
14	674
15	85

Рис. 9.18. (*а*) Рентгеновское изображение куриного филе с фрагментами костей. (*б*) Двоичное изображение после порогового преобразования. (*в*) Результат эрозии изображения по квадратному примитиву размерами 5×5, заполненному единицами. (*г*) Число пикселей в связных компонентах изображения (*в*). (Изображение предоставлено компанией NTB Elektronische Geraete GmbH, Diepholz, Германия; адрес в Интернете www.ntbxray.com)

сведены в таблицу на рис. 9.18(*г*). Видно, что всего имеется 15 компонент связности, четыре из которых доминируют по площади. Этой информации достаточно, чтобы сделать вывод о присутствии нежелательных объектов в исходном изображении. При необходимости можно определить дальнейшие характеристики объектов (например форму), для чего используются методы, описанные в главе 11. ■

9.5.4. Выпуклая оболочка

Множество A называется *выпуклым*[7], если отрезок прямой, соединяющий любые две точки A, целиком лежит внутри A. *Выпуклая оболочка* H произвольного множества S — это наименьшее выпуклое множество, содержащее S. Разность множеств $H \setminus S$ называется *дефицитом выпуклости* S. Как будет видно из подробного рассмотрения в разделах 11.1.6 и 11.3.2, выпуклая оболочка и дефицит выпуклости полезны для построения описания объектов. Здесь мы представим простой морфологический алгоритм построения выпуклой оболочки $C(A)$ исходного множества A.

Обозначим B^i, $i = 1, 2, 3, 4$ четыре примитива, показанные на рис. 9.19(*а*). Описываемая процедура состоит в применении рекуррентного соотношения

$$X_k^i = (X_{k-1}^i \circledast B^i) \cup X_{k-1}^i \quad i = 1, 2, 3, 4 \quad \text{и} \quad k = 1, 2, 3, \ldots \tag{9.5-4}$$

для $X_0^i = A$. В момент сходимости этой процедуры, т. е. когда $X_k^i = X_{k-1}^i$, положим $D^i = X_k^i$. Тогда выпуклая оболочка A есть множество

$$C(A) = \bigcup_{i=1}^{4} D^i . \tag{9.5-5}$$

Другими словами, метод состоит в итеративном применении к множеству A преобразования «попадание/пропуск» по примитиву B^1 и объединении с ранее полученным результатом. Эти шаги повторяются до тех пор, пока происходят изменения; результат обозначается D^1. Затем такие же действия повторяются с применением к A преобразования по примитиву B^2 и так далее. Объединение четырех полученных множеств D^i дает в результате выпуклую оболочку для множества A. Отметим, что здесь используется упрощенная реализация преобразования «попадание/пропуск», в которой не требуется сравнивать фон, о чем уже шла речь в конце раздела 9.4.

Рис. 9.19 иллюстрирует процедуру, задаваемую соотношениями (9.5-4) и (9.5-5). На рис. 9.19(*а*) показаны примитивы, используемые для построения

---

[7] Данное определение, строго говоря, относится к множествам, являющимся подмножествами евклидова пространства, и не может быть прямо перенесено на дискретные множества (подмножества Z^2). В основу определения выпуклости в дискретном случае можно положить следующее верное для непрерывного случая основное свойство: выпуклое множество совпадает со своей выпуклой оболочкой. Поэтому если вначале конструктивно определить выпуклую оболочку, как это делают авторы, то выпуклыми можно затем объявить те множества, которые совпадают со своей выпуклой оболочкой. Отметим, что такое понятие выпуклости зависит от той процедуры, которая используется для построения выпуклой оболочки. — *Прим. перев.*

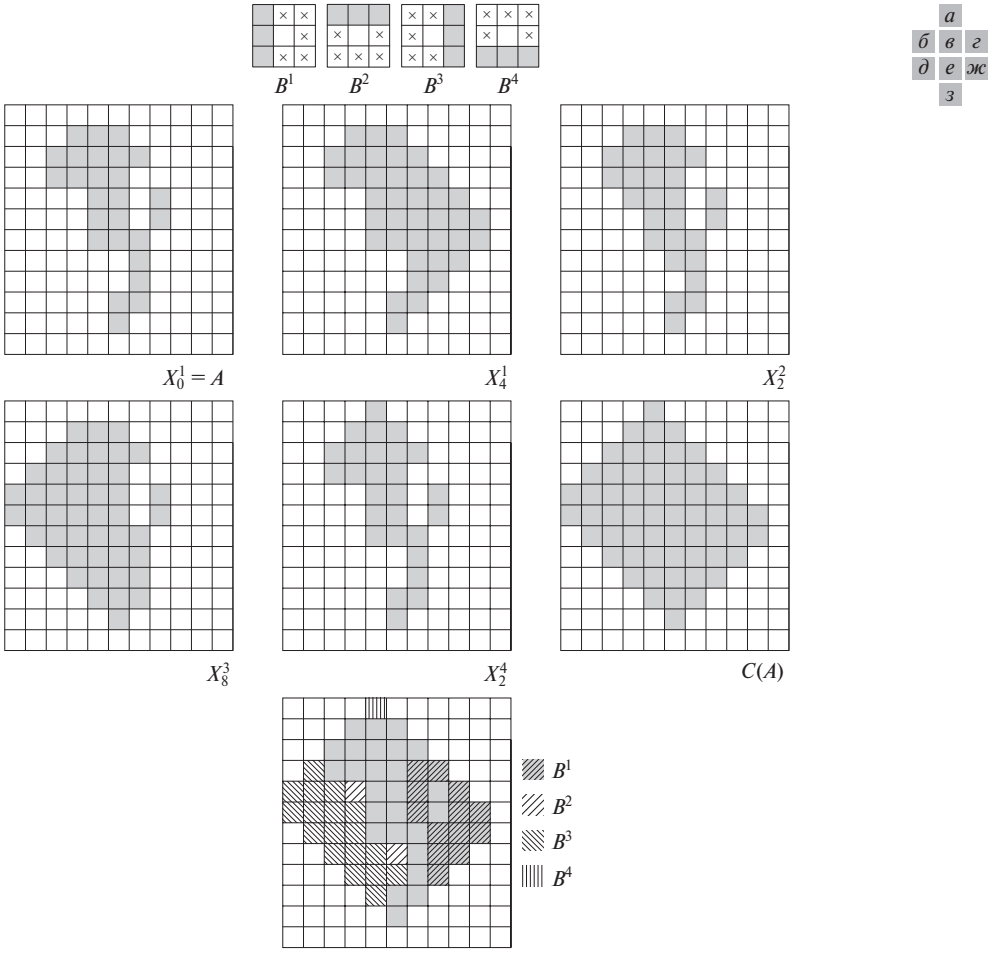


Рис. 9.19. (*а*) Используемые примитивы. (*б*) Множество *A*. (*в*)—(*е*) Результаты сходимости процедуры для каждого из примитивов (*а*). (*ж*) Выпуклая оболочка. (*з*) Вклад каждого примитива в формирование выпуклой оболочки

выпуклой оболочки. Начало координат каждого примитива находится в его центре. Знаки «×» на элементах примитивов указывают, что значения этих пикселей не важны. Иначе говоря, эквивалентом в множестве *A* для данного примитива признается такая область с размерами 3×3 пикселя, значения в которой совпадают с конфигурацией нулей (белое поле) и единиц (темное поле) в этом примитиве, вне зависимости от значений пикселей на позициях, отмеченных знаком «×». Конкретно для каждого примитива используется маска с нулем в центре и тремя единицами вдоль одной из сторон квадрата, а остальные пиксели не учитываются. С учетом обозначений на рис. 9.19(*а*) примитив B^i получается из B^{i-1} поворотом на угол 90° по часовой стрелке.

На рис. 9.19(*б*) показано множество *A*, для которого требуется построить выпуклую оболочку. Начиная с $X_0^1 = A$, после четырех итераций применения соотношения (9.5-4) приходим к множеству, показанному на рис. 9.19(*в*). Затем

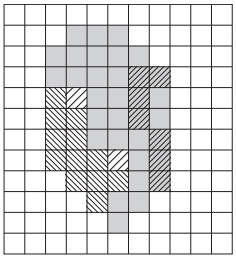


Рис. 9.20. Результат работы алгоритма выделения выпуклой оболочки с ограничением роста области максимальными размерами исходного множества по вертикали и горизонтали

полагаем $X_0^2 = A$ и вновь с помощью (9.5-4) приходим к множеству, показанному на рис. 9.19(*г*) (в этом случае сходимость происходит уже за два шага). Следующие два результата получаются аналогичным образом. В конце строится объединение множеств из рис. 9.19(*в*), (*г*), (*д*) и (*е*), которое и является искомой выпуклой оболочкой (рис. 9.19(*ж*)). На рис. 9.19(*з*) обозначен вклад, который вносит в это составное множество каждый из применявшихся примитивов.

Очевидный недостаток только что описанной процедуры состоит в том, что построенная выпуклая оболочка может превысить размеры, минимально необходимые для выполнения условия выпуклости. Простой способ ослабить данный эффект состоит в ограничении роста области, чтобы она не превышала вертикальный и горизонтальный размеры исходного множества точек. Если наложить такое ограничение на описанный алгоритм, то преобразованием множества на рис. 9.19 будет выпуклая оболочка, изображенная на рис. 9.20. Для ограничения роста выпуклой оболочки в изображениях с бóльшим числом деталей могут применяться граничные условия более сложного вида. Например, можно дополнительно учитывать максимальные размеры исходного множества в диагональных направлениях. Такое уточнение границ выпуклой оболочки достигается ценой дополнительного усложнения алгоритма (и соответствующего увеличения вычислительной сложности).

9.5.5. Утончение

Утончение множества A по примитиву B, обозначаемое $A \otimes B$, можно определить с помощью преобразования «попадание / пропуск»:

$$A \otimes B = A \setminus (A \circledast B) = A \cap (A \circledast B)^c . \qquad (9.5\text{-}6)$$

Как и в предыдущем разделе, интерес представляет только поиск в исходном множестве мест совпадения с конфигурацией пикселей примитива, так что в преобразовании «попадание/пропуск» не участвует окружающий фон. Более полезное выражение для симметричного утончения множества A основано на использовании *последовательности* примитивов

$$\{B\} = \{B^1, B^2, B^3, \ldots, B^n\}, \qquad (9.5\text{-}7)$$

где B^i получается из B^{i-1} поворотом. На этой основе можно определить операцию утончения по последовательности примитивов следующим образом:

$$A \otimes \{B\} = \left(\left(\ldots \left(\left(A \otimes B^1 \right) \otimes B^2 \right) \ldots \right) \otimes B^n \right). \qquad (9.5\text{-}8)$$

Процесс состоит в том, что *на первом проходе* производится утончение множества A по примитиву B^1, затем полученный результат подвергается утончению по примитиву B^2 за второй проход и так далее вплоть до n-го прохода с примитивом B^n. Весь этот процесс повторяется до тех пор, пока наблюдаются изменения. Каждый шаг утончения выполняется с помощью соотношения (9.5-6).

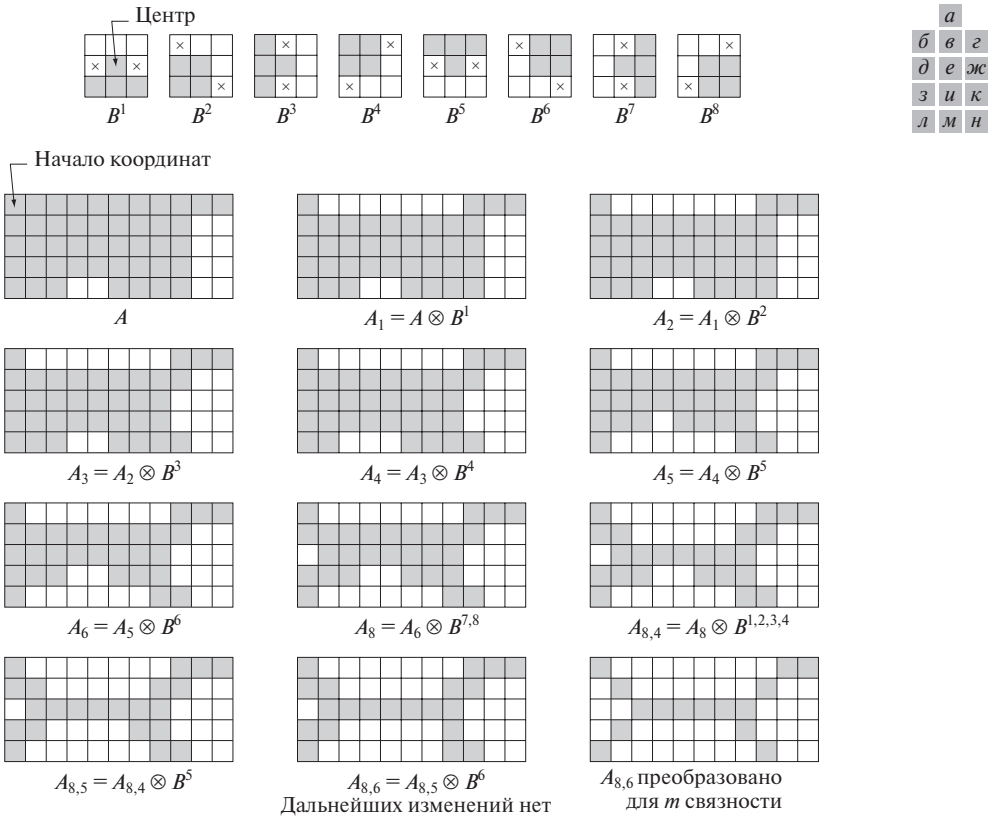


Рис. 9.21. (*а*) Последовательность поворотов примитивов, используемых для утончения. (*б*) Множество *A*. (*в*) Результат утончения по первому примитиву. (*г*)—(*и*) Результаты утончения по следующим семи примитивам (при смене 7-го примитива на 8-й изменений не происходит). (*к*) Результат повторного применения первых четырех примитивов. (*м*) Результат после сходимости алгоритма. (*н*) Преобразование с учетом требования *m*-связности

На рис. 9.21(*а*) приведен набор примитивов, обычно применяемых для утончения, а рис. 9.21(*б*) демонстрирует пример множества *A*, которое подвергается утончению с помощью описанной выше процедуры. На рис. 9.21(*в*) показан результат утончения после одного сканирующего прохода по множеству *A* с помощью примитива B^1, а на рис. 9.21(*г*)—(*л*) — результаты проходов с помощью других примитивов. Сходимость достигается после второго прохода с примитивом B^6. Рис. 9.21(*м*) демонстрирует результат утончения. В заключение на рис. 9.21(*н*) показан результат преобразования утонченного множества с учетом требования *m*-связности (см. раздел 2.5.2), чтобы исключить неоднозначность внутренних путей.

9.5.6. Утолщение

Утолщение представляет собой двойственную морфологическую операцию по отношению к утончению, определяемую выражением

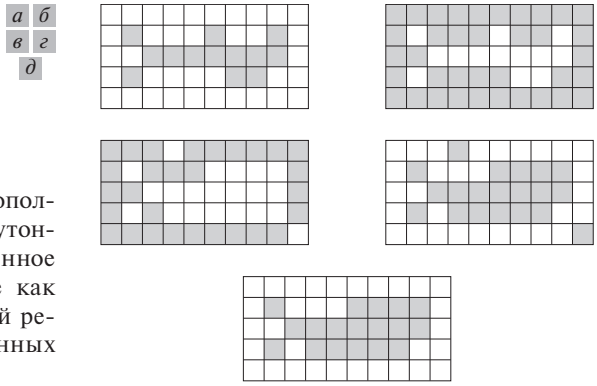


Рис. 9.22. (*а*) Множество *A*. (*б*) Дополнение множества *A*. (*в*) Результат утончения дополнения *A*. (*г*) Утолщенное исходное множество, полученное как дополнение (*в*). (*д*) Окончательный результат после удаления изолированных точек

$$A \odot B = A \cup (A \circledast B), \tag{9.5-9}$$

где B — подходящий примитив. Как и в случае утончения, утолщение может определяться как последовательная операция:

$$A \odot \{B\} = \left(\left(\ldots \left(\left(A \odot B^1 \right) \odot B^2 \right) \ldots \right) \odot B^n \right). \tag{9.5-10}$$

Используемые для утолщения примитивы имеют ту же форму, что и показанные на рис. 9.21(*а*) для операции утончения, но единицы и нули внутри примитивов меняются местами. Однако специальный алгоритм утолщения редко реализуется на практике. Вместо этого обычно применяется процедура, состоящая в утончении фона для рассматриваемого множества, а затем берется дополнение полученного результата. Иными словами, для утолщения множества A сначала строится множество $C = A^c$, проводится его утончение, а затем строится C^c. Рис. 9.22 иллюстрирует такую процедуру.

В зависимости от вида исходного множества A, описанная процедура может привести к появлению отдельных изолированных точек, как видно на рис. 9.22(*г*). Поэтому вслед за таким методом утолщения обычно применяется финальная обработка, цель которой состоит в удалении изолированных точек. На рис. 9.22(*в*) можно заметить, что фон после утончения образует границу для утолщенного исходного множества. Это полезное качество не проявляется при непосредственной реализации утолщения с помощью соотношения (9.5-10), что является одной из главных причин, почему утолщение обычно осуществляется посредством утончения фона.

9.5.7. Построение остова

Как видно из рис. 9.23, понятие остова $S(A)$ множества A является простым и интуитивно понятным. Опираясь на этот рисунок, можно сделать следующие заключения:

а) Если точка z принадлежит $S(A)$ и $(D)_z$ — наибольший круг с центром в z, целиком содержащийся в A, то не существует круга с бóльшим диаметром (не обязательно с центром в точке z), который бы включал в себя $(D)_z$ и при этом содержался в A. Такой круг $(D)_z$ называется *максимальным кругом*.

б) Круг $(D)_z$ касается границы множества A в двух или более различных точках.

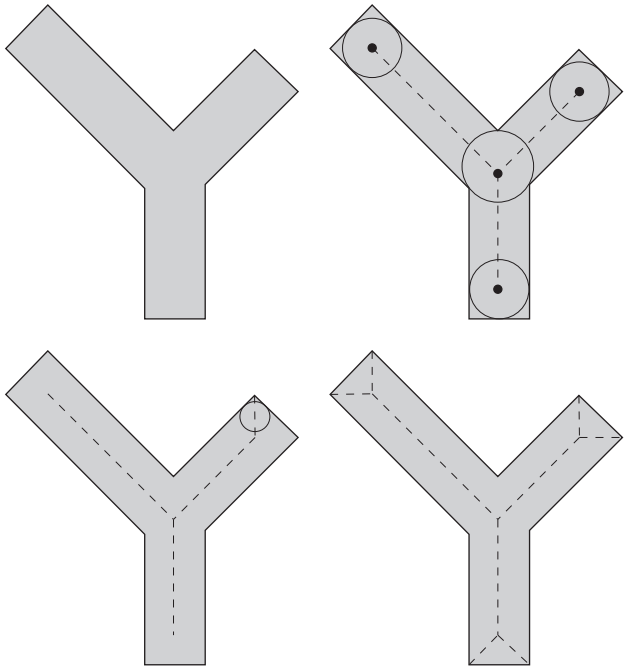


Рис. 9.23. (*а*) Множество *A*. (*б*) Различные положения максимальных кругов с центрами на остове *A*. (*в*) Добавочные максимальные круги на других отрезках остова *A*. (*г*) Полный остов (показан пунктиром)

Множество точек, образующих остов *A*, может быть выражено с использованием операций эрозии и размыкания. Так, можно показать ([Serra, 1982]), что

$$S(A)=\bigcup_{k=1}^{K}S_k(A) \tag{9.5-11}$$

при

$$S_k(A)=(A\ominus kB)\setminus((A\ominus kB)\circ B), \tag{9.5-12}$$

где *B* — примитив, а запись $(A \ominus kB)$ обозначает *k* последовательных применений операции эрозии к множеству *A*:

$$(A\ominus kB)=\big(\big(\ldots(A\ominus B)\ominus B\big)\ominus\ldots\big)\ominus B \tag{9.5-13}$$

(операция $\ominus$ повторяется *k* раз)[8]. Символ *K* в (9.5-11) — номер последнего шага итерации перед превращением множества *A* в пустое множество. Другими словами,

$$K=\max\{k|(A\ominus kB)\neq\varnothing\}. \tag{9.5-14}$$

Формулы (9.5-11) и (9.5-12) утверждают, что остов $S(A)$ можно получить объединением *подмножеств остова* $S_k(A)$. Кроме того, можно показать, что

[8] Полезно обратить внимание на тот факт, что $S_k(A)$ есть то множество точек, которые *не восстанавливаются* операцией дилатации из множества $(A \ominus (k + 1)B)$. Это позволит легко понять уравнение (9.5-15). — *Прим. перев.*

множество A может быть *восстановлено* по этим подмножествам с помощью операции

$$A = \bigcup_{k=0}^{K} \left(S_k(A) \oplus kB \right), \tag{9.5-15}$$

где $(S_k(A) \oplus kB)$ обозначает последовательное применение дилатации к $S_k(A)$ k раз, т. е.

$$(S_k(A) \oplus kB) = \left((\ldots (S_k(A) \oplus B) \oplus B) \oplus \ldots \right) \oplus B. \tag{9.5-16}$$

Пример 9.8. Построение остова простой фигуры.

■ Рис. 9.24 иллюстрирует описанный подход. В первой колонке сверху вниз показаны исходное множество и результаты двух последовательных применений к нему операции эрозии по примитиву B. Заметим, что при следующем приме-

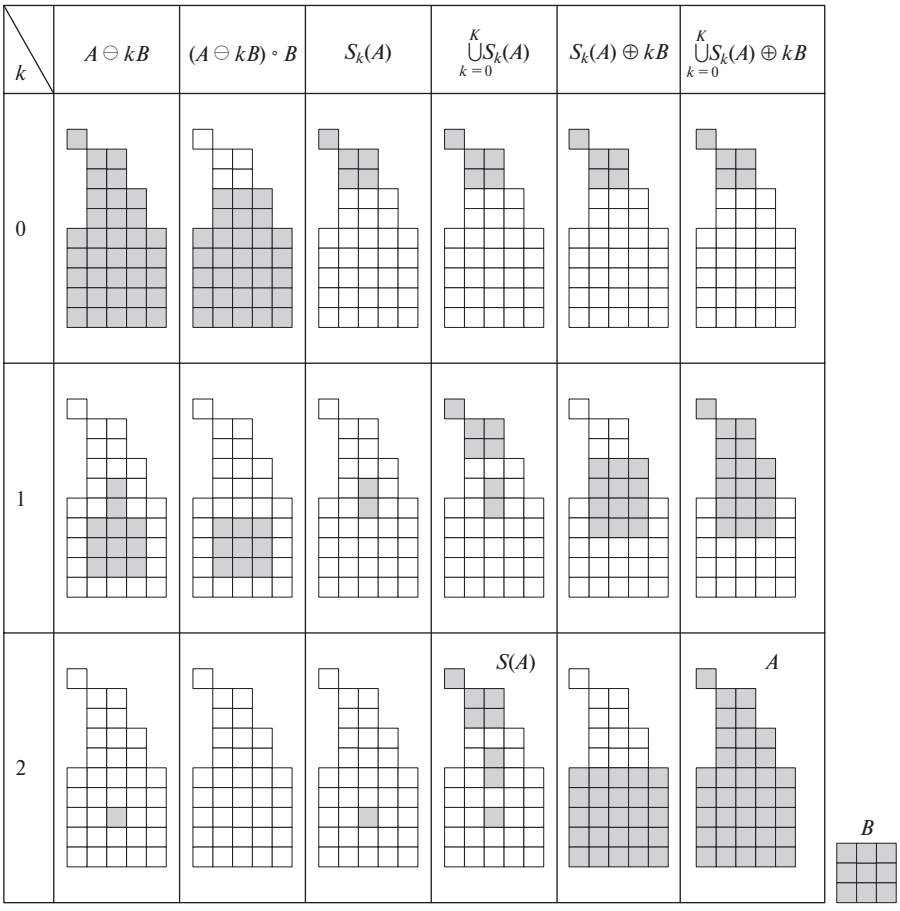


Рис. 9.24. Реализация соотношений (9.5-11)—(9.5-15). Исходное изображение находится слева вверху, а его морфологический остов — в четвертой колонке внизу. Восстановленное множество приведено в шестой колонке внизу

нении эрозии будет получено пустое множество, поэтому в данном случае $K = 2$. Во второй колонке показаны результаты применения к множеству из первой колонки операции размыкания по тому же примитиву B. Эти результаты легко объясняются характерным для операции размыкания свойством прилегания, о котором говорилось в связи с рис. 9.8. В третьей колонке содержится просто разность множеств из первой и второй колонок.

Четвертая колонка содержит (сверху вниз) две части остова и окончательный результат его построения. Этот финальный остов не только толще, чем хотелось бы, но и не является связным, что более существенно. Такой результат вполне закономерен, поскольку в предшествующем построении морфологического остова не было ничего, что гарантировало бы связность. Морфологический подход приводит к красивой формулировке в терминах эрозии и размыкания данного множества. Однако если необходимо построить максимально тонкий, связный и максимально гладкий остов (что обычно и требуется на практике), то приходится прибегать к эвристическим построениям, например подобным алгоритму из раздела 11.1.7. В пятой колонке показаны множества $S_0(A)$, $S_1(A) \oplus B$ и $S_2(A) \oplus 2B = (S_2(A) \oplus B) \oplus B$. Наконец, в шестой колонке показан процесс восстановления множества A, который, согласно соотношению (9.5-15), состоит в объединении подвергнутых дилатации подмножеств остова из пятой колонки. ■

9.5.8. Усечение

Методы усечения являются существенным дополнением алгоритмов утончения и построения остова, поскольку эти алгоритмы склонны оставлять паразитные компоненты, которые необходимо «вычистить» в ходе финальной обработки. Мы начнем обсуждение с задачи усечения областей (или отсечения ложных ветвей), а затем выработаем в рамках морфологического подхода решение, основанное на материале предыдущих разделов. Таким образом, мы воспользуемся этой возможностью, чтобы проиллюстрировать, как приступать к решению поставленной конкретной задачи путем комбинирования методов, рассмотренных к данному моменту.

Общепринятый подход к задаче автоматического распознавания рукописных знаков основан на анализе формы остова каждого символа. Такие остовы часто характеризуются наличием паразитных «отростков». Они могут возникать в результате операции эрозии из-за неоднородностей линий, из которых состоит знак. Исходя из предположения, что длина паразитных составляющих не превышает известного числа пикселей, мы построим морфологический метод для их устранения.

На рис. 9.25(*a*) приведен остов рукописной буквы «а». Паразитная составляющая в левой верхней части символа является типичным примером объекта, который хотелось бы удалить. Решение базируется на подавлении паразитной ветви путем последовательного удаления ее концевой точки. Разумеется, при этом также сокращаются (или пропадают) и другие ветви символа, однако, за неимением другой информации о структуре, в данном примере предполагается, что удалению подлежит любая ветвь, состоящая из трех пикселей или

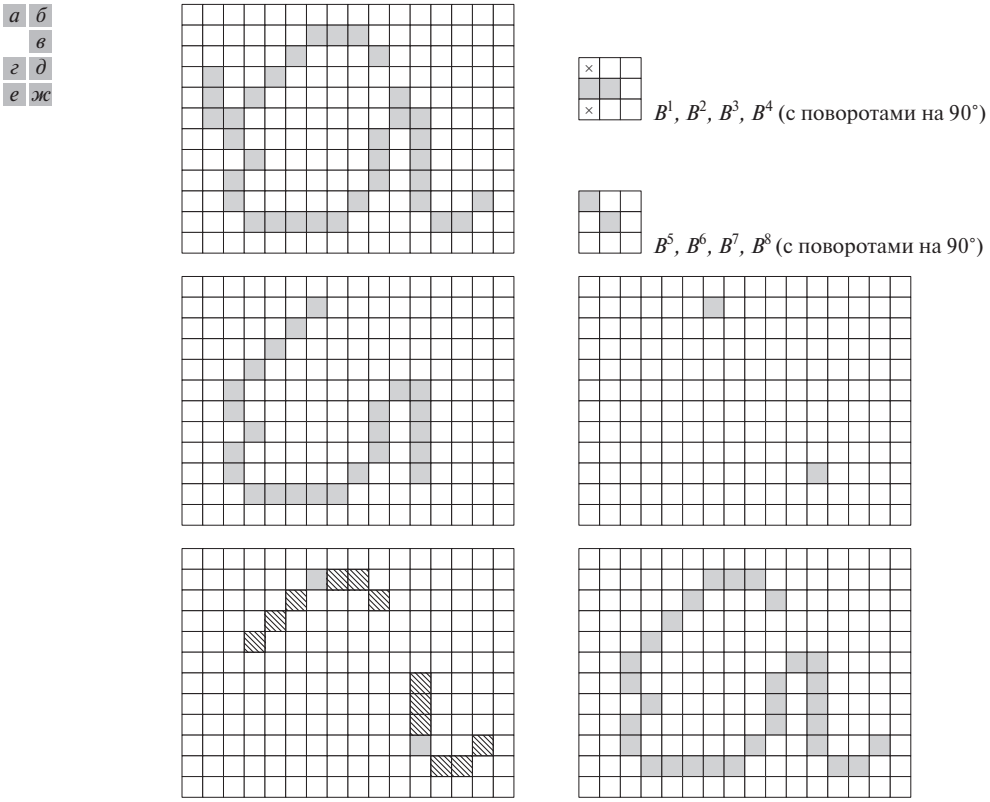


Рис. 9.25. (*а*) Исходное изображение. (*б*) и (*в*) Примитивы для удаления концевых точек. (*г*) Результат трех циклов утончения. (*д*) Концевые точки множества (*г*). (*е*) Дилатация концевых точек при условии принадлежности множеству (*а*). (*ж*) Изображение после усечения

менее. Желаемый результат достигается путем утончения множества A с помощью последовательности примитивов, построенных для обнаружения только концевых точек. Итак, пусть

Можно определить концевую точку как центр квадрата 3×3, соответствующего любой из конфигураций пикселей на рис. 9.25(*б*) или (*в*).

$$X_1 = A \otimes \{B\}, \tag{9.5-17}$$

где $\{B\}$ обозначает последовательность примитивов, показанную на рис. 9.25(*б*) и (*в*) (операции с использованием последовательности примитивов рассматривались при обсуждении соотношения (9.5-7)). Эта последовательность состоит из двух различных структур, каждая из которых путем последовательных поворотов на 90° образует четыре примитива, что в сумме дает 8 примитивов. Знак «×» на рис. 9.25(*б*) обозначает «несущественность» в том смысле, что не имеет значения, какой пиксель оказывается в этой позиции — нулевой или единичный. В литературе по морфологическим методам сообщается о многочислен-

ных результатах, основанных на применении *одиночного* примитива, похожего на показанный на рис. 9.25(*б*), но с признаками несущественности во всем первом столбце. Это ошибочно: например, с помощью такого примитива точка, расположенная в 8-й строке и 4-м столбце на рис. 9.25(*а*), была бы опознана как концевая и удалена, что привело бы к разрыву штриха.

Трехкратное применение операции (9.5-17) к множеству A приводит к показанному на рис. 9.25(*г*) множеству X_1. Дальнейшие шаги состоят в «восстановлении» символа до его первоначального вида, но без удаленных паразитных ветвей. Для этого сначала потребуется построить множество X_2, состоящее из всех концевых точек X_1 (рис. 9.25(*д*)):

$$X_2 = \bigcup_{k=1}^{8} (X_1 \circledast B^k), \tag{9.5-18}$$

где B^k суть те же самые детекторы концевых точек, изображенные на рис. 9.25(*б*) и (*в*). На следующем шаге три раза выполняется дилатация этих концевых точек по заполненному единицами квадратному примитиву H размерами 3×3 при одновременном использовании исходного множества A в качестве ограничителя:

$$X_3 = (X_2 \oplus H) \cap A. \tag{9.5-19}$$

Соотношение (9.5-19) является основой для морфологической реконструкции через дилатацию, как объясняется в следующем разделе.

Как и в случаях заполнения областей и выделения связных компонент, условная дилатация такого вида предотвращает появление единичных пикселей за пределами области интереса, что подтверждает результат, показанный на рис. 9.25(*е*). Наконец, объединение множеств X_3 и X_1 приводит к желаемому результату

$$X_4 = X_1 \cup X_3, \tag{9.5-20}$$

приведенному на рис. 9.25(*ж*).

В более сложных случаях применение операции (9.5-19) иногда захватывает «кончики» некоторых паразитных ветвей, что случается, если они находятся вблизи остова. Хотя соотношением (9.5-17) они могли быть устранены, во время дилатации они захватываются снова, поскольку их правомерно считать точками множества A. Если только это не приводит к восстановлению паразитной составляющей целиком (что является редким случаем, если длина таких составляющих мала по сравнению с действительными штрихами), обнаружение и устранение подобных артефактов не представляет трудностей, поскольку они образуют несвязные области с исходным множеством.

В этом месте возникает естественная мысль, что должны существовать более простые пути решения данной задачи. Например, можно отслеживать все удаленные точки и присоединять обратно соответствующие точки ко всем концевым точкам, оставшимся после применения шага (9.5-17). Такой вариант вполне действенен, но достоинством приведенного выше построения является возможность решения всей задачи целиком с помощью простых

морфологических конструкций. В практических ситуациях, когда доступен набор таких инструментальных средств, это преимущество состоит в отсутствии необходимости написания новых алгоритмов. Мы просто комбинируем нужные морфологические функции, строя из них последовательность операций обработки.

9.5.9. Морфологическая реконструкция

В обсуждавшихся нами до сих пор концепциях морфологии участвовали изображение и структурообразующий элемент (примитив). В этом разделе будет рассмотрена *морфологическая реконструкция* — мощное морфологическое преобразование, в котором участвуют два изображения и примитив. Одно изображение — *маркер* — содержит начальные точки для преобразования. Другое изображение — *маска* — ограничивает область преобразования. Примитив служит для указания связности.[9]

Геодезические дилатация и эрозия

Центральными понятиями морфологической реконструкции являются геодезическая дилатация и геодезическая эрозия. Пусть F — изображение-маркер, а G — изображение-маска. В нашем обсуждении предполагается, что оба они двоичные изображения, причем $F \subseteq G$. *Геодезическая дилатация* первого порядка от изображения-маркера по изображению-маске обозначается $D_G^{(1)}(F)$ и определяется формулой

$$D_G^{(1)}(F) = (F \oplus B) \cap G \,, \tag{9.5-21}$$

где $\cap$ означает операцию пересечения множеств, которая здесь может интерпретироваться как операция логического умножения AND, так как множества двоичные. Геодезическая дилатация порядка n от F по G определяется рекуррентным соотношением

$$D_G^{(n)}(F) = D_G^{(1)}\left[D_G^{(n-1)}(F)\right], \tag{9.5-22}$$

где $D_G^{(0)}(F) = F$. В этом рекуррентном соотношении на каждом шаге выполняется операция пересечения множеств в (9.5-21).[10] Заметим, что оператор пересечения гарантирует, что рост (дилатация) маркера F будет ограничен маской G. На рис. 9.26 представлен простой пример геодезической дилатации порядка 1; шаги на рисунке прямо демонстрируют применение формулы (9.5-21).

Аналогично *геодезическая эрозия* первого порядка от маркера F по маске G определяется по формуле

---

[9] В большинстве литературы по морфологической реконструкции примитив неявно предполагается изотропным и обычно называется *элементарным изотропным примитивом*. В контексте данной главы примером такого примитива является массив 3×3 со всеми единичными элементами и центром в центральном элементе.

[10] Хотя методы морфологической реконструкции интуитивно более понятны, если их излагать с помощью рекуррентных формулировок (как мы это делаем здесь), их практическая реализация обычно опирается на более эффективные в вычислительном отношении алгоритмы (см., например, [Vincent, 1993] и [Soille, 2003]). Все примеры с изображениями в этом разделе были построены с применением таких алгоритмов.

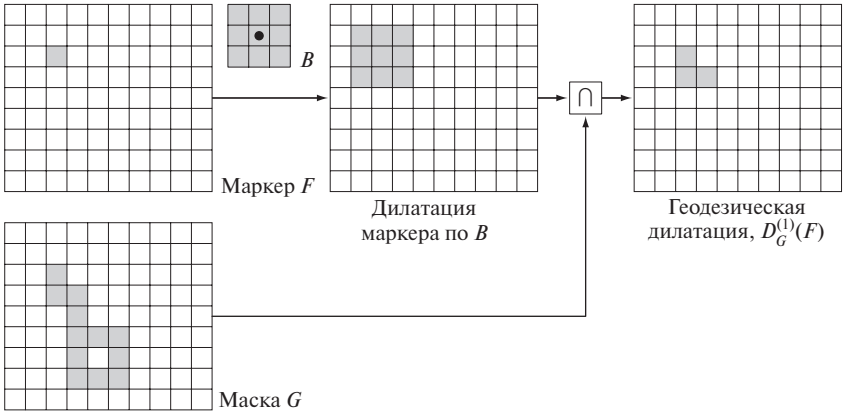


Рис. 9.26. Иллюстрация геодезической дилатации

$$E_G^{(1)}(F) = (F \ominus B) \cup G\,, \tag{9.5-23}$$

где $\cup$ означает операцию объединения множеств (или операцию логического сложения OR). Геодезическая эрозия порядка n от F по G определяется рекуррентным соотношением

$$E_G^{(n)}(F) = E_G^{(1)}\left[E_G^{(n-1)}(F)\right], \tag{9.5-24}$$

где $E_G^{(0)}(F) = F$. Операция объединения множеств в (9.5-23) выполняется на каждом шаге и гарантирует, что в результате геодезической эрозии изображение не станет меньше маски. Как можно ожидать по виду соотношений (9.5-21) и (9.5-23), геодезические дилатация и эрозия являются *двойственными* операциями относительно дополнения множеств (см. задачу 9.29). На рис. 9.27 представлен простой пример геодезической эрозии размера 1; шаги на рисунке прямо демонстрируют применение формулы (9.5-23).

Геодезические дилатация и эрозия конечных изображений всегда сходятся за конечное число шагов итерации, потому что расширение или сужение изображения-маркера ограничено маской.

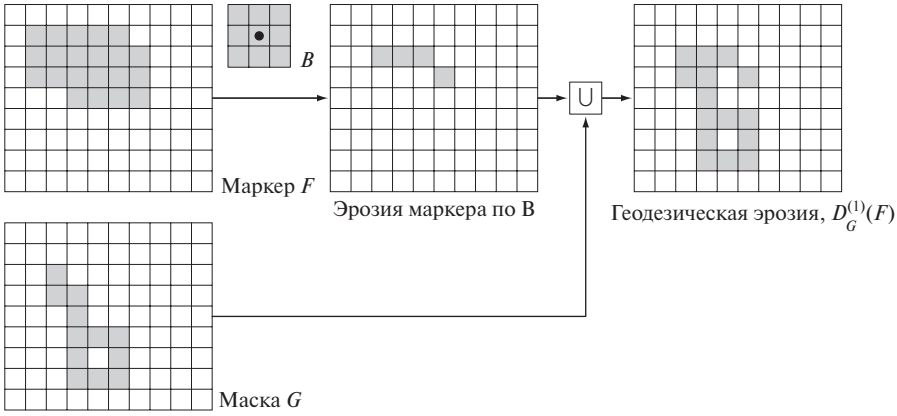


Рис. 9.27. Иллюстрация геодезической эрозии

Морфологическая реконструкция через дилатацию и через эрозию

Опираясь на определенные выше понятия, *морфологическая реконструкция через дилатацию* изображения-маркера F по изображению-маске G обозначается $R_G^D(F)$ и определяется как результат геодезической дилатации F по G, которая повторяется до получения стабильного результата, т. е.

$$R_G^D(F) = D_G^{(k)}(F) \tag{9.5-25}$$

при таком k, что $D_G^{(k)}(F) = D_G^{(k+1)}(F)$.

Рис. 9.28 демонстрирует морфологическую реконструкцию через дилатацию. Рис. 9.28(*а*) является продолжением процесса, начатого на рис. 9.26, т. е. следующий шаг реконструкции после получения $D_G^{(1)}(F)$ состоит в дилатации этого результата и затем операции AND с маской G, что приводит к изображению $D_G^{(2)}(F)$ на рис. 9.28(*б*). Дилатация $D_G^{(2)}(F)$ с маскированием по G дает $D_G^{(3)}(F)$ и т. д. вплоть до получения стабильного результата. Если проделать еще одну итерацию в данном примере, то окажется, что $D_G^{(5)}(F) = D_G^{(6)}(F)$, так что в соответствии с (9.5-25) результат морфологической реконструкции через дилатацию есть $D_G^D(F) = D_G^{(5)}(F)$. Заметим, что в данном случае реконструированное изображение совпадает с маской, поскольку F содержало единственный пиксель со значением 1 (это аналогично свертке изображения с дискретным единичным импульсом, которая просто дает копию элемента изображения в точке положения импульса, как говорилось в разделе 3.4.2).

Аналогичным образом *морфологическая реконструкция через эрозию* изображения-маркера F по изображению-маске G обозначается $R_G^E(F)$ и определяется как результат геодезической эрозии F по G, которая повторяется до получения стабильного результата, т. е.

$$R_G^E(F) = E_G^{(k)}(F) \tag{9.5-26}$$

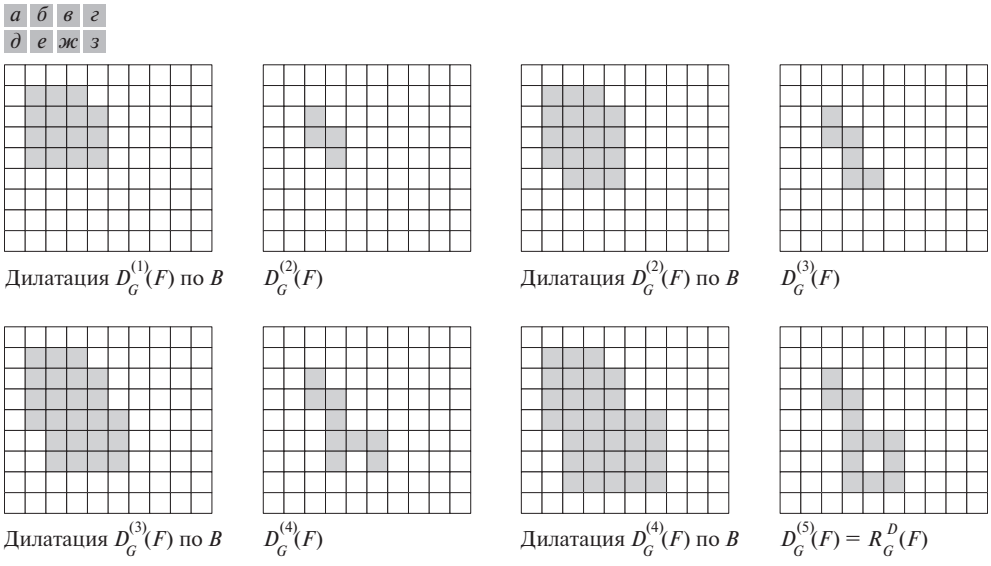


Рис. 9.28. Иллюстрация морфологической реконструкции через дилатацию. F, G, B и $D_G^{(1)}(F)$ те же, что и на рис. 9.26

при таком k, что $E_G^{(k)}(F) = E_G^{(k+1)}(F)$. Читателю рекомендуется в качестве упражнения построить рисунок, аналогичный рис. 9.28, для случая морфологической реконструкции через эрозию.

Реконструкции через дилатацию и через эрозию являются двойственными операциями относительно дополнения множеств (см. задачу 9.30).[11]

Простые применения

Морфологическая реконструкция имеет широкий спектр возможных практических применений, которые определяются выбором изображений в качестве маркера и маски, используемыми примитивами и комбинацией элементарных операций, введенных выше. Следующие примеры иллюстрируют полезность этих концепций.

***Размыкание с реконструкцией*:** в морфологической операции размыкания эрозия удаляет мелкие объекты, а последующая дилатация есть попытка восстановления формы оставшихся объектов. Однако точность такого восстановления сильно зависит от того, насколько объекты схожи по своей форме с используемым примитивом. Операция *размыкания с реконструкцией* восстанавливает *в точности* первоначальную форму объектов, которые сохранились после эрозии. Размыкание с реконструкцией порядка n от изображения F есть по определению реконструкция через дилатацию, примененная к результату эрозии порядка n от изображения F по самому F, т. е.

$$O_R^{(n)}(F) = R_F^D\left[(F \ominus nB)\right], \tag{9.5-27}$$

где $(F \ominus nB)$ означает n-кратное применение эрозии F по B, как объяснялось в разделе 9.5.7. Отметим, что в данном применении F используется в качестве маски. Аналогичное выражение можно записать для замыкания с реконструкцией (см. табл. 9.1).

На рис. 9.29 представлен пример размыкания с реконструкцией. В данном примере задача состоит в том, чтобы выделить на исходном изображении рис. 9.29(*а*) символы, в которых есть длинные вертикальные штрихи. Для размыкания с реконструкцией требуется выполнить по меньшей мере одну эрозию, поэтому вначале сделаем такой шаг. На рис. 9.29(*б*) показан результат эрозии исходного изображения с помощью примитива шириной в один пиксель и высотой, соответствующей средней высоте высоких символов (51 пиксель). Для сравнения вычислено обычное размыкание по тому же примитиву; результат представлен на рис. 9.29(*в*). Наконец, на рис. 9.29(*г*) показан результат размыкания с реконструкцией первого порядка, полученный для исходного изображения F, т. е. $O_R^{(1)}(F)$ согласно (9.5-27). Как видно из рисунка, символы с длинными

[11] После изучения данного раздела у читателя может возникнуть недоумение: в чем, собственно, разница между операциями морфологической реконструкции (скажем, через дилатацию) и просто геодезической дилатации? Содержательное отличие в том, что операция геодезической дилатации произвольного порядка n является детерминированной и состоит в точности из n элементарных шагов. Напротив, число шагов морфологической реконструкции заранее неизвестно и определяется путем сравнения промежуточных результатов (стабилизация результата означает конец операции); такую «операцию» более естественно считать алгоритмом. (То же самое относится к операциям морфологической реконструкции через эрозию и геодезической эрозии.) — *Прим. перев.*

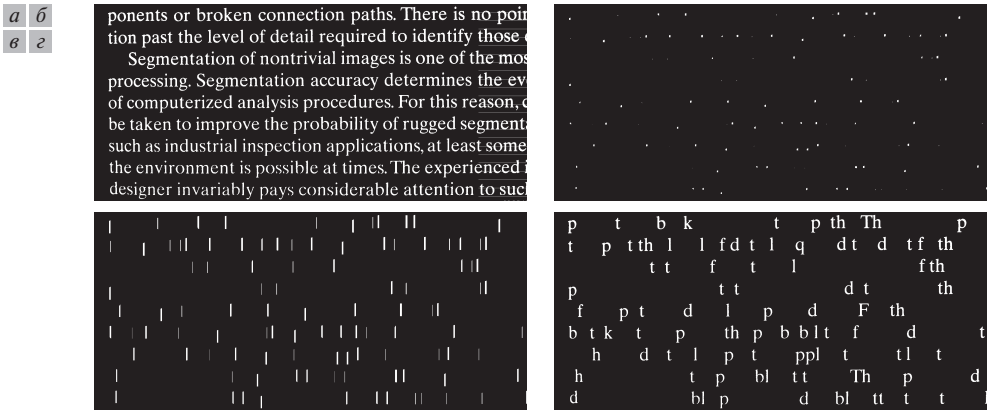


Рис. 9.29. (*а*) Изображение текста с размерами 918×2018 пикселей. Высокие символы имеют среднюю высоту приблизительно 50 пикселей. (*б*) Результат эрозии (*а*) по примитиву размерами 51×1. (*в*) Результат размыкания (*а*) по тому же примитиву; приводится для сравнения. (*г*) Результат размыкания с реконструкцией

вертикальными штрихами восстановлены точно (в отличие от обычного размыкания), а все прочие символы удалены.

Заполнение дырок: в разделе 9.5.2 был изложен алгоритм заполнения дырок, для которого требуется знать начальную точку внутри каждой дырки на объектах изображения. Ниже излагается полностью автоматическая процедура для той же цели, основанная на морфологической реконструкции. Пусть $I(x,y)$ — исходное двоичное изображение; построим для него изображение-маркер F, которое имеет нулевые значения во всех точках, кроме пикселей на краях изображения, для которых принимается значение $1-I$, т. е.

$$F(x,y)=\begin{cases}1-I(x,y) & \text{если } (x,y) \text{ на краю } I,\\ 0 & \text{в остальных случаях.}\end{cases} \tag{9.5-28}$$

Тогда

$$H=\left[R_{I^c}^{D}(F)\right]^c \tag{9.5-29}$$

есть двоичное изображение, совпадающее с I, но со всеми заполненными дырками.

Рассмотрим отдельные составляющие правой части (9.5-29), чтобы лучше понять, как это выражение приводит к заполнению всех дырок на изображении. На рис. 9.30(*а*) приведено простое изображение I, на котором имеется одна дырка, а на рис. 9.30(*б*) — его дополнение. Поскольку при вычислении дополнения I все пиксели переднего плана становятся фоновыми, а фоновые — пикселями переднего плана (меняя значения с 1 на 0 и наоборот), эта операция приводит к построению «стенки» из нулей вокруг каждой дырки. Изображение I^c используется как маска в операторе AND, и смысл этого в том, чтобы защитить все пиксели переднего плана (в том числе пиксели стенок вокруг дырок) от изменений в ходе алгоритма. На рис. 9.30(*в*) изображен массив F, построенный согласно (9.5-28), а на рис. 9.30(*г*) — результат его дилатации по примитиву 3×3

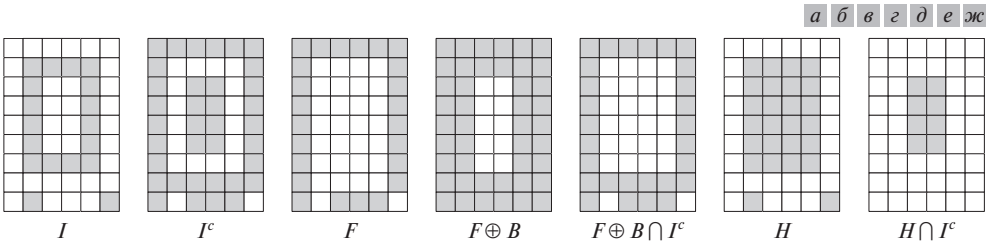


Рис. 9.30. Иллюстрация алгоритма заполнения дырок на примере простого изображения

со всеми единичными значениями. Отметим, что все периферийные элементы маркера F имеют значение 1 (за исключением тех точек, где расположены единичные элементы исходного изображения), поэтому дилатация маркера будет развиваться от краев изображения внутрь. На рис. 9.30(*д*) представлен результат геодезической дилатации F с использованием I^c в качестве маски. Как видим, все точки, соответствующие пикселям переднего плана на изображении I, принимают значение 0, и то же справедливо для пикселей в дырках. Повторение этой операции приводит к тому же результату (т.е. достигнута сходимость), и его дополнение согласно (9.5-29) приводит к результату на рис. 9.30(*е*), где дырка заполнена, а значения прочих пикселей не изменились. Если выполнить операцию $H \cap I^c$, то получится изображение, состоящее только из внутреннего заполнения всех дырок на изображении I (рис. 9.30(*ж*)).

На рис. 9.31 представлен более практический пример. Рис. 9.31(*б*) демонстрирует дополнение исходного изображения текста на рис. 9.31(*а*), а на рис. 9.31(*в*) показано изображение-маркер F, построенное с помощью соотношения (9.5-28). На краях этого изображения имеется сплошная рамка из единичных элементов, за исключением тех мест, где на краях исходного изображения были единичные пиксели.[12] Окончательно на рис. 9.31(*г*) представлено изображение, на котором все дырки заполнены.

Очистка краев изображения: одна из фундаментальных задач автоматической обработки изображений состоит в выделении объектов на изображении с целью последующего анализа их формы. При этом полезно иметь алгоритм для удаления объектов, которые касаются края изображения (т.е. являются связными с рамкой из элементов на краях), потому что (1) такой алгоритм позволяет в дальнейшем обрабатывать изображение, содержащее только объекты целиком, и (2) его можно использовать для сигнализации наличия в поле зрения объектов, которые видны лишь частично. В качестве заключительной иллюстрации идей данного раздела изложим алгоритм очистки краев изображения, основанный на морфологической реконструкции. Для такого применения в качестве маски используется исходное изображение, а изображение-маркер формируется по правилу

$$F(x,y)=\begin{cases} I(x,y) & \text{если } (x,y) \text{ на краю } I, \\ 0 & \text{в остальных случаях.} \end{cases} \qquad (9.5\text{-}30)$$

[12] Читателю придется поверить авторам на слово, потому что из-за особенностей полиграфического процесса этого не видно даже при большом увеличении. — *Прим. перев.*

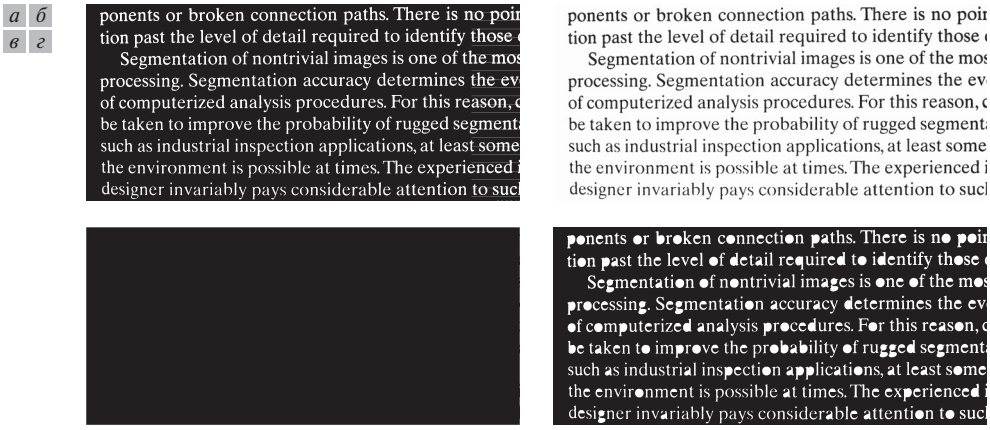


Рис. 9.31. (*а*) Изображение текста с размерами 918×2018 пикселей. (*б*) Дополнение (*а*), используемое как маска. (*в*) Изображение-маркер. (*г*) Результат заполнения дырок с помощью соотношения (9.5-29)

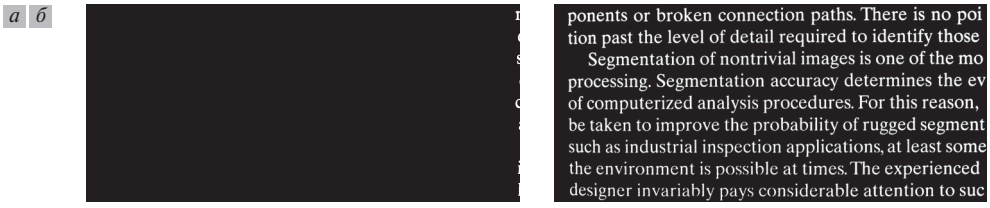


Рис. 9.32. Очистка краев изображения. (*а*) Результат реконструкции через дилатацию, примененную к изображению-маркеру. (*б*) Изображение без объектов, примыкающих к краям. В качестве исходного используется изображение на рис. 9.29(*а*)

Алгоритм очистки краев вначале вычисляет морфологическую реконструкцию $R_I^D(F)$ (которая просто выделяет объекты, касающиеся краев изображения), а затем вычисляет разность

$$X = I \setminus R_I^D(F) \tag{9.5-31}$$

для получения результирующего изображения X, не содержащего объектов, которые касаются краев.

Снова рассмотрим в качестве примера то же самое изображение текста. На рис. 9.32(*а*) представлен результат реконструкции $R_I^D(F)$ с помощью примитива размерами 3×3 со всеми единичными элементами (обратите внимание на объекты, касающиеся правого края изображения), а рис. 9.32(*б*) демонстрирует результирующее изображение X, полученное с помощью соотношения (9.5-31). Если бы рассматривалась задача автоматического распознавания знаков, то наличие изображения, в котором символы не касаются краев, было бы весьма полезно, так как помогает избежать решения по меньшей мере непростой задачи распознавания знаков, которые видны не полностью.

9.5.10. Сводная таблица морфологических операций

В табл. 9.1 приведены итоговые результаты обсуждения морфологических операций, представленных в предшествующих разделах, а на рис. 9.33 показаны основные виды примитивов, применяющихся в различных процессах морфологической обработки, которые были рассмотрены ранее. Числа, записанные римскими цифрами в скобках в третьей колонке таблицы, указывают номера примитивов на рис. 9.33.

Таблица 9.1. Сводная таблица морфологических операций и их свойств

Операция	Выражение	**Примечание** (римские цифры в скобках означают номера примитивов на рис. 9.33)
Параллельный перенос	$(B)_z = \{w \mid w = b + z,\ b \in B\}$	Сдвиг центра (начала координат) множества B в точку z
Центральное отражение	$\hat{B} = \{w \mid w = -b,\ b \in B\}$	Симметричное отражение всех элементов B относительно начала координат
Дополнение	$A^c = \{w \mid w \notin A\}$	Множество точек, не входящих в A
Разность	$A \setminus B = \{w \mid w \in A, w \notin B\} = A \cap B^c$	Множество точек, принадлежащих A, но не принадлежащих B
Дилатация	$A \oplus B = \{z \mid (\hat{B})_z \cap A \neq \varnothing\}$	«Расширение» границы множества A (I)
Эрозия	$A \ominus B = \{z \mid (B)_z \subseteq A\}$	«Сужение» границы множества A (I)
Размыкание	$A \circ B = (A \ominus B) \oplus B$	Сглаживает контуры, разрывает узкие перешейки, убирает небольшие островки и острые выступы (I)
Замыкание	$A \bullet B = (A \oplus B) \ominus B$	Сглаживает контуры, заполняет узкие разрывы, углубления и небольшие дырки (I)
Попадание/пропуск	$A \circledast B = (A \ominus B_1) \cap (A^c \ominus B_2) =$ $= (A \ominus B_1) \setminus (A \oplus \hat{B}_2)$	Множество координат точек, в которых одновременно для B_1 есть совпадение в A, а для B_2 — в A^c
Выделение границы	$\beta(A) = A \setminus (A \ominus B)$	Множество граничных точек множества A (I)
Заполнение дырок	$X_k = (X_{k-1} \oplus B) \cap A^c;\quad k = 1, 2, 3, \ldots$	Заполняет дырки в множестве A; X_0 — массив нулей с единицей в каждой дырке (II)
Выделение связных компонент	$X_k = (X_{k-1} \oplus B) \cap A;\quad k = 1, 2, 3, \ldots$	Находит связные компоненты в множестве A; X_0 — массив нулей с единицей в каждой связной компоненте (I)

Таблица 9.1. (*Продолжение*)

Операция	Выражение	**Примечание** (римские цифры в скобках означают номера примитивов на рис. 9.33)
Выпуклая оболочка	$X_k^i = (X_{k-1}^i \circledast B^i) \cup A; \quad i = 1, 2, 3, 4;$ $k = 1, 2, 3, \ldots; \quad X_0^i = A$ и $D^i = X_{\text{сход}}^i$	Находит для множества A выпуклую оболочку $C(A)$. Индекс «сход» соответствует сходимости в том смысле, что $X_k^i = X_{k-1}^i$ (III)
Утончение	$A \otimes B = A \setminus (A \circledast B)$ $A \otimes B = A \cap (A \circledast B)^c$ $A \otimes \{B\} = \left(\left(\ldots\left(\left(A \otimes B^1\right) \otimes B^2\right)\ldots\right) \otimes B^n\right)$ $\{B\} = \left\{B^1, B^2, B^3, \ldots, B^n\right\}$	Делает множество A «тоньше». Первые два выражения являются базовым определением утончения. Последние выражения отвечают утончению по последовательности примитивов. Этот метод обычно применяется на практике (IV)
Утолщение	$A \odot B = A \cup (A \circledast B)$ $A \odot \{B\} = \left(\left(\ldots\left(\left(A \odot B^1\right) \odot B^2\right)\ldots\right) \odot B^n\right)$	Делает множество A «толще» (см. выше замечание о последовательности примитивов). Используются примитивы (IV), в которых нули заменяются единицами, а единицы — нулями
Построение остова	$S(A) = \bigcup_{k=0}^{K} S_k(A)$ $S_k(A) = (A \ominus kB) \setminus [(A \ominus kB) \circ B]$ Восстановление множества A: $A = \bigcup_{k=0}^{K} (S_k(A) \oplus kB)$	Находит остов $S(A)$ множества A. Последнее выражение показывает, что множество A может быть восстановлено по подмножествам остова $S_k(A)$. В этих выражениях значение K — номер шага итерации, после которого эрозия множества A приводит к пустому множеству. Запись $(A \ominus kB)$ обозначает k применений подряд к множеству A операции эрозии по примитиву B (I)
Усечение	$X_1 = A \otimes \{B\}$ $X_2 = \bigcup_{k=1}^{8} (X_1 \circledast B^k)$ $X_3 = (X_2 \oplus H) \cap A$ $X_4 = X_1 \cup X_3$	X_4 есть результат усечения множества A. Необходимо указать, сколько раз применяется первое выражение для получения X_1. В двух первых выражениях используются примитивы V; H в третьем выражении обозначает примитив типа I

Таблица 9.1. (*Окончание*)

Операция	Выражение	**Примечание** (римские цифры в скобках означают номера примитивов на рис. 9.33)
Геодезическая дилатация порядка 1	$D_G^{(1)}(F) = (F \oplus B) \cap G$	Изображения F и G называются соответственно *маркером* и *маской*
Геодезическая дилатация порядка n	$D_G^{(n)}(F) = D_G^{(1)}\left[D_G^{(n-1)}(F)\right]$; $D_G^{(0)}(F) = F$	
Геодезическая эрозия порядка 1	$E_G^{(1)}(F) = (F \ominus B) \cup G$	
Геодезическая эрозия порядка n	$E_G^{(n)}(F) = E_G^{(1)}\left[E_G^{(n-1)}(F)\right]$; $E_G^{(0)}(F) = F$	
Морфологическая реконструкция через дилатацию	$R_G^D(F) = D_G^{(k)}(F)$	k такое, что $D_G^{(k)}(F) = D_G^{(k+1)}(F)$
Морфологическая реконструкция через эрозию	$R_G^E(F) = E_G^{(k)}(F)$	k такое, что $E_G^{(k)}(F) = E_G^{(k+1)}(F)$
Размыкание с реконструкцией	$O_R^{(n)}(F) = R_F^D\left[(F \ominus nB)\right]$	$(F \ominus nB)$ означает n-кратное применение эрозии F по B
Замыкание с реконструкцией	$C_R^{(n)}(F) = R_F^E\left[(F \oplus nB)\right]$	$(F \oplus nB)$ означает n-кратное применение дилатации F по B
Заполнение дырок	$H = \left[R_{I^c}^D(F)\right]^c$	H совпадает с исходным изображением I, но все дырки на нем заполнены. Изображение-маркер F определено в (9.5-28)
Очистка краев	$X = I \setminus R_I^D(F)$	X совпадает с исходным изображением I, но все объекты, которые касаются краев, удалены. Изображение-маркер F определено в (9.5-30)

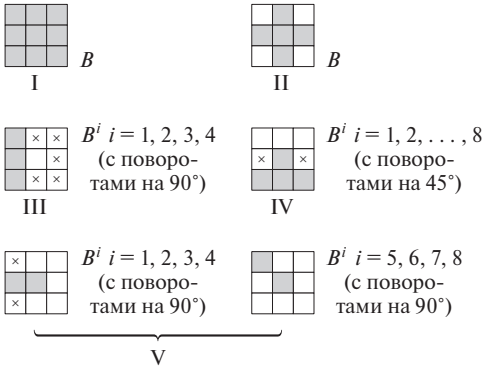


Рис. 9.33. Пять основных типов примитивов, применяемых для двоичных морфологических методов. Начало координат каждого примитива расположено в его центре, и знаки «×» указывают, что значение пикселя на этой позиции не принимается во внимание

9.6. Морфология полутоновых изображений

В этом разделе мы распространим базовые морфологические операции дилатации, эрозии, размыкания и замыкания на полутоновые изображения, а затем воспользуемся ими для построения нескольких основных полутоновых морфологических алгоритмов.

В последующем рассмотрении будут участвовать цифровые изображения, заданные функциями $f(x, y)$ и $b(x, y)$, где $f(x, y)$ — исходное изображение, а функция $b(x, y)$ задает примитив (структурообразующее множество). Предполагается, что эти функции являются дискретными в том смысле, как это определялось в разделе 2.4.2, т. е. пары координат (x, y) берутся из декартова произведения Z^2 (где Z — множество целых чисел), и функции f и b сопоставляют каждой паре координат значение яркости в соответствующей точке — действительное число из множества R. Если значения яркости также целочисленные, то вместо R используется Z.

Примитивы, применяемые при морфологической обработке полутоновых изображений, выполняют те же основные функции, что и в случае двоичных изображений — они используются как «зонды» для исследования данного изображения на предмет некоторых специфических свойств. Эти примитивы разбиваются на две категории: *плоские* (*однотонные*) и *неплоские* (*неоднотонные*) (см. примеры на рис. 9.34). На рис. 9.34(*а*) показан полусферический примитив как полутоновое изображение, а на рис. 9.34(*в*) — горизонтальный

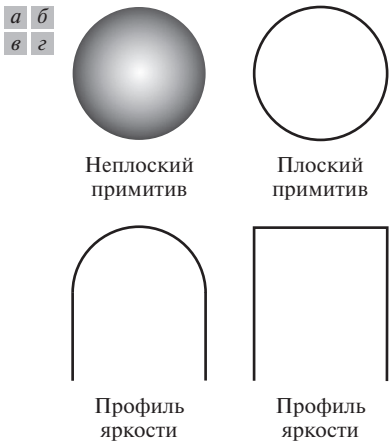


Рис. 9.34. Неплоский и плоский примитивы и соответствующие им профили яркости через центр. Все примеры данного раздела основаны на использовании плоских (однотонных) примитивов

профиль яркости через центр примитива. На рис. 9.34(*б*) представлен плоский примитив в форме круга, а на рис. 9.34(*г*) — соответствующий профиль яркости (форма профиля объясняет происхождение названия «плоский»). Примитивы на рис. 9.34 для наглядности показаны непрерывными, хотя при компьютерной реализации используются дискретные приближения (см. в качестве примера «круглый» примитив справа на рис. 9.2). Ввиду ряда трудностей, рассматриваемых далее в этом разделе, полутоновые примитивы редко применяются на практике. В заключение заметим, что, как и в двоичном случае, должно быть ясно определено начало координат (центр) каждого используемого примитива. Если не оговорено иное, во всех примерах данного раздела предполагается использование центрально-симметричных плоских примитивов с единичным значением яркости и центром в центре симметрии. *Центральное отражение* примитива в полутоновой морфологии определяется так же, как в разделе 9.1, и ниже обозначается $\hat{b}(x,y) = b(-x,-y)$.

9.6.1. Эрозия и дилатация

Полутоновая *эрозия* изображения f по *плоскому* примитиву b в произвольной точке (x,y) определяется как *минимальное* значение f в области, совпадающей с b, когда центр примитива находится в точке (x,y). Эрозия обозначается $f \ominus b$ и задается формулой

$$(f \ominus b)(x,y) = \min_{(s,t)\in b}\{f(x+s, y+t)\}, \qquad (9.6\text{-}1)$$

где по аналогии с процедурой корреляции из раздела 3.4.2 x и y пробегают все значения, так чтобы центр примитива b пробегал все элементы f. Таким образом, для нахождения эрозии f по b необходимо последовательно помещать центр b в каждый пиксель изображения и находить в этом положении минимальное значение f в точках окрестности, накрываемой примитивом. Например, если b — квадратный примитив размерами 3×3, требуется найти наименьшее из 9 значений f в области 3×3 пикселя с центром в точке, где расположен центр примитива.

Аналогично полутоновая дилатация изображения f по плоскому примитиву b в произвольной точке (x,y) определяется как *максимальное* значение f в окрестности, совпадающей с $\hat{b}$, когда центр примитива находится в точке (x,y). Дилатация обозначается $f \oplus b$ и задается формулой

$$(f \oplus b)(x,y) = \max_{(s,t)\in b}\{f(x-s, y-t)\}, \qquad (9.6\text{-}2)$$

где использован отмеченный выше факт, что $\hat{b} = b(-x,-y)$. Объяснение этого равенства аналогично объяснению полутоновой эрозии с той лишь разницей, что здесь используется максимум вместо минимума и примитив подвергнут центральному отражению, на что указывает вычитание s и t в аргументах функции. Такая процедура аналогична свертке из раздела 3.4.2.

Пример 9.9. Иллюстрация действия операций дилатации и эрозии на полутоновое изображение.

■ Поскольку при полутоновой эрозии с плоским примитивом вычисляется минимальное значение яркости f в каждой окрестности, совпадающей с положе-

а б в

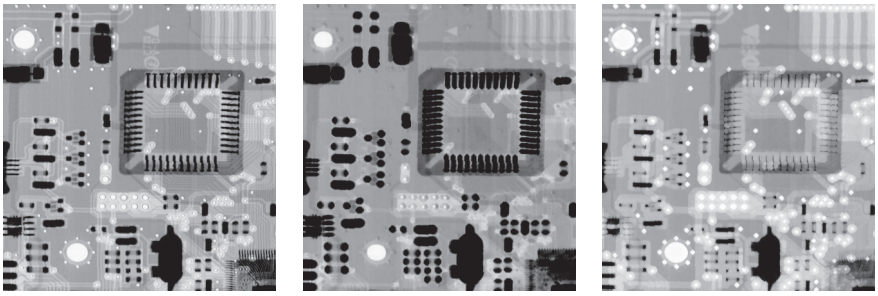


Рис. 9.35. (*а*) Исходное полутоновое рентгеновское изображение размерами 448×425 пикселей. (*б*) Результат эрозии по плоскому примитиву в форме круга радиусом 2 пикселя. (*в*) Результат дилатации по тому же примитиву. (Исходное изображение предоставила компания Lixi, Inc.)

нием примитива b, в общем случае можно ожидать, что полученное изображение будет темнее исходного, а размеры сопоставимых с размерами примитива ярких деталей уменьшатся, а темных увеличатся. На рис. 9.35(*а*) приведено исходное полутоновое изображение, а рис. 9.35(*б*) демонстрирует результат эрозии этого изображения по плоскому примитиву круглой формы единичной высоты и радиусом в 2 пикселя. Указанные выше эффекты ясно видны на рис. 9.35(*б*), например уменьшилась яркость небольших ярких точек, так что они едва видны, а темные детали стали толще. Общий фон изображения стал лишь ненамного темнее. Аналогично рис. 9.35(*в*) демонстрирует результат дилатации исходного изображения по тому же примитиву. Заметны эффекты противоположного характера: размеры ярких деталей увеличились, а яркость темных деталей уменьшилась. В частности, обратим внимание, что темные контактные площадки слева в середине и справа внизу на рис. 9.35(*а*) почти не видны на рис. 9.35(*в*). Темные точки после дилатации также уменьшились, но, в отличие от ярких точек в случае эрозии, они все еще хорошо заметны на полученном изображении. Причина в том, что в исходном изображении размеры темных точек больше размеров примитива, а размеры ярких — меньше. В заключение отметим, что общий фон полученного изображения лишь ненамного ярче, чем на рис. 9.35(*а*). ■

У неплоских примитивов значения яркости меняются по области их определения. Эрозия изображения f по неплоскому примитиву b_N определяется формулой

$$(f \ominus b_N)(x,y) = \min_{(s,t)\in b_N} \{f(x+s,y+t) - b_N(s,t)\}. \qquad (9.6\text{-}3)$$

Здесь мы вычитаем значения яркости примитива из яркости изображения f, чтобы определить значение эрозии в каждой точке. Это означает, что, в отличие от (9.6-1), результаты эрозии по неплоскому примитиву в общем случае не ограничены интервалом значений f, что затрудняет их интерпретацию. По этой причине, а также из-за усложнения вычислений и потенциальных трудностей выбора осмысленных значений элементов b_N неплоские примитивы редко применяются на практике.

Аналогичным образом дилатация по неплоскому примитиву определяется формулой

$$(f \oplus b_N)(x,y) = \max_{(s,t)\in b_N} \{f(x-s, y-t) + b_N(s,t)\}; \quad (9.6\text{-}4)$$

к ней относятся те же комментарии, что и для эрозии по неплоскому примитиву. Если значения всех элементов b_N равны некоторой константе (т. е. примитив плоский), то равенства (9.6-3), (9.6-4) сводятся соответственно к (9.6-1), (9.6-2) с точностью до этой константы.

Как и в двоичном случае, полутоновые операции эрозии и дилатации являются двойственными друг другу по отношению к дополнению и центральному отражению функций, т. е. равенство

$$(f \ominus b)^c(x,y) = (f^c \oplus \hat{b})(x,y),$$

где $f^c = -f(x,y)$ и $\hat{b} = b(-x,-y)$ справедливо и в случае неплоских примитивов[13]. В дальнейшем рассмотрении, за исключением тех случаев, когда это необходимо для ясности, мы будем пользоваться упрощенной системой обозначений, опуская аргументы всех функций, так что последнее равенство примет вид

$$(f \ominus b)^c = (f^c \oplus \hat{b}). \quad (9.6\text{-}5)$$

Аналогично

$$(f \oplus b)^c = (f^c \ominus \hat{b}). \quad (9.6\text{-}6)$$

Сами по себе операции эрозии и дилатации не особенно полезны для обработки полутоновых изображений. Как и в случае их двоичных аналогов, эти операции проявляют свою силу при совместном использовании в составе алгоритмов более высокого уровня, как показывает материал последующих разделов.

9.6.2. Размыкание и замыкание

Выражения для операций размыкания и замыкания в случае полутоновых изображений имеют такую же форму, как и в двоичном случае. Размыкание изображения f по примитиву b обозначается $f \circ b$ и определяется следующим образом:

$$f \circ b = (f \ominus b) \oplus b. \quad (9.6\text{-}7)$$

> Хотя на протяжении оставшейся части данной главы в примерах используются плоские примитивы, обсуждаемые понятия также применимы и для неплоских примитивов.

Как и в двоичном случае, размыкание состоит в эрозии f по b, после чего к результату применяется дилатация опять-таки по b. Аналогично замыкание f по b обозначается $f \bullet b$ и определяется следующим образом:

$$f \bullet b = (f \oplus b) \ominus b. \quad (9.6\text{-}8)$$

---

[13] Здесь и ниже авторы формально допускают существование значений $-f$, т. е. отрицательных значений яркостей. — *Прим. перев.*

Полутоновые операции размыкания и замыкания являются двойственными друг другу по отношению к дополнению и центральному отражению функций, т. е.

$$(f \bullet b)^c = f^c \circ \hat{b} \tag{9.6-9}$$

и

$$(f \circ b)^c = f^c \bullet \hat{b}\,. \tag{9.6-10}$$

Поскольку $f^c = -f\,(x,y)$, равенство (9.6-9) можно также записать в виде $-(f \bullet b) = \left(-f \circ \hat{b}\right)$; аналогичным же образом можно выразить и (9.6-10).

Для операций размыкания и замыкания изображений имеется простая геометрическая интерпретация. Рассмотрим функцию изображения $f(x, y)$ как трехмерную поверхность, высота которой над плоскостью xy в любой точке (x, y) равна значению яркости изображения f при этих координатах, как показано на рис. 2.18(*а*). Тогда механизм размыкания функции f по b можно геометрически интерпретировать как процесс обхода примитивом всей нижней стороны поверхности. При этом огибающая высших точек примитива и будет поверхностью, соответствующей размыканию исходного изображения.

Рис. 9.36 иллюстрирует описанный принцип в одномерном случае. Пусть кривая на рис. 9.36(*а*) представляет профиль яркости вдоль одной из строк полутонового изображения. На рис. 9.36(*б*) показаны различные положения плоского примитива во время обхода этой строки снизу, а рис. 9.30(*в*) демонстрирует окончательный результат размыкания вдоль этой строки. Все пики, узкие по сравнению с размером примитива, срезаны по амплитуде и сглажены, и сте-

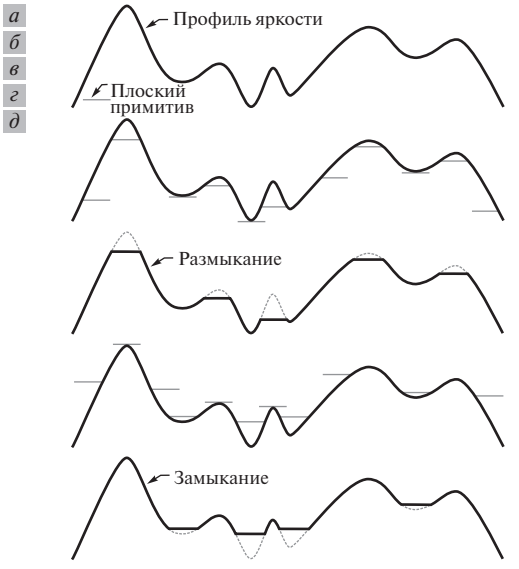


Рис. 9.36. Размыкание и замыкание в одномерном случае[14]. (*а*) Исходный одномерный сигнал. (*б*) Положения плоского примитива при обходе кривой снизу. (*в*) Результат размыкания. (*г*) Положения плоского примитива при обходе кривой сверху. (*д*) Результат замыкания

пень снижения вершины определяется тем, насколько глубоко примитив в состоянии войти в пик. На практике операция размыкания обычно используется для устранения небольших (по сравнению с размерами примитива) светлых деталей при сохранении практически неизменными общей яркости и крупных ярких деталей.

> Иногда операции размыкания и замыкания иллюстрируют с помощью круглого примитива, который обкатывает нижнюю и верхнюю стороны кривой. В трехмерном случае круг превращается в сферу и описываемые процедуры называют алгоритмами «*обкатки шариком*».

Рис. 9.36(*г*) иллюстрирует операцию замыкания. Здесь примитив обходит кривую сверху, и результатом замыкания (рис. 9.36(*д*)) является геометрическое место самых нижних точек примитива при всех возможных координатах его центра.

Полутоновая операция размыкания удовлетворяет следующим свойствам:

(а) $(f \circ b) \leftarrow\!\!\lrcorner\, f$.

(б) Если $f_1 \leftarrow\!\!\lrcorner\, f_2$, то $(f_1 \circ b) \leftarrow\!\!\lrcorner\, (f_2 \circ b)$.

(в) $(f \circ b) \circ b = f \circ b$.

Запись $e \leftarrow\!\!\lrcorner\, r$ обозначает, что область определения e является подмножеством области определения r, и при этом $e(x, y) \le r(x, y)$ для любой точки (x, y) в области определения e.

Аналогично полутоновая операция замыкания удовлетворяет следующим свойствам:

(а) $f \leftarrow\!\!\lrcorner\, (f \bullet b)$.

(б) Если $f_1 \leftarrow\!\!\lrcorner\, f_2$, то $(f_1 \bullet b) \leftarrow\!\!\lrcorner\, (f_2 \bullet b)$.

(в) $(f \bullet b) \bullet b = f \bullet b$.

Эти свойства столь же полезны, как и их аналоги для двоичного случая.

Пример 9.10. Иллюстрация полутоновых операций размыкания и замыкания.

■ Рис. 9.37 демонстрирует расширение представленного на рис. 9.36 одномерного принципа для двумерного случая. На рис. 9.37(*а*) показано то же изображение, что и в примере 9.9, а на рис. 9.37(*б*) приведен результат его размыкания по примитиву в форме круга радиусом 3 пикселя с единичными значениями во всех точках[15]. Как можно было ожидать, яркость всех ярких деталей уменьшилась, причем степень снижения зависит от размеров деталей по сравнению с диаметром примитива. Из сравнения этого результата с рис. 9.35(*б*) видно, что, в отличие от одной эрозии, размыкание практически не влияет на фон и темные детали изображения. Аналогично на рис. 9.37(*в*) представлен результат замыкания исходного изображения по круглому примитиву радиуса 5 (поскольку размеры маленьких круглых темных точек больше, чем у маленьких белых точек,

---

[14] Отметим, что результаты операций (9.6-7) и (9.6-7), которые, как это показано на рис. 9.36, соответствуют скольжению отрезка, будут отличаться от предлагаемой в заметке выше по поводу «обкатки шариком». В последнем случае усечение кривых будет не плоским, а выпуклым по форме шарика. — *Прим. перев.*

[15] Как и в случае эрозии/дилатации, собственное значение плоского примитива в операциях размыкания и замыкания не используется, поэтому может быть любым. — *Прим. перев.*

а б в

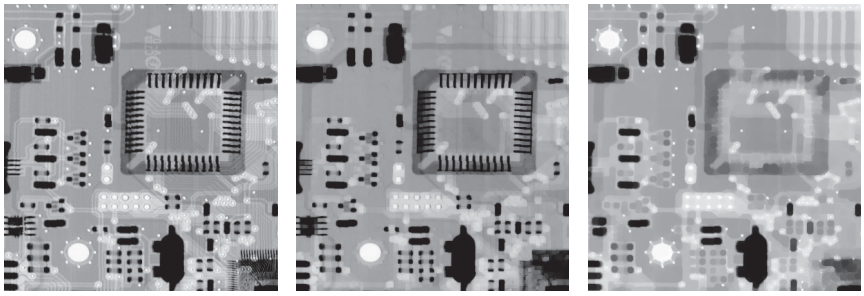


Рис. 9.37. (*а*) Исходное полутоновое рентгеновское изображение размерами 448×425 пикселей. (*б*) Результат размыкания по плоскому примитиву в форме круга радиусом 3 пикселя. (*в*) Результат замыкания по круглому примитиву радиусом 5 пикселей

то для достижения сопоставимого с размыканием эффекта понадобился примитив большего диаметра). В полученном изображении нет ощутимых изменений фона и ярких деталей, зато темные детали стали слабее, причем степень ослабления зависит от размеров детали по сравнению с диаметром примитива. ■

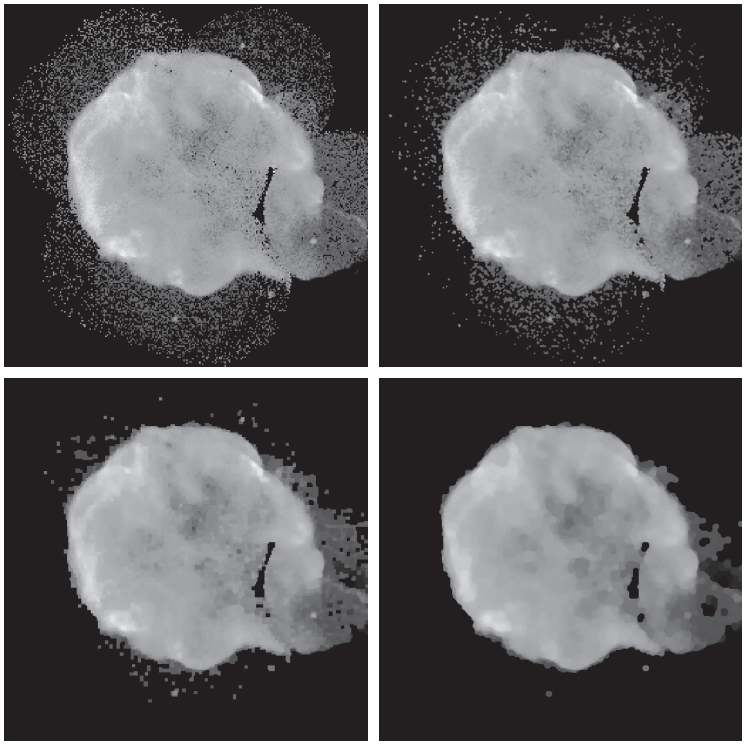
9.6.3. Некоторые основные алгоритмы полутоновой морфологии

Многие морфологические методы основываются на изложенных к этому моменту идеях полутоновой морфологии. Некоторые из этих алгоритмов рассматриваются ниже.

Морфологическое сглаживание

Поскольку операция размыкания удаляет яркие детали, размеры которых меньше заданного примитива, а операция замыкания удаляет темные детали, обе операции часто используются совместно в *морфологических фильтрах* для сглаживания изображения и ослабления шума. Рассмотрим на рис. 9.38(*а*) рентгеновское изображение сверхновой в Петле Лебедя (подробности об этом изображении см. на рис. 1.7). Предположим, что в данном обсуждении интересующий объект — центральная светлая область, а составляющие меньших размеров являются шумовыми. Задача состоит в том, чтобы удалить шум. На рис. 9.38(*б*) представлен результат размыкания исходного изображения по плоскому кругу радиусом 1 пиксель и затем замыкания по тому же примитиву. На рис. 9.38(*в*) и (*г*) показаны результаты аналогичной операции с использованием примитивов радиусом соответственно 3 и 5 пикселей. Как того можно было ожидать, эта последовательность демонстрирует постепенное удаление мелких составляющих по мере увеличения размеров примитива. На последнем рисунке видно, что интересующий объект выделен из шума. Шумовые составляющие в нижней части изображения устранены не полностью, поскольку они плотно расположены.

Изображения на рис. 9.38 получены путем размыкания исходного изображения и затем замыкания результата первой операции. Иногда применяется процедура *попеременной последовательной фильтрации*, при которой первая операция размыкания применяется к исходному изображению, а последую-



а б
в г

Рис. 9.38. (*а*) Рентгеновское изображение (размерами 566×566) сверхновой в Петле Лебедя, полученное с помощью телескопа «Хаббл» агентства NASA. (*б*)—(*г*) Результаты применения к исходному изображению пары операций морфологического размыкания и затем размыкания с круглым примитивом радиусом соответственно 1, 3 и 5 пикселей. (Исходное изображение предоставлено агентством NASA)

щие шаги используют для замыкания и размыкания результат, полученный на предыдущем шаге. Такой алгоритм фильтрации может быть полезен при автоматическом анализе изображений, когда полученные на каждом шаге результаты подвергаются заданным измерениям. В общем случае этот метод при тех же размерах примитива приводит к большему размыванию изображения, чем алгоритм, который демонстрируется на рис. 9.38.

Морфологический градиент

Дилатация и эрозия могут применяться вместе с вычитанием изображений для вычисления *морфологического градиента* изображения, обозначаемого g:

$$g = (f \oplus b) - (f \ominus b). \qquad (9.6\text{-}11)$$

См. определение градиента изображения в разделе 3.6.4.

Дилатация расширяет области на изображении, а эрозия сжимает их. Разность полученных изображений подчеркивает перепады между областями. Однотонные области почти не подвергаются изменениям (если примитив имеет относи-

а б
в г

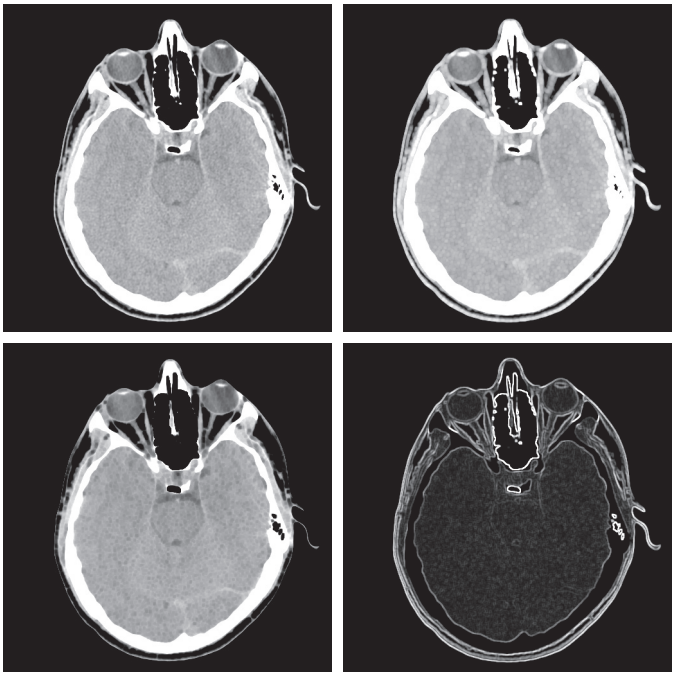


Рис. 9.39. (*а*) Изображение размерами 512×512 пикселей — одиночный скан головы при компьютерной томографии. (*б*) Результат дилатации. (*в*) Результат эрозии. (*г*) Морфологический градиент исходного изображения, вычисленный как разность (*б*) и (*в*). (Исходное изображение предоставил д-р Дэвид Р. Пикенс, университет Вандербилта)

тельно малые размеры) и при вычитании принимают значения, близкие к нулю. Общий результат состоит в получении изображения с усиленными контурами и ослабленными однородными областями, что дает эффект, сходный с дифференцированием (градиентный).

В примере на рис. 9.39(*а*) приведено исходное КТ-изображение головы, а два следующие рисунка демонстрируют результаты вычисления его размыкания и замыкания по квадратному примитиву 3×3 со всеми единичными элементами. Видны упомянутые выше утолщение и утончение деталей. На рис. 9.39(*г*) представлен результат вычисления морфологического градиента с помощью операции (9.6-11). Как и можно было ожидать, отчетливо выделяются резкие перепады яркости на исходном изображении.

Преобразования «выступ» и «впадина»

Результатом совместного применения операций вычитания изображений и морфологического размыкания или замыкания являются так называемые преобразования «выступ» и «впадина»[16]. Морфологическое преобразование *выступ* полутонового изображения f определяется как разность f и результата его размыкания:

$$T_{hat} = f - (f \circ b). \tag{9.6-12}$$

[16] В оригинале использованы термины *top-hat* («выступ») и *bottom-hat* («впадина»). — *Прим. перев.*

Аналогично морфологическое преобразование *впадина* полутонового изображения f определяется как разность результата размыкания f и самого исходного изображения:

$$B_{hat} = (f \bullet b) - f. \tag{9.6-13}$$

Одно из главных применений этих преобразований состоит в том, чтобы удалить на изображении объекты с помощью операции размыкания или замыкания по такому примитиву, который не вмещается в объекты, подлежащие удалению. После этого путем вычитания формируется изображение, в котором остаются только удаленные компоненты. Преобразование «выступ» применяется для светлых объектов на темном фоне, а преобразование «впадина» — в противоположной ситуации, отчего эти два преобразования часто называют, соответственно, *белый выступ* и *черный выступ*.

Другим важным применением преобразований *выступ/впадина* является коррекция эффекта неравномерности освещения. Как мы увидим в следующей главе, равномерное освещение играет центральную роль в процессе выделения объектов на фоне изображения. Этот процесс называется *сегментацией* и является одним из первых шагов при автоматическом анализе изображений. Часто применяемым методом сегментации является пороговая обработка исходного цифрового изображения.

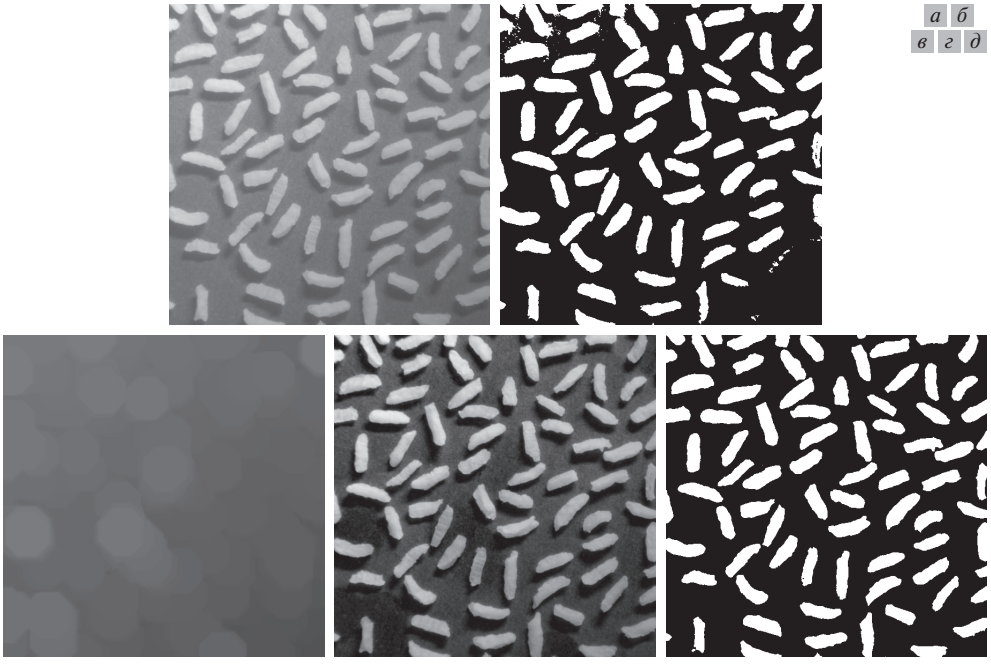


Рис. 9.40. Применение преобразования «выступ» для коррекции неравномерности фона изображения. (*а*) Исходное изображение размерами 600×600 пикселей. (*б*) Изображение после пороговой обработки. (*в*) Результат операции размыкания по круглому примитиву радиусом 40 пикселей. (*г*) Результат преобразования «выступ» (исходное изображение минус результат его размыкания). (*д*) Результат пороговой обработки преобразованного изображения

В качестве иллюстрации рассмотрим рис. 9.40(*а*), где приведено изображение зерен риса, имеющее размеры 600×600 пикселей. Это изображение было получено при неравномерном освещении, о чем свидетельствует более темный правый нижний угол. На рис. 9.40(*б*) приведен результат оптимальной пороговой обработки исходного изображения методом Оцу, который рассматривается в разделе 10.3.3. Общий эффект неравномерного освещения проявляется в ошибках сегментации в темной области (несколько зерен не выделены из фона), а также в ошибочной классификации некоторых участков фона слева вверху в качестве объектов. На рис. 9.40(*в*) показан результат размыкания исходного изображения по круглому примитиву радиусом 40 пикселей. Этот примитив взят достаточно большим, чтобы вместить любой объект, поэтому все объекты удаляются, а остается только приближенное значение фона. На данном изображении хорошо виден характер неравномерности освещения. После вычитания этого изображения из исходного (т. е. выполнения преобразования «выступ») фон становится более равномерным, как показывает рис. 9.40(*г*). Значения фона не в точности одинаковы, но разница между наибольшим и наименьшим значениями уменьшилась, и этого стало достаточно для правильной сегментации тем же методом Оцу, в результате чего выделяются все зерна, как показывает рис. 9.40(*д*).

Гранулометрия

Применительно к цифровой обработке изображений *гранулометрия* — это область, которая занимается в основном изучением распределения размеров частиц на сложных изображениях. В практических задачах частицы редко бывают аккуратно отделены друг от друга, а обычно они перекрываются, что делает идентификацию отдельных частиц на изображении трудной задачей. Для косвенной оценки распределения частиц по размеру можно применить морфологический подход, при котором не требуется идентифицировать и измерять каждую частицу по отдельности.

Данный подход основан на простом принципе. Пусть частицы имеют правильную форму и они светлее, чем фон изображения. К исходному изображению применяется операция размыкания по примитиву с увеличением размеров примитива на каждом шаге. В основе подхода лежит та идея, что операция размыкания по примитиву некоторых конкретных размеров в наибольшей степени воздействует на те области исходного изображения, в которых содержатся частицы аналогичных размеров. После каждого шага размыкания вычисляется *сумма* значений пикселей полученного изображения. Эта сумма, которую естественно назвать *интегральной яркостью*, убывает с увеличением размеров примитива, потому что, как уже говорилось, операция размыкания приводит к уменьшению яркости светлых деталей. В результате такой процедуры строится числовой массив, в котором каждый элемент представляет сумму значений всех пикселей в результате размыкания исходного изображения по примитиву с размерами, соответствующими этому элементу массива. Чтобы подчеркнуть различия между последовательными шагами размыкания, вычисляются и наносятся на график разности значений соседних элементов массива. Пики на графике указывают размеры частиц, преобладающие в распределении на исходном изображении.

В качестве примера рассмотрим изображение на рис. 9.41(*а*), на котором показаны деревянные штифты двух преобладающих размеров. Фактура дерева способна стать причиной изменчивости результатов размыкания, поэтому

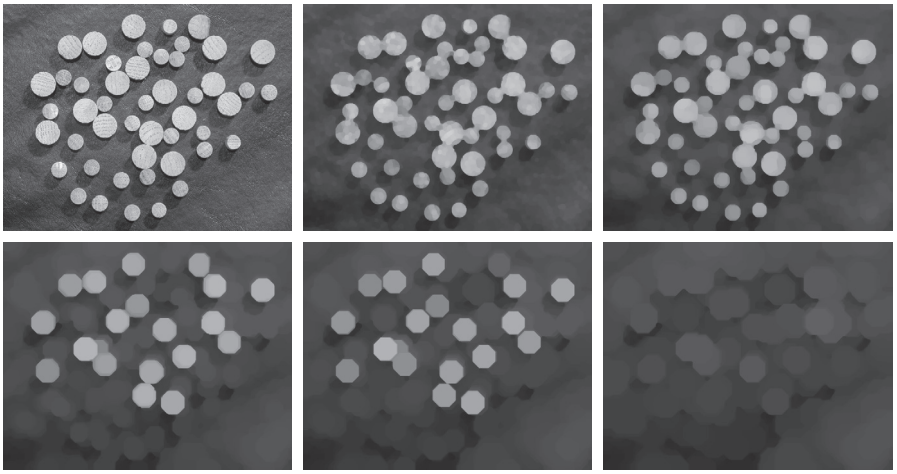


Рис. 9.41. (*а*) Исходное изображение деревянных штифтов, имеющее размеры 531×675 пикселей. (*б*) Изображение после сглаживания. (*в*)—(*е*) Результаты размыкания изображения (*б*) по круглым примитивам диаметром 10, 20, 25 и 30 пикселей соответственно. (Исходное изображение предоставил д-р Стив Эддинс, компания The MathWorks, Inc.)

разумным шагом предварительной обработки является сглаживание исходного изображения. На рис. 9.41(*б*) представлен результат применения рассмотренного выше морфологического сглаживающего фильтра на основе круглого примитива с радиусом 5. Рис. 9.41(*в*)—(*е*) демонстрируют результаты размыкания исходного изображения по круглому примитиву с радиусами 10, 20, 25 и 30. Можно заметить, что на рис. 9.41(*г*) вклад в яркость от малых штифтов почти устранен. На рис. 9.41(*д*) существенно ослаблен вклад от больших штифтов, а на рис. 9.41(*е*) — в еще большей степени. (Обратите внимание, что на рис. 9.41(*д*) большой штифт вверху справа намного темнее остальных из-за его меньшего размера. Эта информация может быть полезной, если стоит задача обнаружения бракованных штифтов.)

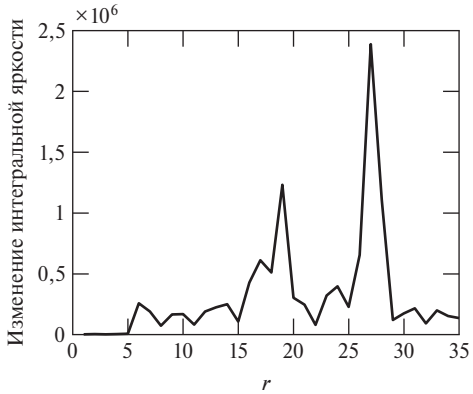


Рис. 9.42. Разности интегральных яркостей как функция от радиуса примитива *r*. Два пика на графике говорят о том, что на изображении присутствуют частицы двух преобладающих размеров

Рис. 9.42 демонстрирует полученный в данном примере график значений разности соседних элементов массива. Как отмечалось выше, можно ожидать, что такой график будет содержать значительные пики вблизи тех значений радиуса примитива, при которых он становится достаточно большим, чтобы вместить набор частиц приблизительно с теми же размерами. На графике рис. 9.42 видны два отчетливых пика, что ясно указывает на присутствие в изображении объектов двух преобладающих размеров.

Текстурная сегментация

На рис. 9.43(*а*) представлено зашумленное изображение темных пятен на светлом фоне, состоящее из двух областей с различными текстурами. Правая часть изображения содержит круглые пятна большего диаметра, чем те, которые находятся в левой части. Задача состоит в нахождении границы между этими двумя областями на основе их текстурных признаков (текстура рассматривается в разделе 11.3.3). Как уже говорилось, процесс разбиения цифрового изображения на области называется *сегментацией* и рассматривается в главе 10.

а б
в г

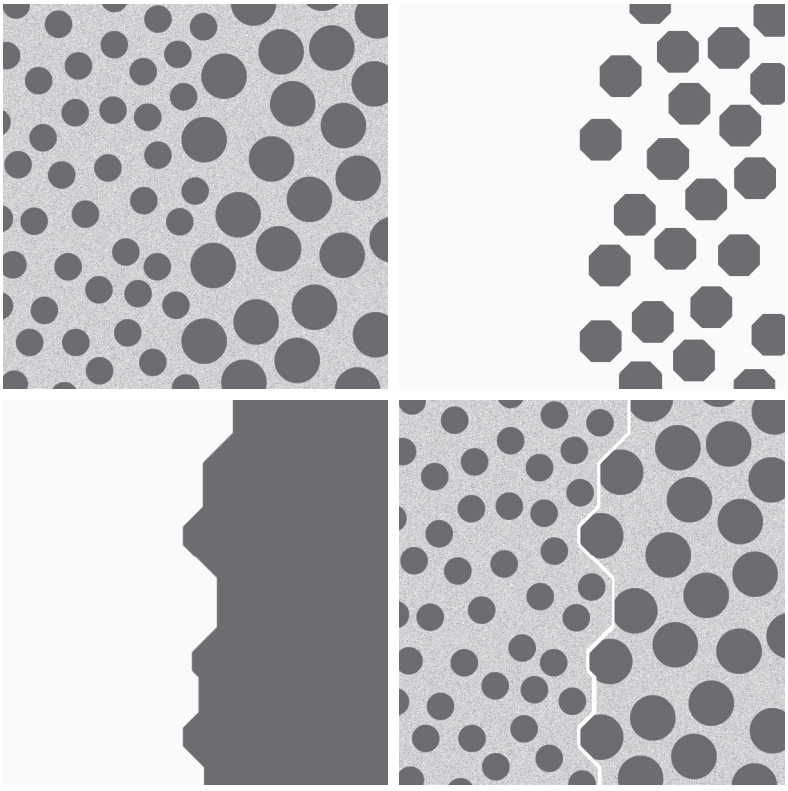


Рис. 9.43. Текстурная сегментация. (*а*) Исходное изображение размерами 600×600, содержащее пятна двух видов. (*б*) Изображение, на котором пятна меньшего размера удалены операцией замыкания (*а*). (*в*) Изображение, на котором просветы между пятнами большего размера удалены операцией размыкания (*б*). (*г*) Исходное изображение с наложенной границей между двумя областями из (*в*). Граница получена с помощью операции морфологического градиента

Интересующие объекты темнее фона, и мы знаем, что с помощью операции замыкания по примитиву с размерами, большими чем пятна меньшего размера, эти пятна будут удалены. Это подтверждает результат на рис. 9.43(*б*), полученный с помощью круглого примитива радиусом 30 пикселей (радиус малых пятен равен приблизительно 25 пикселей). Итак, получено изображение, содержащее только крупные темные пятна на светлом фоне. Если теперь применить к нему операцию *размыкания* по примитиву, размеры которого превышают расстояние между оставшимися темными пятнами, то просветы между пятнами исчезают, так что и сами пятна, и промежутки между ними будут приблизительно одинакового темного цвета. Этот результат представлен на рис. 9.43(*в*), полученном с помощью круглого примитива радиусом 60 пикселей.

Применяя к этому изображению операцию морфологического градиента по примитиву размерами, скажем, 3×3 со всеми единичными элементами, мы получим границу между двумя областями на рисунке рис. 9.43(*в*). На рис. 9.43(*г*) показана эта граница, наложенная на исходное изображение. Можно сказать, что все пиксели справа от границы принадлежат к области с текстурой в виде крупных пятен, а слева — с текстурой в виде мелких пятен. Полезно проработать этот пример более подробно, воспользовавшись аналогией с шариком на рис. 9.36.

9.6.4. Полутоновая морфологическая реконструкция

Морфологическая реконструкция полутоновых изображений определяется в основном так же, как и для двухградационных изображений в разделе 9.5.9. Пусть f — изображение-маркер, а g — изображение-маска. Мы предполагаем, что оба они являются полутоновыми изображениями одинаковых размеров, причем $f \le g$. *Геодезическая дилатация* f первого порядка относительно маски g определяется по формуле

$$D_g^{(1)}(f) = (f \oplus g) \downarrow g, \tag{9.6-14}$$

Ясно, что в этих выражениях фигурируют функции от (x, y), но мы для краткости опускаем координаты.

где $\downarrow$ обозначена поэлементная операция выбора минимума. Согласно этому соотношению, геодезическая дилатация первого порядка находится так: сначала вычисляется дилатация f по заданному примитиву b, а затем в каждой точке (x, y) выбирается меньшее из значений полученного результата и маски g. Дилатация в (9.6-14) вычисляется по формуле (9.6-2) для плоского примитива b или по формуле (9.6-4) в случае неплоского. Геодезическая дилатация порядка n от f относительно g определяется рекуррентным соотношением

$$D_g^{(n)}(f) = D_g^{(1)}\left[D_g^{(n-1)}(f)\right], \tag{9.6-15}$$

где $D_g^{(0)}(f) = f$.

Аналогично *геодезическая эрозия* первого порядка от маркера f относительно маски g определяется по формуле

$$E_g^{(1)}(f) = (f \ominus b) \uparrow g, \tag{9.6-16}$$

где $\uparrow$ обозначена поэлементная операция выбора максимума. Геодезическая эрозия порядка n определяется рекуррентным соотношением

$$E_g^{(n)}(f) = E_g^{(1)}\left[E_g^{(n-1)}(f)\right], \tag{9.6-17}$$

где $E_G^{(0)}(F) = F$.

См. задачу 9.33 касательно набора отношений двойственности между выражениями данного раздела.

Морфологическая реконструкция через дилатацию полутонового изображения-маркера f по полутоновому изображению-маске g определяется как результат геодезической дилатации f по g, которая выполняется повторно до получения стабильного результата, т. е.

$$R_g^D(f) = D_g^{(k)}(f) \tag{9.6-18}$$

при таком k, что $D_g^{(k)}(f) = D_g^{(k+1)}(f)$. Аналогично *морфологическая реконструкция через эрозию f* по g определяется как

$$R_g^E(f) = E_g^{(k)}(f) \tag{9.6-19}$$

при таком k, что $E_g^{(k)}(f) = E_g^{(k+1)}(f)$.

Как и в двоичном случае, операция размыкания с реконструкцией полутоновых изображений состоит в том, что исходное изображение сначала подвергается эрозии и затем используется в качестве маркера. *Размыкание с реконструкцией* порядка n от изображения f есть по определению реконструкция через дилатацию для результата эрозии порядка n от изображения f по самому f, т. е.

$$O_R^{(n)}(f) = R_f^D\left[(f \ominus nb)\right], \tag{9.6-20}$$

где $(f \ominus nb)$ означает n-кратное применение эрозии f по b, как объяснялось в разделе 9.5.7. Вспомним из обсуждения уравнения (9.5-27) в случае двоичных изображений, что цель размыкания с реконструкцией состоит в сохранении формы деталей изображения после операции эрозии. Аналогично *замыкание с реконструкцией* порядка n от изображения f определяется как реконструкция через эрозию для результата дилатации порядка n от изображения f по самому f, т. е.

$$C_R^{(n)}(f) = R_f^E\left[(f \oplus nb)\right], \tag{9.6-21}$$

где $(f \oplus nb)$ означает n-кратное применение дилатации f по b. Учитывая двойственность, замыкание с реконструкцией для данного изображения можно вычислить следующим образом: построить дополнение, вычислить для него размыкание с реконструкцией и затем найти дополнение результата. В заключение изложим полезный, как демонстрирует следующий пример, метод, который можно назвать *преобразованием «выступ» с реконструкцией*. Этот метод состоит в вычитании из исходного изображения результата его размыкания через реконструкцию.

Пример 9.11. Применение морфологической реконструкции для выравнивания сложного фона.

■ В данном примере демонстрируется использование полутоновой морфологической реконструкции в составе последовательности шагов для нормализации не-

регулярного фона изображения на рис. 9.44(*а*), когда задача состоит в том, чтобы оставить только надписи на фоне с постоянной яркостью. Решение этой задачи является хорошей иллюстрацией силы морфологических концепций. Начнем с подавления горизонтальных бликов в верхней части клавиш. Эти блики шире любого символа на изображении, поэтому их скорее всего удастся подавить с помощью операции размыкания с реконструкцией, если эрозию выполнять по примитиву в форме длинного горизонтального отрезка. В результате такой операции будет получен фон, на котором имеются клавиши и блики на них. Вычитание этого результата из исходного изображения (т. е. применение преобразования «выступ» с реконструкцией) приводит к устранению в нем горизонтальных бликов и выравнивает яркость фона.

На рис. 9.44(*б*) приведен результат размыкания с реконструкцией исходного изображения, с вычислением эрозии по примитиву в форме горизонтального отрезка размерами 1×71 пикселей. Здесь можно было применить для устранения символов обычное размыкание, но тогда полученный фон был бы неравномерным, как показывает рис. 9.44(*в*) (сравните, например, области между клавишами на этих двух изображениях). На рис. 9.44(*г*) приведены результаты вычитания изображения на рис. 9.44(*б*) из исходного изображения. Как можно было ожидать, горизонтальные блики и неравномерности фона устранены. Для сравнения на рис. 9.44(*д*) показан результат обычного преобразования «выступ» (т. е. вычитания из исходного изображения результата примененного к нему «обычного» размыкания, рассмотренного ранее). Как можно было ожидать по характеристикам фона на рис. 9.44(*в*), фон на рис. 9.44(*д*) далеко не такой равномерный, как выше на рис. 9.44(*г*).

Следующий шаг направлен на удаление хорошо заметных на рис. 9.44(*г*) вертикальных бликов сбоку на клавишах. Это можно сделать операцией размыкания с реконструкцией, применяя линейчатый примитив с шириной, приблизительно равной ширине бликов (в данном случае около 11 пикселей). Рис. 9.44(*е*) демонстрирует применения такой операции к изображению на рис. 9.44(*г*). Вертикальные блики устранены, но вместе с ними исчезли тонкие вертикальные штрихи в составе символов (например буква I в надписи SIN), поэтому необходимо найти способ их восстановления. Ошибочно удаленные символы находятся очень близко к другим символам, поэтому если бы мы использовали для прочих символов дилатацию по горизонтали, расширившиеся символы перекрыли бы область, ранее занятую пропавшими символами. Этот эффект демонстрирует рис. 9.44(*ж*), полученный дилатацией изображения на рис. 9.44(*е*) по примитиву размерами 1×21 пиксель.

Все, что сейчас осталось, — это восстановить пропавшие символы. Рассмотрим изображение на рис. 9.44(*з*), сформированное поэлементной операцией выбора минимального значения между результатами дилатации на рис. 9.44(*ж*) и преобразования «выступ» с реконструкцией на рис. 9.44(*г*). Хотя этот результат кажется более близким к желаемому, заметим, что символ I в составе SIN попрежнему отсутствует. Используя это изображение как маркер, а результат дилатации как маску и применяя полутоновую операцию реконструкции согласно (9.6-18), получаем окончательный результат, представленный на рис. 9.44(*и*). Как видно на этом изображении, все символы выделены из неравномерного фона исходного изображения, включая фон клавиш. На полученном изображении фон одинаков по всему полю. ■

в *б* *в*
г *д* *е*
ж *з* *и*

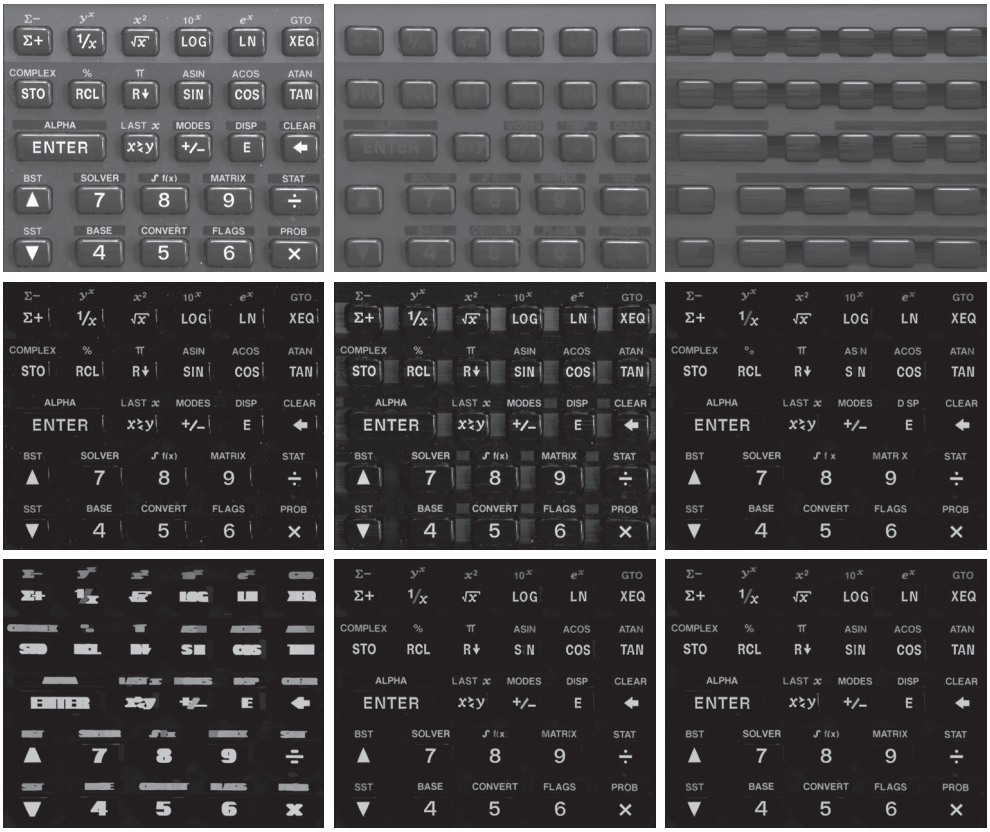


Рис. 9.44. (*а*) Исходное изображение размерами 1134×1360 пикселей. (*б*) Изображение (*а*) после операции размыкания с реконструкцией, в которой эрозия выполнялась по примитиву в форме горизонтального отрезка длиной 71 пиксель. (*в*) Изображение после размыкания (*а*) по такому же линейному примитиву. (*г*) Результат преобразования «выступ» с реконструкцией. (*д*) Результат обычного преобразования «выступ». (*е*) Результат размыкания с реконструкцией (*г*) по горизонтальному линейному примитиву длиной 11 пикселей. (*ж*) Результат дилатации (*е*) по горизонтальному линейному примитиву длиной 21 пиксель. (*з*) Результат поэлементной операции взятия минимума для (*г*) и (*ж*). (*и*) Конечный результат реконструкции. (Изображения предоставил д-р Стив Эддинс, компания The MathWorks, Inc.)

Заключение

Представленные в этой главе морфологические понятия и методы образуют мощный набор инструментов для выделения интересующих признаков в изображении. Один из наиболее привлекательных аспектов морфологических методов обработки изображений состоит в их исчерпывающем теоретическом обосновании. С точки зрения реализации важным преимуществом является то,

что и дилатация, и эрозия представляют собой примитивные операции, которые лежат в основе широкого класса морфологических алгоритмов. Как будет показано в следующей главе, на базе морфологии могут быть разработаны процедуры сегментации изображений для многочисленных приложений. Как обсуждается в главе 11, морфологические методы играют важную роль в процедурах построения описаний изображений.

Ссылки и литература для дальнейшего изучения

Книга Серра [Serra, 1982] является фундаментальным источником по морфологической обработке изображений. См. также работы [Serra, 1988], [Giardina, Dougherty, 1988] и [Haralick, Shapiro, 1992]. К числу дополнительных более ранних источников, связанных с предметом рассмотрения, относятся статьи [Blum, 1967], [Lantuéjoul, 1980], [Maragos, 1987] и [Haralick et al., 1987]. Обзор методов двоичной и полутоновой морфологии проведен в работах [Basart, Gonzalez, 1992] и [Basart et al., 1992]. Этот набор источников является достаточно широкой базой для материала, изложенного в разделах 9.1—9.4. Хороший обзор материала, относящегося к разделам 9.5 и 9.6, дан в книге [Soille, 2003].

Важные вопросы, относящиеся к реализации морфологических алгоритмов, подобных изложенным в разделах 9.5 и 9.6, рассматриваются в статьях [Jones, Svalbe, 1994], [Park, Chin, 1995], [Sussner, Ritter, 1997], [Anelli et al., 1998] и [Shaked, Bruckstein, 1998]. Статья [Vincent, 1993] является особенно важной в плане подробностей практической реализации полутоновых морфологических алгоритмов. См. также книгу [Gonzalez, Woods, Eddins, 2004].

Дополнительный материал в области теории и приложений морфологической обработки изображений имеется в книге [Goutsias, Bloomberg 2000] и специальном выпуске журнала «Распознавание образов» [*Pattern Recognition*, 2000]. См. также недавний обзор литературы, составленный А. Розенфельдом [Rosenfeld, 2000]. Кроме того, представляют интерес книги по обработке двоичных изображений [Marchand-Maillet, Sharaiha, 2000] и по алгебре изображений [Wilson, 2001]. Примерами недавних работ, в которых цифровая обработка изображений ведется с применением морфологических методов, являются статьи [Kim, 2005] и [Evans, Liu, 2006].

Задачи

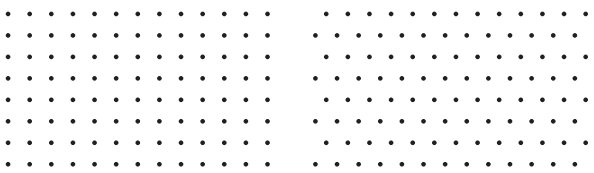
Подробные решения задач, отмеченных звездочкой, приводятся на посвященном книге сайте в Интернете. Там же можно найти темы предлагаемых проектов, основанных на материале данной главы.

9.1 Обсуждаемые в этой книге цифровые изображения построены на квадратной сетке пикселей, которые могут находиться в отношениях 4-, 8- или m-связности. Возможны, однако, и другие формы сетки. В частности, иногда применяется гексагональная сетка, в которой рассматривается 6-связность (см. рисунок ниже).

(а) Как бы Вы преобразовали изображение с квадратной сеткой элементов в изображение с гексагональной сеткой?

(б) Рассмотрите и сравните инвариантность формы объектов к повороту в случае квадратной сетки и в случае гексагональной сетки.

(в) Могут ли на гексагональной сетке возникать неоднозначные диагональные конфигурации, как в случае 8 связности? (См. раздел 2.5.2.)



9.2 ★ **(а)** Предложите морфологический алгоритм для преобразования 4-связной двоичной границы в m-связную (см. раздел 2.5.2). Вы можете предполагать, что граница имеет толщину в 1 пиксель и состоит из одной компоненты связности.

(б) Требует ли работа Вашего алгоритма выполнения более чем одной итерации с каждым примитивом? Ответ объясните.

(в) Зависят ли характеристики алгоритма от порядка использования примитивов? В случае отрицательного ответа докажите это; в противном случае приведите контрпример, иллюстрирующий зависимость Вашей процедуры от порядка применения примитивов.

9.3 Результат эрозии множества A по примитиву B является подмножеством A до тех пор, пока начало координат B содержится в B. Приведите пример случая, когда результат эрозии лежит вне множества A, полностью или частично.

9.4 Следующие четыре утверждения верны. Выдвиньте аргументы в поддержку их правильности. Пункт (*а*) справедлив в общем случае. Пункты (*б*)—(*г*) верны только для *дискретных* множеств. Чтобы показать правильность утверждений (*б*)—(*г*), нарисуйте квадратную дискретную сетку (подобную показанной в задаче 9.1) и приведите пример для каждого случая, используя множества точек на этой сетке. *Указание:* используйте в каждом случае минимально возможное число точек, при котором еще соблюдается проверяемое утверждение[17].

★ **(а)** Эрозия выпуклого множества по выпуклому примитиву приводит к получению выпуклого множества.

★ **(б)** Результат дилатации выпуклого множества по выпуклому примитиву не обязательно является выпуклым множеством.

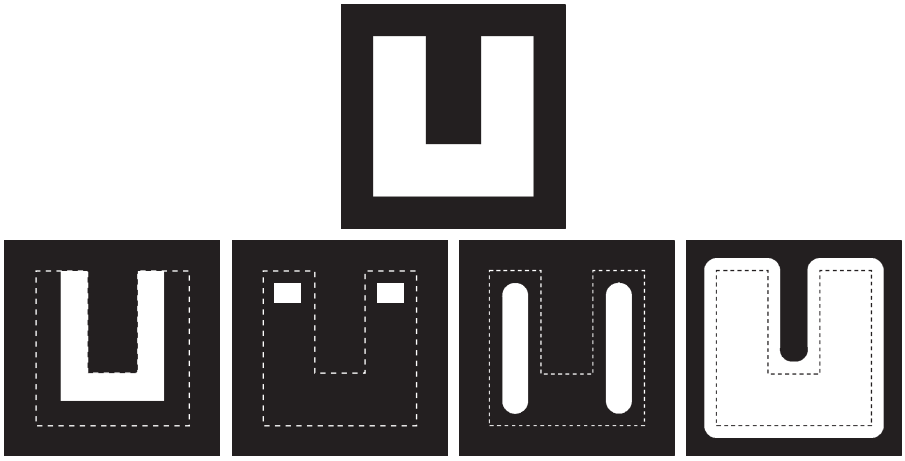
(в) Точки выпуклого дискретного множества не всегда являются связными.

(г) Существует множество точек, в котором отрезки, соединяющие всевозможные пары точек, принадлежат этому множеству, но оно не является выпуклым.

★**9.5** Используя приведенное изображение, укажите, какой (или какими) морфологической операцией и по какому примитиву получен каждый из рисунков (*а*)—(*г*). Укажите начало координат каждого примитива. Пунктирные линии обозначают границу исходного множества и приведены для сведения. Обратите внимание, что в случае (*г*) все углы закруглены.

---

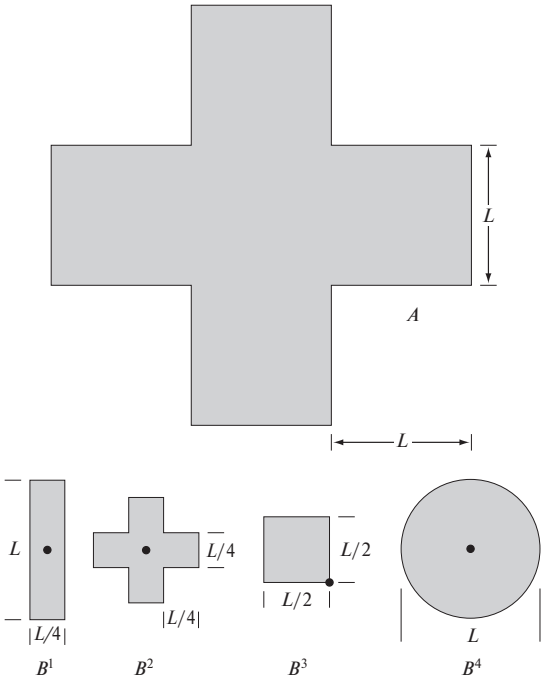
[17] См. также примечание 7 в данной главе. — *Прим. перев.*



а б в г

9.6 Пусть A — множество, показанное темным цветом на следующем рисунке, а ниже него изображены четыре вида примитивов (черными точками обозначены их начала координат). Изобразите результаты выполнения следующих морфологических операций:

(а) $(A \ominus B^2) \oplus B^4$.
(б) $(A \ominus B^1) \oplus B^3$.
(в) $(A \ominus B^3) \oplus B^1$.
(г) $(A \oplus B^3) \ominus B^2$.



★**9.7** **(а)** Чем может ограничиваться многократная дилатация изображения, если не используется тривиальный примитив, состоящий из единственной точки?

(б) Какое минимальное изображение может использоваться в качестве начального в Вашем ответе на пункт (*а*) без нарушения правильности рассуждений?

9.8 **(а)** Чем может ограничиваться многократная эрозия изображения, если не используется тривиальный примитив, состоящий из единственной точки?

(б) Какое минимальное изображение может использоваться в качестве начального в Вашем ответе на пункт (*а*) без нарушения правильности рассуждений?

★**9.9** Альтернативное определение эрозии формулируется следующим образом:

$$A \ominus B = \left\{ w \in Z^2 \middle| w + b \in A, \forall b \in B \right\}.$$

Покажите, что это определение эквивалентно определению (9.2-2).

9.10 **(а)** Покажите, что определение из задачи 9.9 эквивалентно еще одному определению эрозии:

$$A \ominus B = \bigcap_{b \in B} (A)_{-b}$$

(при замене $-b$ на b данное выражение называют также *разностью Минковского* двух множеств).

(б) Покажите, что выражение в пункте (а) также эквивалентно определению (9.2-2).

★**9.11** Альтернативное определение дилатации формулируется следующим образом:

$$A \oplus B = \left\{ w \in Z^2 \middle| w = a + b \text{ для некоторых } a \in A \text{ и } b \in B \right\}.$$

Покажите, что это определение эквивалентно определению (9.2-4).

9.12 **(а)** Покажите, что определение из задачи 9.11 эквивалентно еще одному определению дилатации:

$$A \oplus B = \bigcup_{b \in B} (A)_b$$

(данное выражение называют также *суммой Минковского* двух множеств).

(б) Покажите, что выражение в пункте (а) также эквивалентно определению (9.2-4).

9.13 Докажите правильность соотношений двойственности (9.2-6) и (9.2-5).

★**9.14** Докажите правильность соотношений двойственности $(A \bullet B)^c = (A^c \circ \hat{B})$ и $(A \circ B)^c = (A^c \bullet \hat{B})$.

9.15 Докажите правильность следующих утверждений:

★ **(а)** $A \circ B$ является подмножеством A (т. е. вложенным изображением).

(б) Если C есть подмножество D, то $C \circ B$ является подмножеством $D \circ B$.

(в) $(A \circ B) \circ B = A \circ B$.

9.16 Докажите правильность следующих утверждений (предполагая, что начало координат B содержится в множестве B и что утверждения задач 9.14 и 9.15 истинны):

(а) A является подмножеством (вложенным изображением) $A \bullet B$.

(б) Если C есть подмножество D, то $C \bullet B$ является подмножеством $D \bullet B$.

(в) $(A \bullet B) \bullet B = A \bullet B$.

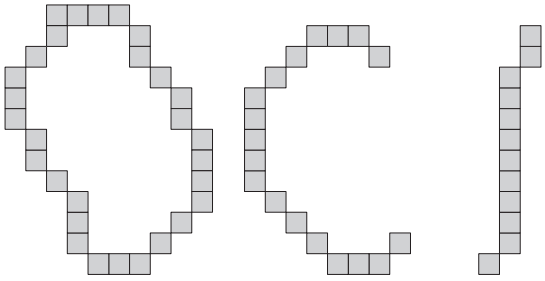
9.17 Обратитесь к показанным на рисунке изображению и примитиву B (белый кружок в правом нижнем углу изображения). Нарисуйте вид множеств C, D, E и F, получаемых при выполнении такой последовательности операций: $C = A \ominus B$; $D = C \oplus B$; $E = D \ominus B$; и $F = E \oplus B$. Исходное множество A содержит все компоненты изображения, показанные белым цветом, исключая сам примитив. Заметим, что данная последовательность операций представляет собой размыкание A по B, после чего следует замыкание по B полученного результата. Вы можете предполагать, что примитив B достаточно велик, чтобы покрыть любую из шумовых составляющих.

★9.18 Рассмотрите три показанных ниже изображения. Левое изображение содержит квадраты со сторонами 1, 3, 5, 7, 9 и 15 пикселей. Изображение в середине получено эрозией левого изображения по квадратному примитиву размерами 13×13 со всеми единичными значениями. Цель состояла в устранении всех квадратов, кроме самых больших. Наконец, правое изображение есть результат дилатации изображения в центре по тому же примитиву, чтобы восстановить форму самых крупных квадратов. Как вы знаете, эрозия и затем дилатация есть операция размыкания изображения, которая в общем случае не восстанавливает в точности первоначальную форму объектов. Объясните, почему в данном случае оказалось возможным точно восстановить большие квадраты.

9.19 Изобразите результат выполнения преобразования «попадание/пропуск» для показанных ниже изображения и примитива. Четко укажите выбранные Вами центр и границу примитива.

Изображение Примитив

★**9.20** Для описания объектов изображения, полученных в результате утончения, полезно различать три их вида (озеро, залив и отрезок), показанные на следующем рисунке. Разработайте морфолого-логический алгоритм для различения этих трех типов фигур. Исходными данными для алгоритма является изображение одного из этих типов, а на выходе должно даваться его название. Вы можете предполагать, что фигуры всегда имеют толщину в 1 элемент и являются связными, но они могут появляться в любой ориентации.



Озеро Залив Отрезок

9.21 Опишите результаты, которых можно ожидать в каждом из следующих случаев:

(а) Начальная точка в алгоритме заполнения дырок, описанном в разделе 9.5.2, выбирается на границе объекта.

(б) Начальная точка в алгоритме заполнения дырок выбирается вне границ объекта.

(в) Изобразите, как будет выглядеть выпуклая оболочка фигуры из задачи 9.6, найденная по алгоритму, описанному в разделе 9.5.4. Считайте, что $L = 4$ пикселям.

9.22 ★ **(а)** Опишите эффект от использования примитива рис. 9.15(*в*) для выделения границ вместо показанного на рис. 9.13(*б*).

(б) Какой эффект вызовет использование в алгоритме заполнения дырок, основанном на соотношении (9.5-2), заполненного единицами примитива размерами 2×2 вместо примитива рис. 9.15(*в*)?

9.23 ★ **(а)** Предложите (на основе методов, описанных в разделах 9.1—9.5) метод автоматизации для примера, показанного на рис. 9.16. Можно предполагать, что сферы не касаются друг друга и краев изображения.

(б) То же самое, но в предположении, что сферы могут касаться произвольным образом друг друга и краев изображения.

★**9.24** Для алгоритма выделения связных компонент, описанного в разделе 9.5.3, требуется знать точку в каждой компоненте связности, чтобы выделить их все. Предположим, что имеется двоичное изображение, содержащее произвольное заранее неизвестное число связных компонент. Предложите автоматическую процедуру выделения связных компонент, предполагая, что точки этих компонент имеют значения 1, а точки фона — 0.

9.25 Предложите выражение на основе реконструкции через дилатацию, с помощью которого можно выделить все дырки на двоичном изображении.

9.26 Применительно к алгоритму заполнения дырок из раздела 9.5.9:

(а) Объясните, что произойдет, если все крайние элементы изображения f имеют единичные значения.

(б) Если в пункте (а) получен ожидаемый результат, объясните почему. Если нет, объясните, как исправить алгоритм для получения ожидаемых результатов.

★**9.27** Объясните, каковы будут результаты двоичных операций эрозии и дилатации по примитиву, состоящему из одного элемента со значением 1. Ответ обоснуйте.

9.28 Как объяснялось в разделе 9.6.4 в связи с уравнением (9.5-27), операция размыкания с реконструкцией восстанавливает форму компонент изображения, оставшихся после выполнения эрозии. А что делает операция замыкания с реконструкцией?

★**9.29** Докажите, что операции геодезической эрозии и дилатации (раздел 9.5.9) являются двойственными относительно дополнения множеств, т. е. докажите, что $E_G^{(n)}(F) = \left[D_{G^c}^{(1)}\left[D_{G^c}^{(n-1)}(F^c)\right]\right]^c$ и, наоборот, $D_G^{(n)}(F) = \left[E_{G^c}^{(1)}\left[E_{G^c}^{(n-1)}(F^c)\right]\right]^c$. Считайте, что примитив центрально-симметричен относительно своего центра.

9.30 Докажите, что операции реконструкции через дилатацию и через эрозию (раздел 9.5.9) являются двойственными относительно дополнения множеств, т. е. докажите, что $R_G^D(F) = \left[R_{G^c}^E(F^c)\right]^c$ и, наоборот, $R_G^E(F) = \left[R_{G^c}^D(F^c)\right]^c$. Считайте, что примитив центрально-симметричен относительно своего центра.

★**9.31** Объясните, почему:

(а) $[(F \ominus nB)]^c = \left(F^c \oplus n\hat{B}\right)$, где $(F \ominus nB)$ означает n-кратную эрозию F по B.

(б) $[(F \oplus nB)]^c = \left(F^c \ominus n\hat{B}\right)$.

9.32 Докажите, что двоичная операция замыкания с реконструкцией двойственна операции размыкания с реконструкцией относительно дополнения множеств, т. е. $O_R^{(n)}(F) = \left[C_R^{(n)}(F^c)\right]^c$ и аналогично $C_R^{(n)}(F) = \left[O_R^{(n)}(F^c)\right]^c$. Считайте, что примитив центрально-симметричен относительно своего центра.

9.33 Докажите справедливость следующих соотношений полутоновой морфологии в предположении, что b — плоский примитив. Помните, что $f^c(x, y) = -f(x, y)$ и $\hat{b}(x, y) = b(-x, -y)$:

★ **(а)** Двойственность эрозии и дилатации: $(f \ominus b)^c = f^c \oplus \hat{b}$ и $(f \oplus b)^c = f^c \ominus \hat{b}$.
(б) $(f \bullet b)^c = f^c \circ \hat{b}$ и $(f \circ b)^c = f^c \bullet \hat{b}$.

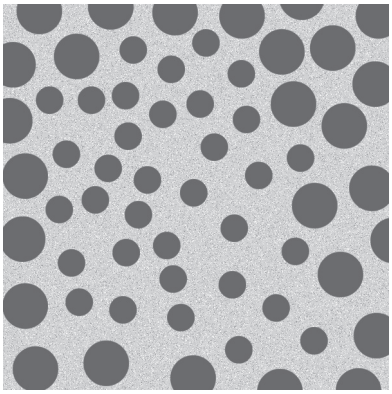
★ **(в)** $D_g^{(n)}(f) = \left[E_{g^c}^{(1)}\left[E_{g^c}^{(n-1)}(f^c)\right]\right]^c$ и $E_g^{(n)}(f) = \left[D_{g^c}^{(1)}\left[D_{g^c}^{(n-1)}(f^c)\right]\right]^c$ в предположении, что примитив центрально-симметричный.
(г) $R_g^D(f) = \left[R_{g^c}^E(f^c)\right]^c$ и $R_g^E(f) = \left[R_{g^c}^D(f^c)\right]^c$.

(д) $\left[(f \ominus nb)\right]^c = \left(f^c \oplus n\hat{b}\right)$, где $(f \ominus nb)$ означает n-кратную эрозию f по b, и также $\left[(f \oplus nb)\right]^c = \left(f^c \ominus n\hat{b}\right)$.

(е) $O_R^{(n)}(f) = \left[C_R^{(n)}(f^c)\right]^c$ и $C_R^{(n)}(f) = \left[O_R^{(n)}(f^c)\right]^c$. Считайте, что примитив центрально-симметричен относительно своего центра.

9.34 Построение границы между областями с различной текстурой на рис. 9.43 не вызвало трудностей. Рассмотрите следующее изображение, где область с кругами меньшего размера окружена областью с кругами большего размера.

(а) Будет ли метод, который использовался для получения рис. 9.43(*г*) работать и с данным изображением? Ответ обоснуйте и укажите любые предположения, необходимые для того, чтобы метод был работоспособен.
(б) Если дан положительный ответ, нарисуйте вид построенной границы.

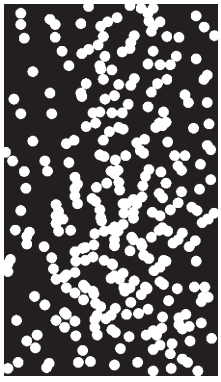


9.35 На полутоновом изображении $f(x, y)$ присутствуют неперекрывающиеся импульсы аддитивного шума. В пространственном представлении каждый импульс выглядит как небольшой цилиндр радиуса r и высотой a ($R_{\min} \le r \le R_{\max}$, $A_{\min} \le a \le A_{\max}$).

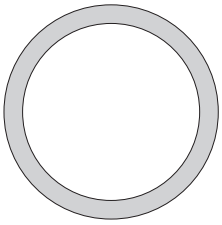
(а) Разработайте алгоритм морфологической фильтрации для устранения шума на этом изображении.
(б) Повторите пункт (а) в предположении, что импульсы шума могут перекрываться, образуя группы, содержащие не более четырех импульсов.

9.36 В прикладной задаче микроскопии на шаге предварительной обработки ставится задача выделения одиночных круглых частиц среди набора таких частиц, которые могут перекрываться, образуя группы из двух или более частиц (см. пример изображения на рисунке). Считая, что диаметр всех частиц одинаков, предложите морфологический алгоритм для построения трех изображений, которые содержали бы соответственно:

★ **(а)** только частицы, касающиеся краев изображения;
(б) только перекрывающиеся частицы;
(в) только одиночные частицы.



9.37 Высокотехнологичное производственное предприятие получило правительственный заказ на изготовление прецизионных шайб показанной на рисунке формы. По условиям контракта форма всех шайб должна контролироваться с помощью системы технического зрения. Контроль состоит в обнаружении отклонений формы внутреннего и внешнего краев шайбы от круглой. Вы можете предполагать, что: (1) имеется эталонное изображение годной шайбы и (2) используемые системы позиционирования и регистрации изображений имеют достаточную точность, позволяющую не принимать во внимание ошибки, вносимые из-за дискретизации и позиционирования. Вы приглашены в качестве консультанта для составления спецификаций той части системы технического зрения, которая осуществляет контроль качества по изображению объекта. Предложите решение на основе морфологических и/или логических операций. Ответ следует дать в форме блок-схемы.



ГЛАВА 10

СЕГМЕНТАЦИЯ ИЗОБРАЖЕНИЙ

Целое равно сумме его частей.

Евклид

Целое есть нечто большее, чем сумма его частей.

Макс Вертгеймер

Введение

Начиная с предыдущей главы, мы перешли от рассмотрения низкоуровневых методов обработки изображений, на входе и на выходе которых присутствуют изображения, к изучению методов среднего и высокого уровня, когда на вход подается изображение, а на выходе возникают некоторые атрибуты, извлеченные из этого изображения (в том смысле, как это описывалось в разделе 1.1). Сегментация — следующий важный шаг в этом направлении.

Сегментация подразделяет изображение на составляющие его области или объекты. Та степень детализации, до которой доводится такое разделение, зависит от решаемой задачи. Иначе говоря, сегментацию следует прекратить, когда интересующие объекты или области обнаружены. Например, в задаче автоматизированного контроля сборки узлов радиоэлектронной аппаратуры интерес представляет анализ изображений изготавливаемых изделий с целью выявления определенных дефектов, таких как отсутствие компонентов или разрыв контактных дорожек на плате. Поэтому не имеет смысла проводить сегментацию мельче того уровня детализации, который необходим для обнаружения подобных дефектов.

Сегментация изображений, не являющихся тривиальными, представляет собой одну из самых сложных задач обработки изображений. Конечный успех компьютерных процедур анализа изображений во многом определяется точностью сегментации, по этой причине значительное внимание должно быть уделено повышению ее надежности. В некоторых ситуациях, например в задачах технического контроля, возможно хотя бы в некоторой степени управлять условиями съемки. Опытный проектировщик системы обработки изображений неизменно обращает внимание на подобные возможности. В других прикладных задачах, например в автономных системах наведения на цель, разработчик не может контролировать окружающие условия, поэтому обычный подход состоит в том, чтобы сосредоточиться на выборе сенсоров такого вида, которые, скорее всего, будут усиливать сигнал от интересующих объектов и одновременно ослаблять влияние несущественных деталей изображения. Хорошим при-

мером такого подхода может служить съемка в инфракрасном диапазоне, используемая в военных целях для обнаружения объектов с мощным тепловым излучением, например боевой техники или движущихся войск.

Большинство рассматриваемых в этой главе алгоритмов сегментации изображений основываются на одном из двух базовых свойств сигнала яркости: разрывности и однородности. В первом случае подход состоит в разбиении изображения на основании резких изменений сигнала, таких как перепады яркости на изображении. Вторая категория методов использует разбиение изображения на области, однородные в смысле заранее выбранных критериев. Примерами таких методов могут служить пороговая обработка, выращивание областей, слияние и разбиение областей. В этой главе мы рассмотрим и проиллюстрируем некоторые из таких подходов и покажем, что улучшения качества сегментации можно достичь комбинацией методов из разных категорий, например соединяя выделение контуров с пороговым преобразованием. Мы также рассмотрим морфологический подход к сегментации, который, в частности, привлекателен тем, что сочетает в себе положительные свойства нескольких методов сегментации, изложенных в первой части данной главы. В заключение будет рассмотрено использование для сегментации изображений некоторых ключевых особенностей, характеризующих движение объектов.

> См. также разделы 6.7 и 10.3.8, где рассматриваются методы сегментации, основанные не только на значениях яркости.

10.1. Основы

Обозначим R всю пространственную область, занимаемую изображением. Сегментацию изображения можно рассматривать как процесс, который разбивает R на n подобластей $R_1, R_2, ..., R_n$ так, что

(а) $\bigcup_{i=1}^{n} R_i = R$,

(б) множество R_i является связным, $i = 1, 2, ..., n$,

> Связные множества рассматриваются в разделе 2.5.2.

(в) $R_i \cap R_j = \varnothing$ для любых i и j, $i \neq j$,

(г) $Q(R_i)$ = TRUE для $i = 1, 2, ..., n$,

(д) $Q(R_i \cup R_j)$ = FALSE для любых *смежных* областей R_i и R_j.

Здесь $Q(R_k)$ — логический предикат, определенный на точках множества R_k и принимающий истинное (TRUE) или ложное (FALSE) значение, а $\varnothing$ — пустое множество. Знаками $\cup$ и $\cap$ обозначены соответственно операции объединения и пересечения множеств, согласно определениям в разделе 2.6.4. Две области R_i и R_j называются смежными, если их объединение образует связное множество, как говорилось в разделе 2.5.2.

Условие (а) указывает, что сегментация должна быть полной, т. е. каждый пиксель должен попасть в какую-то область. Условие (б) требует, чтобы точки в области были связными в некотором заранее определенном смысле (напри-

мер 4- или 8-связными, как определялось в разделе 2.5.2). Согласно условию (в) области должны быть непересекающимися. Условие (г) касается свойств, которым должны удовлетворять пиксели сегментированной области, например $Q(R_i)$ = TRUE, если все пиксели R_i имеют одну и ту же яркость. Наконец, условие (д) указывает, что две смежные области R_i и R_j должны различаться в смысле предиката Q.[1]

Итак, мы видим, что фундаментальная проблема при сегментации состоит в разбиении изображения на области, удовлетворяющие вышеприведенным условиям. Алгоритмы сегментации монохромных изображений обычно относятся к одной из двух основных категорий, исходя из свойств значений яркости — наличия разрывов и близости значений. В первой категории предполагается, что края областей достаточно сильно отличаются как от фона изображения, так и друг от друга, что позволяет обнаруживать границу на основании локальных разрывов яркости. Преобладающий подход в этой категории — *сегментация на основе контуров*. Ко второй категории относятся методы *сегментации на основе областей*, которые осуществляют разбиение изображения на области, обладающие внутренним сходством в соответствии с набором наперед заданных критериев.

Рис. 10.1 иллюстрирует введенные понятия. На рис. 10.1(*а*) приведено изображение области с постоянной яркостью на темном фоне, также имеющем постоянную яркость. Вместе обе эти области охватывают все изображение. На рис. 10.1(*б*) представлен результат вычисления границы внутреннего региона на основе разрывов яркости. Точки внутри и снаружи границы имеют нулевые значения (черный цвет), потому что внутри областей отсутствуют разрывы яркости. Чтобы провести сегментацию изображения, мы помечаем одним способом (скажем, белым цветом) все пиксели внутри границы и на самой границе и другим способом (скажем, черным цветом) все точки снаружи границы. На рис. 10.1(*в*) показан результат такой процедуры. Как видим, он удовлетворяет условиям (а)—(в) из перечисленных в начале данного раздела. Предикат в условии (г) следующий: если пиксель находится внутри границы или на ней, он отмечен белым, в противном случае — черным. Видно, что этот предикат принимает значение TRUE для точек, отмеченных на рис. 10.1(*в*) и черным, и белым. Две выделенных области (объект и фон) также удовлетворяют условию (д).

Следующие три изображения иллюстрируют сегментацию на основе областей. Рис. 10.1(*г*) аналогичен рис. 10.1(*а*), но яркость внутренней области не постоянна, а образует текстуру. На рис. 10.1(*д*) показан результат выделения контурных перепадов на таком изображении. Ясно, что многочисленные неинформативные изменения яркости затрудняют однозначное выделение границы на исходном изображении из-за того, что многие точки с ненулевым перепадом яркости оказываются связными с истинной границей. Поэтому метод сегментации на основе контуров не подходит для данного случая. Заметим, однако, что внешняя область имеет постоянную яркость, поэтому для решения этой несложной задачи сегментации достаточно построить предикат, который бы отличал области с текстурой от областей с постоянной яркостью. Для этой

[1] В общем случае Q может задаваться составным выражением, например $Q(R_i)$ = TRUE, если средняя яркость пикселей в R_i меньше величины m_i, И если стандартное отклонение яркости этих пикселей больше σ_i, где m_i и σ_i — заданные константы.

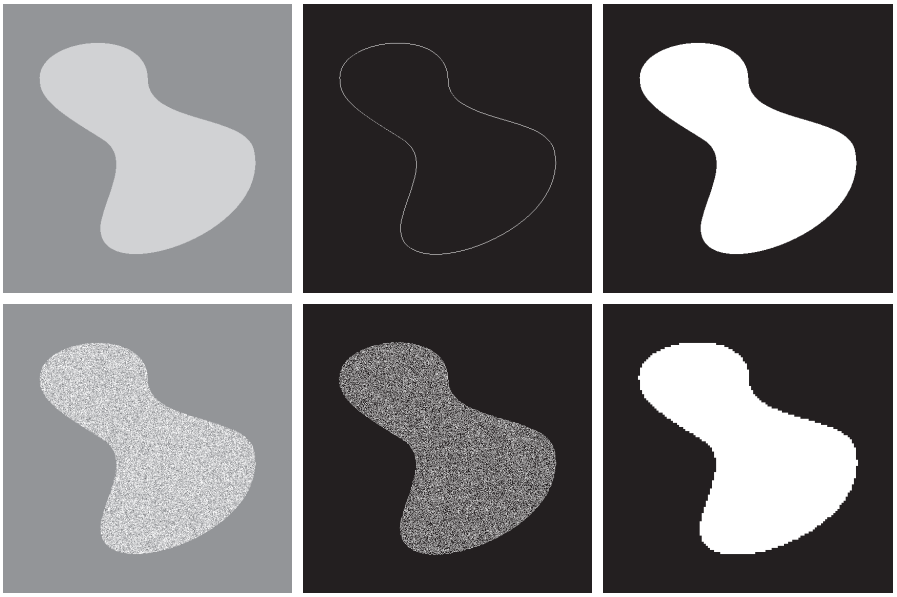


Рис. 10.1. (*а*) Изображение, содержащее область с постоянной яркостью. (*б*) Граница внутренней области, полученная на основании разрывов яркости. (*в*) Результат сегментации изображения на две области. (*г*) Изображение, содержащее область с текстурой. (*д*) Результат выделения контуров. Обратите внимание на большое число контурных перепадов внутри самой области, причем связных с границей области, что затрудняет однозначное выделение ее границы только на основе информации о перепадах. (*е*) Результат сегментации на основе свойств области

цели в качестве меры используется стандартное отклонение значений яркости, поскольку оно отлично от нуля в области с текстурой и равно нулю в области фона. На рис. 10.1(*е*) представлен результат разбиения исходного изображения на непересекающиеся подобласти размерами 4×4 пикселей. Каждая подобласть затем помечается белым, если стандартное отклонение значений яркости ее пикселей больше нуля (т. е. если предикат принимает истинное значение), или черным в противоположном случае. На границе области видна ступенчатость, потому что всем пикселям квадрата 4×4 присваивается одинаковое значение яркости. В заключение заметим, что этот результат также удовлетворяет пяти условиям, сформулированным в начале данного раздела.

10.2. Обнаружение точек, линий и перепадов

В этом разделе мы сосредоточимся на методах сегментации, основанных на обнаружении резких *локальных* изменений яркости. Нас интересуют три вида признаков, встречающихся в цифровых изображениях: изолированные точки, линии и перепады. *Пиксели перепада* — это те пиксели изображения, в которых функция яркости резко изменяется, а *перепад* (или *участок перепада*) — это связ-

ное множество пикселей перепада (понятие связности рассматривалось в разделе 2.5.2). *Детекторы перепадов* представляют собой методы локальной обработки изображения, направленные на обнаружение пикселей перепада. Линию можно рассматривать как участок перепада, у которого яркость фона с обеих сторон от линии либо значительно выше, либо значительно ниже, чем яркость пикселей линии. Фактически, как мы увидим в следующем разделе и разделе 10.2.4, линии дают так называемые «треугольные импульсы». Аналогично изолированная точка может рассматриваться как линия, у которой не только ширина, но и длина составляет один пиксель.

> Говоря о линиях, мы имеем в виду тонкие структуры обычно толщиной в несколько пикселей. Такие линии могут соответствовать, например, элементам оцифрованного строительного чертежа или дорогам на снимке из космоса.

10.2.1. Основы

Как мы видели в разделах 2.6.3 и 3.5, локальное усреднение приводит к сглаживанию изображения. Поскольку такое усреднение аналогично интегрированию, то неудивительно, что локальные изменения яркости можно обнаруживать с помощью производных. По причинам, которые вскоре станут очевидными, для этих целей особенно хорошо подходят производные первого и второго порядков.

Производная дискретной функции определяется через разности ее значений. Существуют различные способы аппроксимации производных разностями, но, как объяснялось в разделе 3.6.1, мы требуем, чтобы любая аппроксимация *первой производной* была: (1) равна нулю в областях с постоянной яркостью; (2) ненулевой в начале ступенчатого или линейного изменения яркости и (3) ненулевой на всем участке линейного изменения яркости. Аналогично от аппроксимации *второй производной* требуется, чтобы она была: (1) равна нулю в областях с постоянной яркостью; (2) ненулевой в начале *и конце* ступенчатого или линейного изменения яркости и (3) равна нулю на участке линейного изменения яркости. Поскольку мы рассматриваем величины с дискретными конечными значениями, максимально возможное изменение яркости также конечно, и кратчайшее расстояние, на котором может произойти изменение, — между соседними пикселями.

Аппроксимируем производную первого порядка одномерной функции $f(x)$ в точке x разложением функции $f(x + \Delta x)$ в ряд Тейлора в окрестности x, полагая $\Delta x = 1$ и оставляя только линейные члены (задача 10.1). В результате получим дискретную разность

$$\frac{\partial f}{\partial x} = f'(x) = f(x+1) - f(x). \qquad (10.2\text{-}1)$$

> Напомним из раздела 2.4.2, что величина шага между соседними отсчетами цифрового изображения считается единичной для удобства обозначений. Отсюда вытекает использование $\Delta x = 1$ в выражении для производной (10.2-1).

Здесь используется частная производная для сохранения единства обозначений в будущем, когда будет рассматриваться функция изображения $f(x, y)$, зависящая от двух переменных. Тогда будут использоваться частные производные вдоль пространственных осей, а в случае одномерной функции f ясно, что $\partial f/\partial x = df/dx$.

Дифференцируя выражение (10.2-1) по x, получаем выражение для второй производной:

$$\begin{aligned}\frac{\partial^2 f}{\partial x^2} &= \frac{\partial f'(x)}{\partial x} = f'(x+1) - f'(x) = \\ &= f(x+2) - f(x+1) - f(x+1) + f(x) = \\ &= f(x+2) - 2f(x+1) + f(x),\end{aligned}$$

где вторая строка следует из (10.2-1). Это разложение в ряд соответствует окрестности точки $x + 1$, а так как нас интересует вторая производная в точке x, следует вычесть всюду 1 из значения аргумента; окончательно получаем

$$\frac{\partial^2 f}{\partial x^2} = f''(x) = f(x+1) + f(x-1) - 2f(x). \qquad (10.2\text{-}2)$$

Легко проверить, что соотношения (10.2-1) и (10.2-2) удовлетворяют условиям для производных первого и второго порядков, которые были сформулированы выше. Чтобы проиллюстрировать это, а также выделить фундаментальные сходства и различия между производными первого и второго порядков в контексте цифровой обработки изображений, рассмотрим рис. 10.2.

На рис. 10.2(*а*) показано изображение, содержащее различные сплошные объекты, линию и одиночную шумовую точку. На рис. 10.2(*б*) представлен горизонтальный профиль яркости (строка сканирования) изображения приблизительно в его середине, проходящий через изолированную точку. Изменения яркости между сплошными объектами и фоном вдоль линии сканирования демонстрируют два вида перепадов: *наклонный перепад* (слева) и *ступенчатый перепад* (справа). Как говорится ниже, изменения яркости на тонких объектах, таких как линия, часто называют *треугольным импульсом*. На рис. 10.2(*в*) тот же профиль показан в упрощенном виде с небольшим числом дискретных точек, достаточным для численного анализа поведения производных первого и второго порядков в окрестности шумовых точек, линий и краев объектов. На этой упрощенной схеме наклонный перепад занимает четыре пикселя, шумовая точка — один пиксель, линия имеет толщину три пикселя, а ступенчатый перепад происходит между соседними пикселями. Число градаций яркости для простоты ограничено восемью.

Рассмотрим свойства первой и второй производных, двигаясь вдоль профиля слева направо. Прежде всего видно, что первая производная отлична от нуля в начале и на протяжении всего наклонного перепада яркости, тогда как вторая производная принимает ненулевые значения только в начале и конце наклонного перепада. Поскольку перепады на цифровых изображениях часто имеют такую форму, можно заключить, что первая производная выделяет «толстые» перепады, а вторая производная — значительно более тонкие. Затем мы встречаем изолированную шумовую точку. В этой точке величина отклика второй производной значительно больше, чем первой. Этого

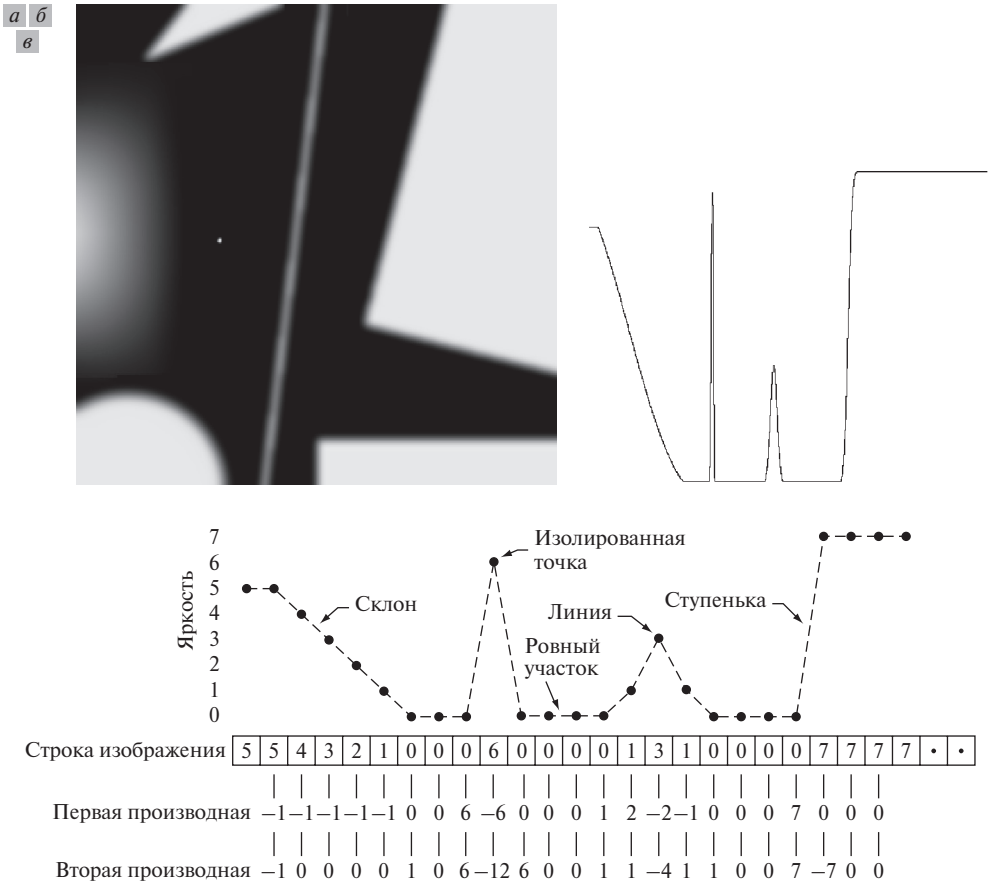


Рис. 10.2. (*а*) Изображение. (*б*) Горизонтальный профиль яркости, проходящий через изолированную шумовую точку. (*в*) Упрощенный вид профиля (соседние отсчеты для наглядности соединены пунктирной линией). Под графиком приведены значения яркости в отмеченных на профиле точках соответствующей строки изображения. Производные вычислялись с помощью соотношений (10.2-1) и (10.2-2)

можно было ожидать, потому что производная второго порядка намного сильнее реагирует на резкие изменения, чем первая производная. Таким образом, вторая производная имеет тенденцию усиливать мелкие детали (в том числе шум) в значительно большей степени, чем производная первого порядка. Линия в этом примере сравнительно тонкая и потому также является «мелкой деталью», так что мы снова видим, что на ней величина второй производной значительно выше. В заключение отметим, что и на наклонных, и на ступенчатых перепадах вторая производная имеет два всплеска противоположных знаков (переходя с отрицательного на положительный или с положительного на отрицательный в начале и в конце перепада). Этот эффект «удвоения перепада» является важной характеристикой, которая может применяться для нахождения перепадов, как мы увидим в разделе 10.2.6. Знак второй производной также позволяет определить, является ли точка началом изменения от светло-

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Рис. 10.3. Общее представление маски пространственного фильтра размерами 3×3 элемента

го к темному (отрицательная вторая производная) или от темного к светлому (положительная вторая производная), причем знак проявляется в точке начала (и конца) перепада.

В итоге мы приходим к следующим заключениям. (1) Производные первого порядка в общем случае выделяют на изображении более широкие перепады. (2) Производные второго порядка дают более сильный отклик на мелкие детали, такие как тонкие линии, изолированные точки и шум. (3) Вторые производные на наклонных и ступенчатых перепадах яркости дают отклик дважды. (4) Знак второй производной можно использовать для определения направления перепада яркости (от светлого к темному или наоборот)[2].

Метод вычисления первой и второй производных в каждом пикселе изображения состоит в использовании пространственных фильтров. Для показанной на рис. 10.3 маски фильтра размерами 3×3 элемента эта процедура состоит в вычислении линейной комбинации коэффициентов маски со значениями яркости элементов изображения, покрываемых маской. Иначе говоря, по аналогии с формулой (3.4-3) при использовании этой маски отклик в точке изображения, совпадающей с центром маски, задается выражением

$$R = w_1 z_1 + w_2 z_2 + \ldots + w_9 z_9 = \sum_{k=1}^{9} w_k z_k \,, \qquad (10.2\text{-}3)$$

где z_k — значение яркости пикселя, соответствующего коэффициенту w_k маски. Детали выполнения этой операции для всех пикселей изображения рассматривались в разделах 3.4 и 3.6. Другими словами, вычисление производных осуществляется путем пространственной фильтрации изображения с помощью пространственных масок, как говорилось в указанных разделах.[3]

---

[2] Знак первой производной для этого также годится. — *Прим. перев.*

[3] Как объяснялось в разделе 3.4.3, соотношение (10.2-3) представляет собой упрощенную запись либо пространственной корреляции, заданной уравнением (3.4-1), либо пространственной свертки, заданной уравнением (3.4-2). Следовательно, в результате вычисления R по всему изображению получается массив. Вся пространственная фильтрация в данной главе выполняется с помощью корреляции. Иногда мы используем устоявшийся термин *свертка изображения с маской*, но это делается только в тех случаях, когда маска фильтра симметрична и, значит, корреляция и свертка приводят к одному и тому же результату.

10.2.2. Обнаружение изолированных точек

С учетом выводов предыдущего раздела обнаружение изолированных точек следует проводить на основе значений второй производной. Согласно рассмотрению в разделе 3.6.2, это влечет за собой использование лапласиана

$$\nabla^2 f(x,y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}, \tag{10.2-4}$$

где частные производные находятся с использованием (10.2-2)

$$\frac{\partial^2 f(x,y)}{\partial x^2} = f(x+1,y) + f(x-1,y) - 2f(x,y) \tag{10.2-5}$$

и

$$\frac{\partial^2 f(x,y)}{\partial y^2} = f(x,y+1) + f(x,y-1) - 2f(x,y), \tag{10.2-6}$$

после чего лапласиан принимает вид

$$\nabla^2 f(x,y) = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y). \tag{10.2-7}$$

Как говорилось в разделе 3.6.2, такое выражение можно реализовать с использованием маски на рис. 3.37(*а*). Также в этом разделе говорилось, что можно дополнить выражение (10.2-7) диагональными членами и использовать маску на рис. 3.37(*б*). Воспользуемся маской лапласиана, показанной на рис. 10.4(*а*), которая совпадает с маской на рис. 3.37(*в*). Будем считать, что изолированная точка обнаруживается в позиции (x,y), совпадающей с центром маски, если абсолютное значение отклика маски превышает некоторый заданный порог. Такие точки помечаются значением 1 в выходном изображении, а все прочие точки имеют значение 0, т. е. формируется двоичное изображение. Иначе говоря, выходной результат находится с помощью следующего выражения:

$$g(x,y) = \begin{cases} 1 & \text{если } |R(x,y)| \geq T \\ 0 & \text{в противном случае,} \end{cases} \tag{10.2-8}$$

где g — выходное изображение, T — неотрицательный порог, а R вычисляется в соответствии с (10.2-3). По существу, в данной формуле измеряется взвешенная сумма разностей значений центрального элемента и его 8 соседей. Идея состоит в том, что *изолированная* точка (т. е. расположенная в однородной или почти однородной области точка, значение яркости которой существенно отличается от окружающего фона) будет заметно отличаться по яркости от ближайших соседей, а значит, будет легко обнаруживаться с помощью маски приведенного вида. В данном случае интерес представляют только достаточно большие различия (определяемые порогом T), при которых точка может считаться изолированной. Обратите внимание, что сумма коэффициентов маски равна нулю, как это обычно бывает для дифференцирующих масок, так что на областях постоянной яркости она будет давать нулевой отклик.

Пример 10.1. Обнаружение изолированных точек в изображении.

■ Проиллюстрируем сегментацию изолированных точек изображения с помощью рис. 10.4(*б*), на котором показано рентгеновское изображение лопатки

1	1	1
1	−8	1
1	1	1

а
б *в* *г*

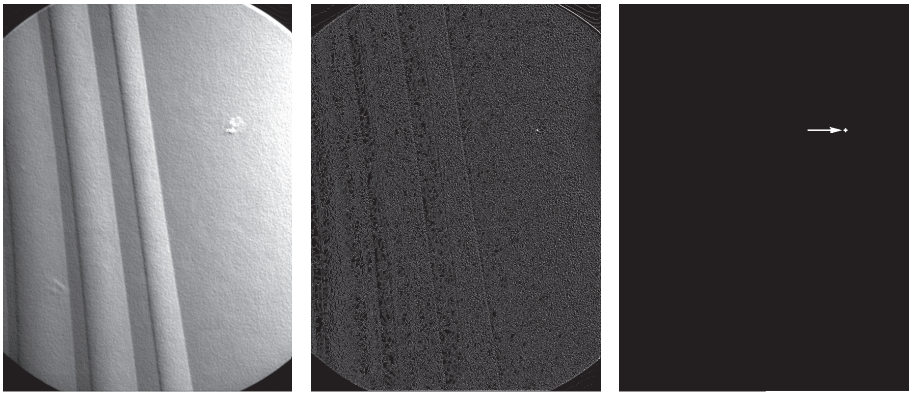


Рис. 10.4. (*а*) Маска лапласиана для обнаружения точек. (*б*) Рентгеновское изображение лопатки турбины с каверной, в области которой содержится одиночный черный пиксель. (*в*) Результат свертки изображения с маской. (*г*) Результат применения пороговой операции (10.2-8), где видна одиночная точка (точка для наглядности увеличена). (Исходное изображение предоставлено компанией X-TEK Systems Ltd.)

турбины реактивного двигателя с дефектом в виде каверны в правой верхней части. Внутри области каверны имеется одиночный пиксель черного цвета. На рис. 10.4(*в*) показан результат применения к исходному изображению маски для обнаружения точек, а на рис. 10.4(*г*) — результаты применения пороговой операции (10.2-8) со значением T, равным 90 % наибольшего по абсолютной величине значения пикселей изображения на рис. 10.4(*в*). На финальном изображении четко виден одиночный элемент (он был искусственно увеличен, чтобы оставаться заметным на иллюстрации в книге). Данный способ обнаружения изолированных точек достаточно специфичен, поскольку предполагается резкий скачок яркости в одной точке, вокруг которой (в области маски детектора) располагается однородный фон. Если такое условие не соблюдается, то для обнаружения изменений яркости более подходящими будут другие методы, рассматриваемые ниже в данной главе. ■

10.2.3. Обнаружение линий

Следующим по уровню сложности является обнаружение линий. Из обсуждения в разделе 10.2.1 мы знаем, что при обнаружении линий вторые произво-

дные будут давать более сильный отклик и выделять более тонкие линии, чем при использовании первых производных. Таким образом, можно применить изображенную на рис. 10.4(*а*) маску лапласиана и для обнаружения линий, однако необходимо учитывать эффект удвоения линий, возникающий при использовании второй производной. Данную процедуру иллюстрирует следующий пример.

Пример 10.2. Использование лапласиана для обнаружения линий.

■ На рис. 10.5(*а*) показано двоичное изображение шаблона для изготовления выводов кристалла интегральной микросхемы, имеющее размеры 486×486 пикселей, а на рис. 10.5(*б*) — изображение его лапласиана. Коль скоро лапласиан может принимать отрицательные значения[4], для его отображения необходимо выполнить градационную коррекцию; в данном случае нуль яркости был сдвинут в середину диапазона. Как видно на увеличенном фрагменте, нулевое значение представляется средним уровнем серого цвета, более темные оттенки серого соответствуют отрицательным значениям лапласиана, а более светлые — положительным. В этом фрагменте хорошо заметен эффект удвоения линий.

На первый взгляд может показаться, что с отрицательными значениями можно работать, используя просто абсолютное значение лапласиана. Однако, как видно из рис. 10.5(*в*), при таком подходе толщина линий удваивается. Более подходящим способом будет использование только положительных значений лапласиана (а в случае зашумленных изображений — только значений, превышающих некоторый положительный порог, чтобы исключить случайные флуктуации в окрестности нуля, вызванные шумом). Как показывает рис. 10.5(*г*), этот метод позволяет получить более тонкие линии, что важно для практических задач. Обратите внимание на рис. 10.5(*б*)—(*г*): в тех случаях, когда на исходном изображении ширина линии значительна по сравнению с маской лапласиана, на выходном изображении между двумя краями линии присутствует серая полоса, отражающая нулевые значения. Этого можно было ожидать. Например, когда центральный элемент маски фильтра 3×3 расположен в середине линии с постоянной яркостью и шириной 5 пикселей, отклик фильтра равен нулю, что и приводит к отмеченному выше эффекту. Говоря об обнаружении линий, мы предполагаем, что толщина линий мала по сравнению с размерами детектора. Если линия не соответствует такому предположению, ее правильнее рассматривать как область и применять для обнаружения краев области другие методы, рассматриваемые ниже в данной главе. ■

Представленный на рис. 10.4(*а*) детектор на основе лапласиана является изотропным, поэтому его отклик не зависит от направления линии (применительно к четырем направлениям, имеющимся в маске лапласиана 3×3: вертикальному, горизонтальному и двум диагональным). Часто интерес представля-

---

[4] При свертке изображения с маской, сумма коэффициентов которой равна нулю, суммарное значение пикселей выходного изображения также равно нулю (задача 3.16), откуда следует, что оно содержит как положительные, так и отрицательные значения. Для отображения результата необходимо преобразовать градационную шкалу так, чтобы значения всех пикселей были неотрицательными.

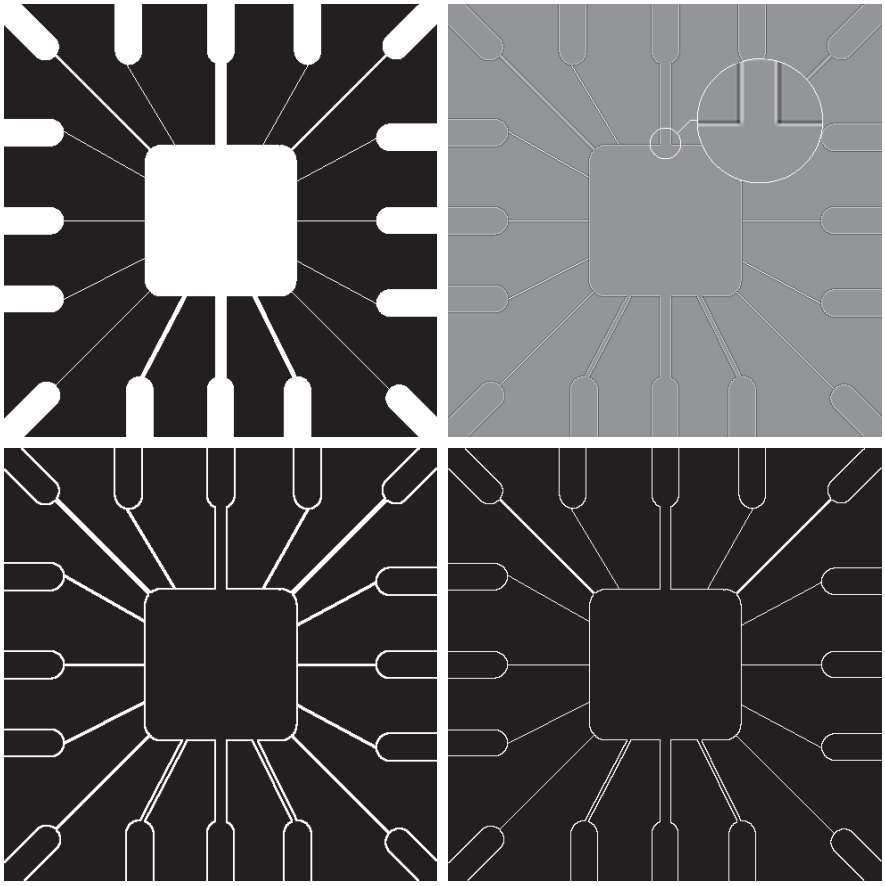


Рис. 10.5. (*а*) Исходное изображение. (*б*) Изображение лапласиана с увеличенным фрагментом, который демонстрирует характерный для лапласиана эффект удвоения линий. (*в*) Абсолютное значение лапласиана. (*г*) Положительные значения лапласиана

ет обнаружение линий *определенного* направления. Рассмотрим набор масок, показанный на рис. 10.6. Предположим, что изображение, содержащее линии с направлениями 0°, ±45° и 90° на фоне с постоянной яркостью, фильтруется с помощью первой маски. Максимальный отклик будет на горизонтальных линиях, проходящих через центр маски. Это легко проверить на простом примере однородного поля со значениями элементов, равными 1, содержащего горизонтальную линию из элементов с отличающимся значением яркости (скажем, 5). Аналогичные эксперименты подтвердят, что вторая маска на рис. 10.6 дает наибольший отклик на линиях, проходящих под углом +45°, третья — на вертикальных линиях, четвертая — на линиях под углом −45°. Предпочтительные направления каждой из масок характеризуются бóльшими значениями весовых коэффициентов (а именно 2), чем любые другие направления. Обратите внимание, что сумма коэффициентов каждой маски равна нулю, так что они будут давать нулевой отклик на областях постоянной яркости.

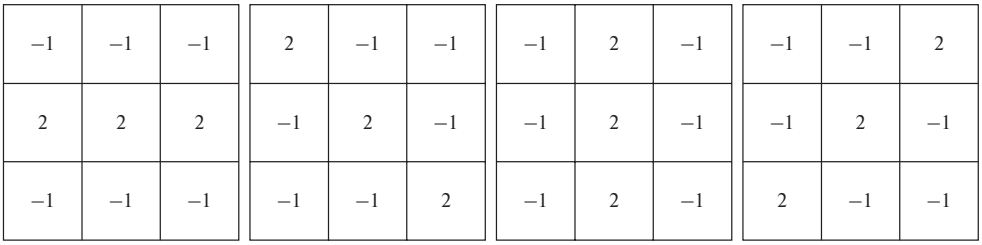


Рис. 10.6. Маски для обнаружения линий. Углы отсчитываются в системе координат, показанной на рис. 2.18(*б*)

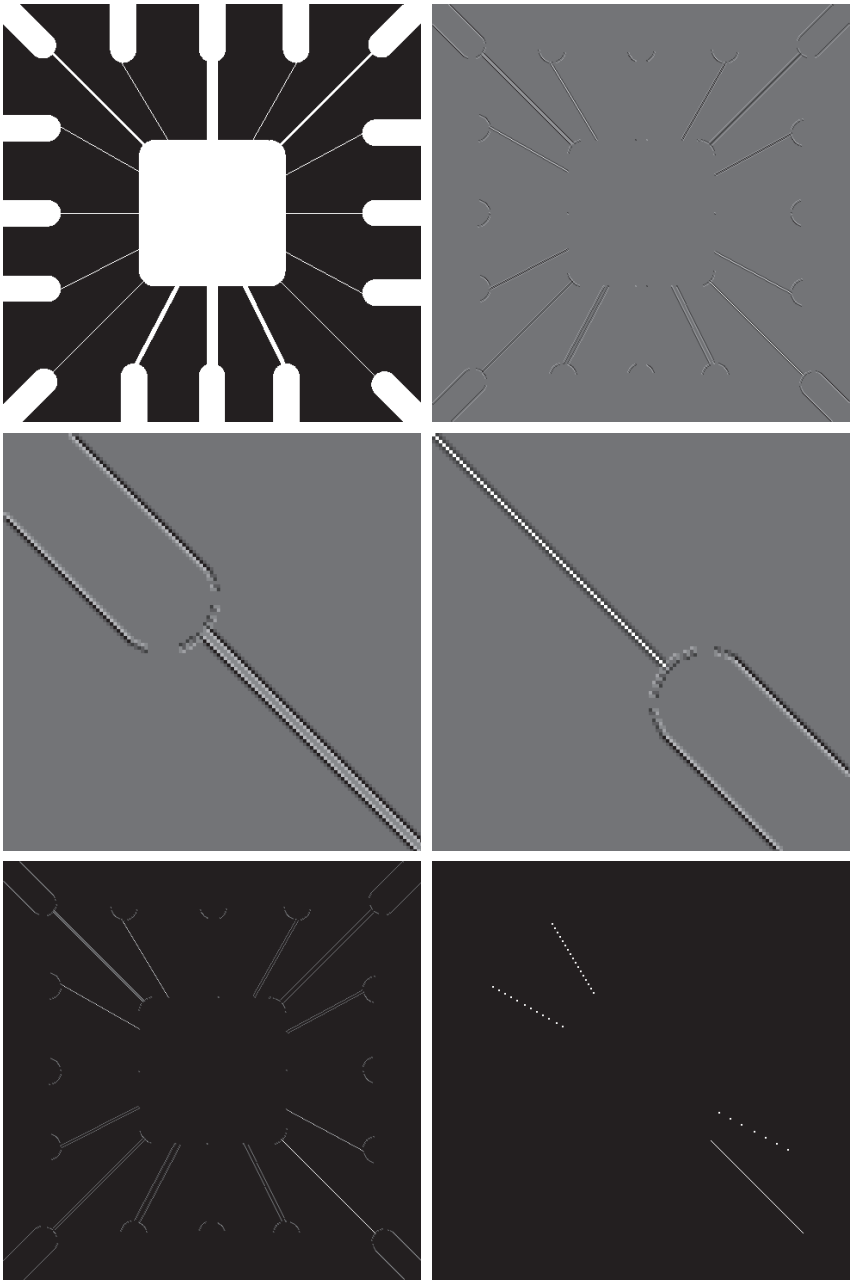
Вспомним из раздела 2.4.2, что по соглашению о системе координат цифрового изображения начало координат находится в верхнем левом углу, ось x направлена вниз, а ось y — вправо. Угол направления линии отсчитывается от оси x, так что, например, вертикальная линия на изображении имеет направление 0°, а линия с направлением +45° идет в сторону правого нижнего угла изображения.

Обозначим через R_1, R_2, R_3 и R_4 отклики масок, показанных на рис. 10.6 (слева направо), где значения R_i вычисляются согласно соотношению (10.2-3). Будем считать, что изображение независимо фильтруется с помощью каждой из этих масок. Если в некоторой точке изображения $|R_k| > |R_j|$ для всех $j \neq k$, то эта точка, скорее всего, связана с линией, ориентированной вдоль направления маски k. Например, если в какой-то точке изображения $|R_1| > |R_j|$ для $j = 2, 3, 4$, то эта точка, по-видимому, принадлежит горизонтальной линии. Наоборот, нас может интересовать поиск линий, идущих в заданном направлении. В таком случае можно обработать все изображение с помощью маски для этого направления, применяя пороговое преобразование (10.2-8) к получаемому отклику. Другими словами, если интерес представляют все линии в изображении, которые ориентированы в соответствии с направлением некоторой заданной маски, достаточно пройти этой маской по всему изображению, сравнивая абсолютное значение результата с заданным порогом. Оставшиеся при этом точки соответствуют наибольшим значениям отклика, которые в случае линий толщиной в один пиксель наиболее близки к направлению, определяемому маской. Данную процедуру иллюстрирует следующий пример.

Не следует путать R в обозначениях откликов масок с обозначениями областей, получаемых при сегментации, в разделе 10.1.

Пример 10.3. Обнаружение линий заданного направления.

■ На рис. 10.7(*а*) показано изображение, использовавшееся в предыдущем примере. Предположим, что мы хотим найти все линии, имеющие толщину в один пиксель и идущие под углом +45°. Воспользуемся для этой цели второй маской из приведенных на рис. 10.6. Результат фильтрации изображения по указанной маске приводится на рис. 10.7(*б*). Как и раньше, на рис. 10.7(*б*) пиксели, более темные, чем фон, соответствуют отрицательным значениям. На исходном изображении присутствуют две основные составляющие, направленные под углом



а *б*
в *г*
д *е*

Рис. 10.7. (*а*) Изображение шаблона соединений. (*б*) Результат обработки по маске обнаружения линий под углом +45°, показанной на рис. 10.6. (*в*) Левый верхний угол (*б*) в увеличенном виде. (*г*) Правый нижний угол (*б*) в увеличенном виде. (*д*) Изображение (*б*), в котором все отрицательные значения заменены нулями. (*е*) Белым цветом показаны все точки изображения (*б*), значения в которых удовлетворяют условию $g \geq T$, где g обозначает изображение (*д*). (Изолированные точки увеличены для большей заметности)

+45° — слева вверху и справа внизу. На рис. 10.7(*в*) и (*г*) соответствующие участки рис. 10.7(*б*) показаны в увеличенном виде. Обратите внимание, насколько ярче отрезок прямой на рис. 10.7(*г*), чем аналогичный отрезок на рис. 10.7(*в*). Причина в том, что на исходном изображении отрезок внизу справа имеет толщину в 1 пиксель, а отрезок вверху слева — бóльшую толщину. Маска приспособлена для обнаружения линий толщиной 1 пиксель, проходящих под углом +45°, поэтому она дает более сильный отклик именно на таких линиях. На рис. 10.7(*д*) показаны положительные значения лапласиана на рис. 10.7(*б*). Поскольку в данном примере интерес представляют точки с наибольшим откликом, в качестве порога T берем значение максимума на рис. 10.7(*д*). На рис. 10.7(*е*) белым показаны точки, значения в которых удовлетворяют условию $g \geq T$, где g обозначает изображение на рис. 10.7(*д*). На рис. 10.7(*е*) видны изолированные точки, в которых применяемая маска также дает сильный отклик. В исходном изображении эти точки и их ближайшие соседи расположены таким образом, что отклик маски в данных отдельных точках также оказывается максимальным. Такие изолированные точки легко обнаруживаются с помощью маски, показанной на рис. 10.4(*а*), и затем удаляются. Их также можно устранить с помощью морфологических операций, рассмотренных в предыдущей главе. ■

10.2.4. Модели перепадов

Обнаружение перепадов является наиболее часто применяемым подходом к сегментации изображений на основе резких (локальных) изменений яркости. Сначала мы опишем некоторые модели перепадов яркости, а затем рассмотрим ряд методов обнаружения таких перепадов.

Модели перепадов классифицируются в соответствии с их профилем яркости. *Ступенчатый перепад* выражается в резком переходе между двумя различными уровнями яркости, который в идеальном случае происходит на расстоянии в 1 пиксель. На рис. 10.8(*а*) изображен ступенчатый перепад и горизонтальный профиль изменения яркости на таком перепаде. Ступенчатые перепады встречаются, например, на изображениях, формируемых компьютером в задачах моделирования объемных объектов и анимации. Такие четкие, *идеальные* перепады происходят на расстоянии в 1 пиксель, если не производится никакой дополнительной обработки (например сглаживания), чтобы они выглядели более «реалистично». Модель дискретного ступенчатого перепада часто используется при разработке алгоритмов. Так, например, рассматриваемый в разделе 10.2.6

а б в

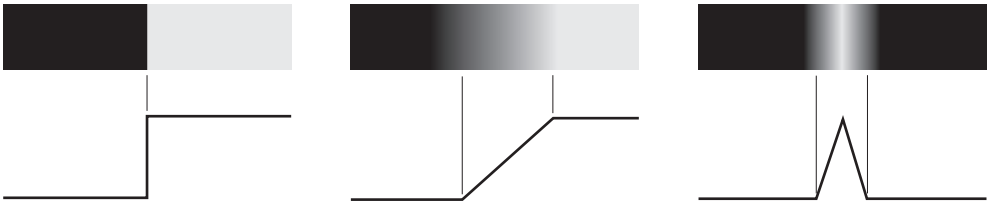


Рис. 10.8. Слева направо: модели идеального и наклонного перепадов, треугольного импульса и соответствующие им профили яркости

алгоритм выделения контуров Кэнни разработан с использованием ступенчатой модели перепада.

На практике перепады на цифровых изображениях размыты и зашумлены, причем степень размывания определяется главным образом ограниченными возможностями фокусировки (например объектива в случае оптической регистрации изображения), а уровень шума определяется в основном электронными компонентами системы формирования изображения. В таких случаях перепады яркости более точно моделируются *наклонным* профилем, подобным показанному на рис. 10.8(*б*). Крутизна наклонного участка обратно пропорциональна степени размывания перепада. В такой модели уже больше нет траектории тонкой контурной линии (шириной в 1 пиксель). Вместо этого контурной точкой теперь может считаться любая точка, лежащая на наклонном участке профиля, а сам контурный перепад представляет собой связное множество, образованное всеми такими точками.

Третья модель перепада яркости, называемая *треугольным импульсом*, представлена на рис. 10.8(*в*). Треугольный импульс моделирует линию, проходящую через область с постоянной яркостью, причем ширина основания треугольника определяется толщиной и резкостью линии. В пределе, когда ширина основания становится равной одному пикселю, треугольный импульс превращается в не что иное, как линию шириной в 1 пиксель, проходящую через область на изображении. Треугольные импульсы возникают, например, при фотосъемке городского пейзажа, когда в кадр попадает провод на фоне неба и профиль яркости в направлении перпендикулярно проводу будет аналогичен графику

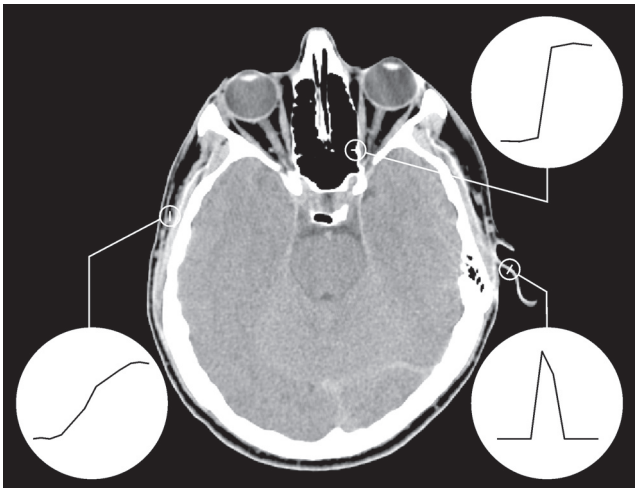


Рис. 10.9. Изображение размерами 1508×1970 пикселей с увеличенными фрагментами, демонстрирующими реальные профили наклонного (внизу слева), ступенчатого (вверху справа) перепадов и треугольного импульса (внизу справа). Диапазон профилей — от темного до светлого в областях, отмеченных соответственными кружками на изображении. Наклонный и «ступенчатый» перепады занимают 9 и 2 пикселей соответственно. Ширина основания треугольного импульса — 3 пикселя. (Исходное изображение предоставил д-р Дэвид Р. Пикенс, университет Вандербилта)

на рис. 10.8(*в*) с точностью до переворота. Как уже отмечалось, треугольные импульсы также часто появляются при оцифровке чертежей и на спутниковых изображениях, где тонкие детали, такие как дороги, естественно представляются такой моделью.

Обычно встречаются цифровые изображения, содержащие перепады всех трех видов сразу. Хотя расфокусировка и шум приводят к отклонениям от идеальной формы, на достаточно резких изображениях с умеренным шумом перепады действительно *напоминают* по своим характеристикам модели на рис. 10.8, как иллюстрируют профили яркости, представленные на рис. 10.9.[5] Что позволяют сделать показанные на рис. 10.8 модели — это представить форму перепада аналитически при построении алгоритмов обработки изображений, а качество работы этих алгоритмов будет зависеть от того, насколько фактические перепады яркости на изображении отличаются от моделей, использованных при разработке алгоритмов.

На рис. 10.10(*а*) приведено изображение, участок которого крупным планом был показан на рис. 10.8(*б*). На рис. 10.10(*б*) приведен горизонтальный профиль перепада яркости. На этом рисунке также показаны первая и вторая производные такого профиля яркости. Как уже говорилось в разделе 10.2.1, при движении вдоль профиля слева направо первая производная имеет разрыв в начале и конце наклонного участка, постоянное положительное значение на протяжении склона, и равна нулю в областях постоянства яркости. Вторая производная положительна в точке перехода от темного участка к наклонному, отрицательна в точке перехода от наклонного участка к светлому и равна нулю на линейном склоне и участках постоянной яркости. В случае обратного перепада яркости (от светлого к темному) знаки производных на рис. 10.10(*б*) изменятся на противоположные. Прямая линия, соединяющая максимальные положительное и отрицательное значения второй производной вблизи перепада, пересекает нулевой уровень в точке, называемой точкой *пересечения нулевого уровня*[6] второй производной.

Из проведенного рассмотрения можно заключить, что значение первой производной может использоваться для обнаружения наличия перепада яркости в данной точке изображения. Аналогично знак второй производной позволяет определить, лежит ли пиксель, находящийся на перепаде, на темной или светлой его части. Обратим внимание на два дополнительных свойства второй производной вблизи перепада яркости: (1) она дает два ненулевых (положительное и отрицательное) значения для каждого перепада, что является нежелательным свойством; и (2) ее точка пересечения нулевого уровня может использоваться для локализации середины широких перепадов, как мы увидим позже в этом разделе. В некоторых моделях перепада яркости используется наклонный участок с плавными переходами в начале и в конце (задача 10.7). Тем не менее сделанные выше выводы остаются справедливыми и для такого случая, а рассмотрение идеального наклонного перепада упрощает теоретические формулировки. Наконец, хотя до сих пор наше рассмотрение ограничивалось горизон-

---

[5] Крутые наклонные перепады на протяжении нескольких пикселей часто рассматривают как ступенчатые, чтобы отличать их от имеющихся на том же изображении наклонных перепадов с более плавным изменением яркости.

[6] В оригинале — *zero-crossing*. — *Прим. перев.*

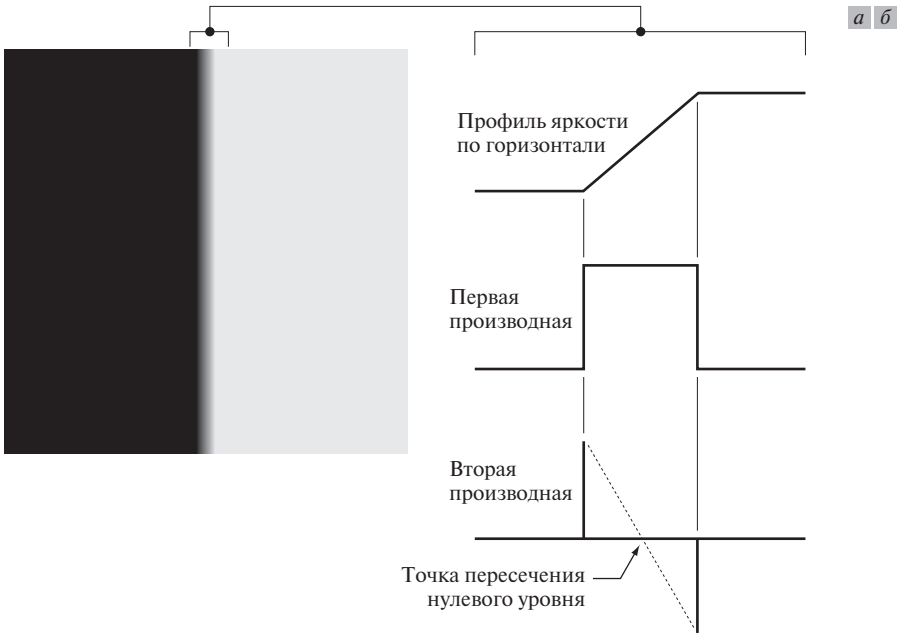


Рис. 10.10. (*а*) Две области постоянной яркости, разделенные идеальным наклонным перепадом, расположенным вертикально. (*б*) Горизонтальный профиль яркости вблизи перепада, а также графики первой и второй производных

тальным (одномерным) профилем яркости, те же соображения действуют и для перепадов, имеющих любое другое направление на изображении. Для этого достаточно рассмотреть профиль вдоль перпендикуляра к направлению перепада яркости в интересующей точке, после чего результаты интерпретируются так же, как описывалось выше.

Пример 10.4. Поведение первой и второй производных на перепаде яркости с шумом.

■ На перепадах яркости, показанных на рис. 10.8, шум отсутствует. Приведенные в первом столбце на рис. 10.11 увеличенные фрагменты изображений представляют собой четыре варианта перепада яркости между черной областью слева и белой областью справа (важно помнить, что весь такой переход от черного к белому представляет собой один перепад). Верхнее изображение не содержит шума. Оставшиеся три изображения этого столбца искажены аддитивным гауссовым шумом с нулевым средним и стандартными отклонениями 0,1, 1,0 и 10,0 градаций яркости соответственно. График под каждым изображением показывает горизонтальный профиль яркости в центре изображения. Все изображения имеют яркостное разрешение 8 бит, и значения 0 и 255 представляют соответственно черное и белое.

Рассмотрим верхнее изображение второго столбца. Как уже говорилось в связи с рис. 10.10(*б*), производная равна нулю в областях постоянной яркости. На изображении первой производной им отвечают две области черного цвета. На наклонном участке первая производная есть константа, равная его крутизне.

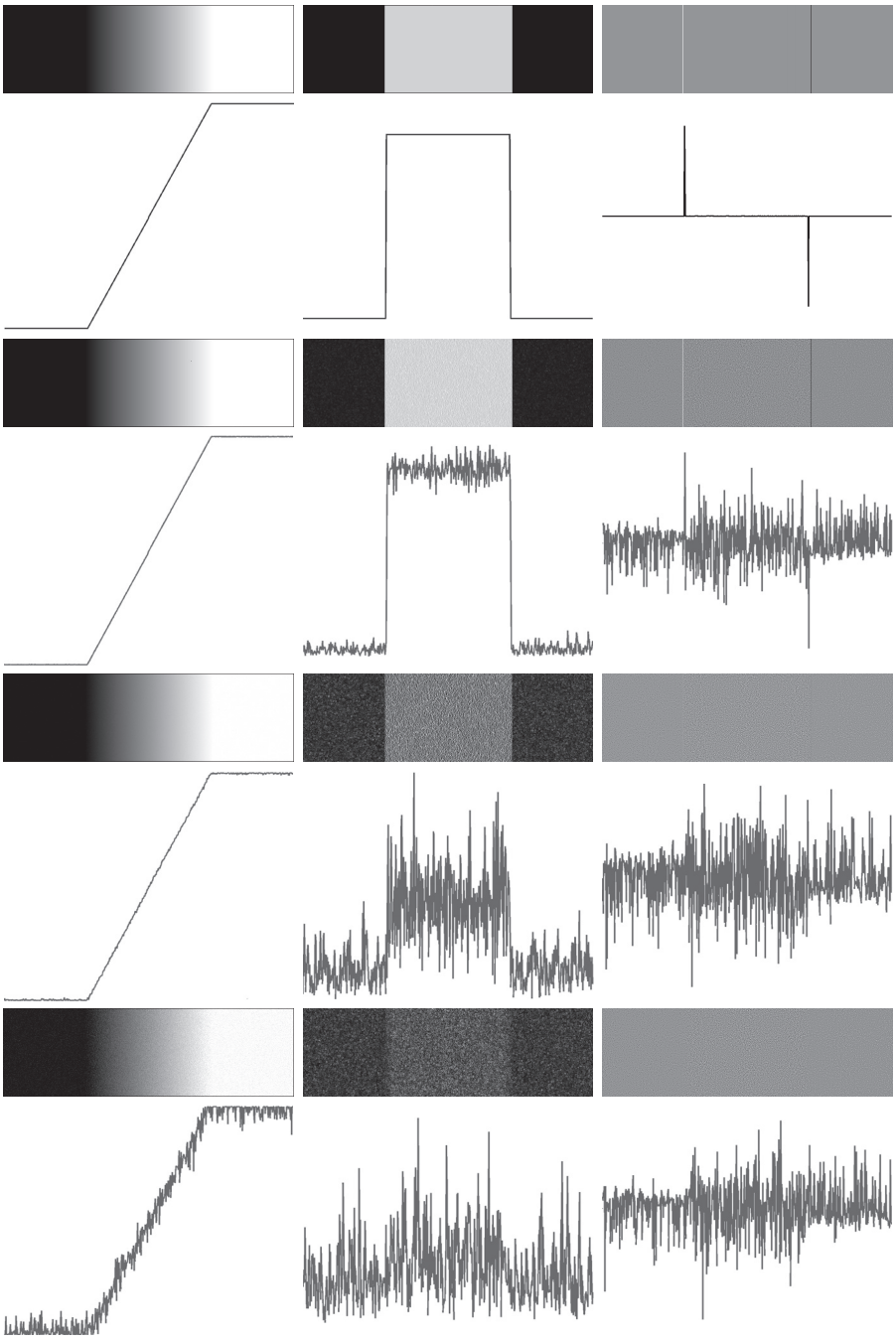


Рис. 10.11. Первый столбец: изображения и профили яркости наклонного перепада, искаженного гауссовым шумом со средним 0 и $\sigma = 0$; 0,1; 1,0; и 10,0 градаций яркости соответственно. Второй столбец: изображения первой производной и их профили яркости. Третий столбец: изображения второй производной и их профили яркости

Эта область на изображении производной имеет серый цвет. По мере движения вниз по второму столбцу рисунка производные все больше отличаются от случая без шума. В самом деле, последний профиль этого столбца вообще трудно соотнести с первой производной линейного перепада[7]. Этот результат особенно интересен тем, что на исходных изображениях в левом столбце шум почти незаметен. Данные примеры являются хорошей иллюстрацией чувствительности производных к присутствию шума.

> Детали вычисления первой и второй производных для целого изображения рассматриваются в следующем разделе. Здесь нас интересуют только их профили яркости.

Как этого можно было ожидать, вторая производная оказывается еще чувствительнее к шуму. Изображение второй производной для случая без шума приведено в правом верхнем углу. Тонкие белая и черная вертикальные линии — это положительная и отрицательная составляющие второй производной, которые объяснялись при обсуждении рис. 10.10. Средний уровень яркости изображений вторых производных выбран так, что серый фон соответствует нулевому значению, как уже говорилось выше. Единственное из зашумленных изображений вторых производных, которое еще как-то напоминает случай без шума, — это при уровне шума со стандартным отклонением 0,1 градации яркости. Остальные изображения вторых производных и их профили ясно иллюстрируют, что и в самом деле трудно найти те два импульса (положительный и отрицательный), которые, согласно свойствам второй производной, являются истинными признаками начала и конца перепада.

Таким образом, важно помнить, что даже небольшой шум может оказывать значительное воздействие на первую и вторую производные, применяемые для обнаружения перепадов на изображениях. В частности, в практических задачах, где возможно появление заметного шума, целесообразно рассмотреть вопрос о сглаживании изображения перед вычислением производных. ■

В заключение данного раздела заметим, что при обнаружении контурных перепадов выполняются три фундаментальных шага.

1. *Сглаживание изображения с целью уменьшения шума*. Необходимость этого шага отчетливо демонстрируют результаты во втором и третьем столбцах на рис. 10.11.
2. *Обнаружение точек перепада*. Как отмечалось выше, это локальная операция, которая выделяет на изображении все точки, которые могут быть точками перепада.
3. *Локализация перепада*. Цель этого шага состоит в том, чтобы выбрать среди выделенных кандидатов лишь те точки, которые действительно входят в множество точек, вместе составляющих контурный перепад.

---

[7] Приведенные во втором и третьем столбцах изображения (а значит, и графики) были, очевидно, подвергнуты операции градационной коррекции, описанной в разделе 2.6.3. Поэтому видна такая разница в средних контрастах верхнего и нижнего изображений во втором столбце, а также исчезли темная и светлая полоски в третьем столбце. — *Прим. ред. перев.*

В оставшейся части данного раздела рассматриваются методы решения указанных задач.

10.2.5. Простые методы обнаружения контурных перепадов

Как было продемонстрировано в предыдущем разделе, обнаруживать изменения яркости с целью выделения перепадов можно с использованием первых или вторых производных изображения. В данном разделе мы рассмотрим производные первого порядка, а вторые производные будут использоваться в разделе 10.2.6.

Градиент изображения и его свойства

Для нахождения в точке (x, y) изображения f одновременно величины перепада яркости и его направления обычно применяют градиент изображения $\nabla\mathbf{f}$, определяемый как *вектор*

$$\nabla\mathbf{f} \equiv \text{grad}(f) \equiv \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}. \tag{10.2-9}$$

Для удобства мы повторяем здесь ряд формул из раздела 3.6.4.

Вектор градиента имеет важное геометрическое свойство: его направление совпадает с направлением максимальной скорости изменения функции f в точке (x, y).

Модуль (*длина*) вектора $\nabla\mathbf{f}$ обозначается $M(x, y)$ и равен

$$M(x, y) = \text{mag}(\nabla f) = \sqrt{g_x^2 + g_y^2}\,. \tag{10.2-10}$$

Эта величина равна *значению* скорости изменения функции f в направлении вектора градиента. Заметим, что g_x, g_y и $M(x, y)$ представляют собой изображения тех же размеров, что и исходное изображение f; они формируются, когда координаты x и y пробегают все пиксели f. Изображение $M(x, y)$ также часто (хотя и не вполне правильно) называют *изображением градиента* или просто *градиентом*. Мы будем следовать этой укоренившейся практике и использовать этот термин для обеих целей, делая различия между вектором и его модулем только в тех случаях, когда возможна путаница. Операции сложения, возведения в квадрат и извлечения квадратного корня являются *поэлементными операциями*, в соответствии с определением в разделе 2.6.1.

Направление вектора градиента задается углом

$$\alpha(x, y) = \text{arctg}\left(\frac{g_y}{g_x}\right) \tag{10.2-11}$$

между направлением вектора $\nabla\mathbf{f}$ в точке (x, y) и осью x. Как и изображение градиента, $\alpha(x, y)$ также является изображением тех же размеров, что и исходное f, и формируется (поэлементной) операцией арктангенса от (поэлементного) деления изображения g_y на изображение g_x, Направление контура в произ-

вольной точке (x, y) *перпендикулярно* направлению $\alpha(x, y)$ вектора градиента в этой точке.

Пример 10.5. Свойства градиента.

■ На рис. 10.12(*а*) представлен увеличенный фрагмент изображения с прямолинейным участком перепада, где каждый квадратик представляет один пиксель изображения. Задача — получить величину и направление перепада яркости в точке, которая отмечена рамкой. Значения темных пикселей равны 0, а белых — 1. Как будет показано чуть позже, метод вычисления частных производных по направлениям *x* и *y* при помощи окрестности 3×3 с центром в интересующей точке состоит в следующем. Из суммы значений пикселей в нижней строке окрестности вычитается сумма значений пикселей в верхней строке окрестности, что дает частную производную по направлению *x*. Аналогично, вычитая сумму значений в левом столбце из суммы значений в правом столбце, получаем значение частной производной в направлении *y*. Таким способом для интересующей точки получим оценки частных производных $\partial f/\partial x = -2$ и $\partial f/\partial y = 2$, т. е.

> Напомним из раздела 2.4.2, что начало координат изображения находится в верхнем левом углу, ось *x* направлена вниз, а ось *y* — вправо.

$$\nabla \mathbf{f} = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} -2 \\ 2 \end{bmatrix},$$

откуда получаем в этой точке модуль градиента $M(x, y) = 2\sqrt{2}$. Аналогично направление вектора градиента в этой точке согласно (10.2-11) будет $\alpha(x, y) = \operatorname{arctg}(g_y / g_x) = -45°$ или 135° в положительном направлении относительно оси *x*. На рис. 10.12(*б*) показан вектор градиента и отложен угол его направления.

Рис. 10.12(*в*) иллюстрирует важный факт, уже отмечавшийся выше: направление контурного перепада в точке перпендикулярно направлению вектора градиента в этой точке. Таким образом, в данном примере угол направления перепада относительно оси *x* равен $\alpha - 90° = 45°$. Во всех точках перепада

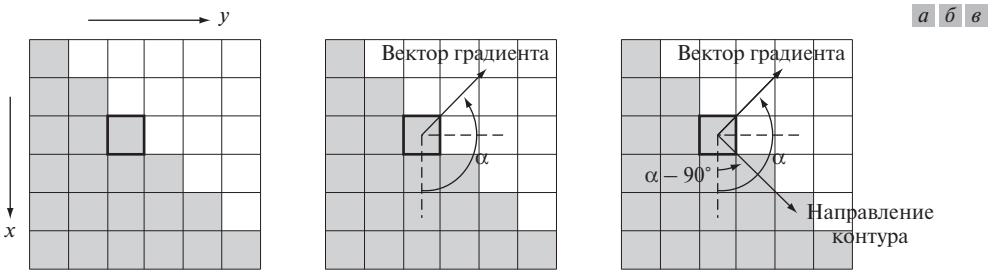


Рис. 10.12. Применение градиента для определения величины и направления яркостного перепада в заданной точке. Обратите внимание, что направление контура перпендикулярно направлению вектора градиента в этой точке. Каждый квадратик представляет один пиксель изображения

на рис. 10.12(*а*) градиент одинаковый, поэтому весь участок перепада имеет одно направление. Вектор градиента иногда называют нормалью к контурному перепаду. Если нормализовать этот вектор до единичной длины путем деления на модуль градиента (10.2-10), полученный вектор принято называть единичной нормалью к контуру. ■

Операторы градиента

Вычисление градиента изображения состоит в получении величин частных производных $\partial f/\partial x$ и $\partial f/\partial y$ для каждой точки. Поскольку мы имеем дело с дискретными величинами, требуется найти дискретную аппроксимацию частных производных в окрестности данной точки. Как уже известно из раздела 10.2.1,

$$g_x = \frac{\partial f(x,y)}{\partial x} = f(x+1,y) - f(x,y) \tag{10.2-12}$$

и

$$g_y = \frac{\partial f(x,y)}{\partial x} = f(x,y+1) - f(x,y). \tag{10.2-13}$$

Вычисления по этим двум уравнениям для всех нужных значений x и y можно реализовать путем фильтрации изображения $f(x,y)$ с помощью одномерных масок, показанных на рис. 10.13.

Для обнаружения перепадов, идущих в диагональных направлениях, требуется применять двумерные маски. *Перекрестный градиентный оператор Робертса* [Roberts, 1965] является одной из первых попыток применения двумерных масок с предпочитаемым диагональным направлением. Рассмотрим показанную на рис. 10.14(*а*) окрестность 3×3 некоторого элемента изображения. Оператор Робертса вычисляет разности по диагонали:

$$g_x = \frac{\partial f}{\partial x} = (z_9 - z_5) \tag{10.2-14}$$

В оставшейся части данного раздела мы неявно предполагаем, что f является функцией двух переменных, опуская эти переменные для простоты обозначений.

и

$$g_y = \frac{\partial f}{\partial y} = (z_8 - z_6). \tag{10.2-15}$$

Эти производные могут быть реализованы путем фильтрации всего изображения с помощью масок, представленных на рис. 10.14(*б*) и (*в*).

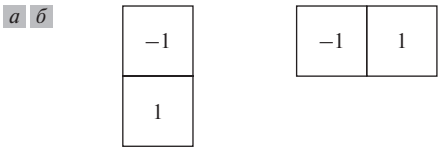


Рис. 10.13. Одномерные маски, применяемые для вычисления градиента согласно (10.2-12) и (10.2-13)

Маски размерами 2×2 концептуально просты, но не столь удобны для вычисления направлений перепадов, как маски, симметричные относительно центрального элемента, минимальные размеры которых 3×3. Такие маски учитывают значения с обеих сторон от центральной точки и поэтому дают больше информации о направлении перепада яркости. Простейшее дискретное приближение частных производных в случае использования масок 3×3 задается выражениями

$$g_x = \frac{\partial f}{\partial x} = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3) \qquad (10.2\text{-}16)$$

Хотя эти уравнения охватывают окрестность большего размера, в них по-прежнему участвуют разности значений яркости, поэтому все еще справедливы выводы, сделанные при первоначальном рассмотрении производных первого порядка.

и

$$g_y = \frac{\partial f}{\partial y} = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7). \qquad (10.2\text{-}17)$$

В этих формулах разность между суммами по нижней и верхней строкам окрестности 3×3 является приближенным значением производной по оси *x*, а разность между суммами по правому и левому столбцам этой окрестности — производной по оси *y*. Интуитивно можно ожидать, что такие приближения будут точнее тех, которые дает оператор Робертса. Для реализации формул (10.2-16) и (10.2-17) используется оператор, описываемый масками на рис. 10.14(*г*) и (*д*), который называется *оператором Превитта* [Prewitt, 1970].

Маски фильтров, применяемые для вычисления частных производных при нахождении градиента, часто называют *градиентными операторами*, *разностными операторами*, *операторами выделения контуров* или *детекторами контуров*.

Небольшое видоизменение последних двух формул состоит в использовании весового коэффициента 2 для средних элементов:

$$g_x = \frac{\partial f}{\partial x} = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3) \qquad (10.2\text{-}18)$$

и

$$g_y = \frac{\partial f}{\partial y} = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7). \qquad (10.2\text{-}19)$$

Это увеличенное значение используется для уменьшения эффекта сглаживания за счет придания большего веса средним точкам (задача 10.10). Для реализации двух последних выражений используются маски на рис. 10.14(*е*) и (*ж*), отвечающие *оператору Собела* [Sobel, 1970].

Маски оператора Превитта проще реализовать, чем маски оператора Собела, однако у последнего оператора влияние шума угловых элементов несколько меньше, что существенно при работе с производными. Отметим, что у каждой из масок на рис. 10.14 сумма коэффициентов равна нулю, т. е. эти операторы бу-

а
б *в*
г *д*
е *ж*

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

−1	0	0	−1
0	1	1	0

Маски оператора Робертса

−1	−1	−1	−1	0	1
0	0	0	−1	0	1
1	1	1	−1	0	1

Маски оператора Превитта

−1	−2	−1	−1	0	1
0	0	0	−2	0	2
1	2	1	−1	0	1

Маски оператора Собела

Рис. 10.14. Окрестность 3×3 внутри изображения (переменные z_i суть значения яркости) и различные маски, применяемые для вычисления градиента в центральной точке окрестности

дут давать нулевой отклик на областях постоянной яркости, как и следовало ожидать от дифференциального оператора.

Рассмотренные выше маски применяются для получения составляющих градиента g_x и g_y в каждом пикселе изображения, после чего эти частные производные используются для оценки величины перепада и направления контура. Для вычисления модуля градиента эти составляющие необходимо использовать совместно согласно (10.2-10). Однако такая реализация требует вычисления квадратов и квадратных корней. Часто используется подход, при котором величина градиента вычисляется приближенно через абсолютные значения частных производных:

$$M(x,y) \approx |g_x| + |g_y|. \qquad (10.2\text{-}20)$$

С точки зрения сложности вычислений такое выражение выглядит намного привлекательнее и по-прежнему несет информацию об изменениях яркости[8]. Как уже говорилось в разделе 3.7.3, это достигается ценой потери изотропно-

[8] Для этих же целей можно воспользоваться чуть более сложной, но и более точной следующей формулой приближения: $(a^2 + b^2)^{1/2} \approx \max(|a|,|b|) + \min(|a|,|b|)/2$. — *Прим. перев.*

0	1	1	–1	–1	0
–1	0	1	–1	0	1
–1	–1	0	0	1	1

а б в г

Маски оператора Превитта

0	1	2	–2	–1	0
–1	0	1	–1	0	1
–2	–1	0	0	1	2

Маски оператора Собела

Рис. 10.15. Маски операторов Превитта и Собела для обнаружения диагональных контуров

сти (т. е. инвариантности в отношении поворота) получаемых фильтров. Однако если для вычисления частных производных g_x и g_y применяются маски типа Превитта и Собела, то вопрос изотропности не возникает, так как эти маски инвариантны лишь для поворотов на углы, кратные 90°. В случае вертикальных и горизонтальных контуров вычисления градиента согласно выражениям (10.2-10) и (10.2-20) будут давать одинаковые результаты (задача 10.8).

Можно изменить приведенные на рис. 10.14 маски 3×3 таким образом, чтобы они давали максимальный отклик для контуров, направленных диагонально. Эти дополнительные пары масок операторов Превитта и Собела, предназначенных для обнаружения разрывов в диагональных направлениях, показаны на рис. 10.15.

Пример 10.6. Иллюстрация модуля и направления двумерного градиента.

■ На рис. 10.16 демонстрируются отклики двух составляющих градиента, $|g_x|$ и $|g_y|$, а также градиентное изображение, формируемое путем суммирования этих составляющих. На рис. 10.16(*б*) и (*в*) хорошо заметна направленность этих двух составляющих; в частности обратите внимание, насколько сильно выделяются элементы кровли, горизонтальные швы кладки и горизонтальные отрезки окон на рис. 10.16(*б*). Напротив, на рис. 10.16(*в*) выделены вертикальные составляющие, например вертикальные элементы фасада и окон. Говоря об изображениях, в которых главной является информация о перепадах яркости (как например в изображении модуля градиента), обычно используют термин «*карта перепадов*». Диапазон яркостей изображения на рис. 10.16(*а*) был преобразован к интервалу [0, 1], чтобы упростить выбор параметров в различных методах обнаружения перепадов, которые рассматриваются в данном разделе.

На рис. 10.17 приведено изображение угла (направления) градиента, вычисленное по формуле (10.2-11). Вообще говоря, изображения угла не так полезны для обнаружения контурных перепадов, как изображения модуля градиента,

а б
в г

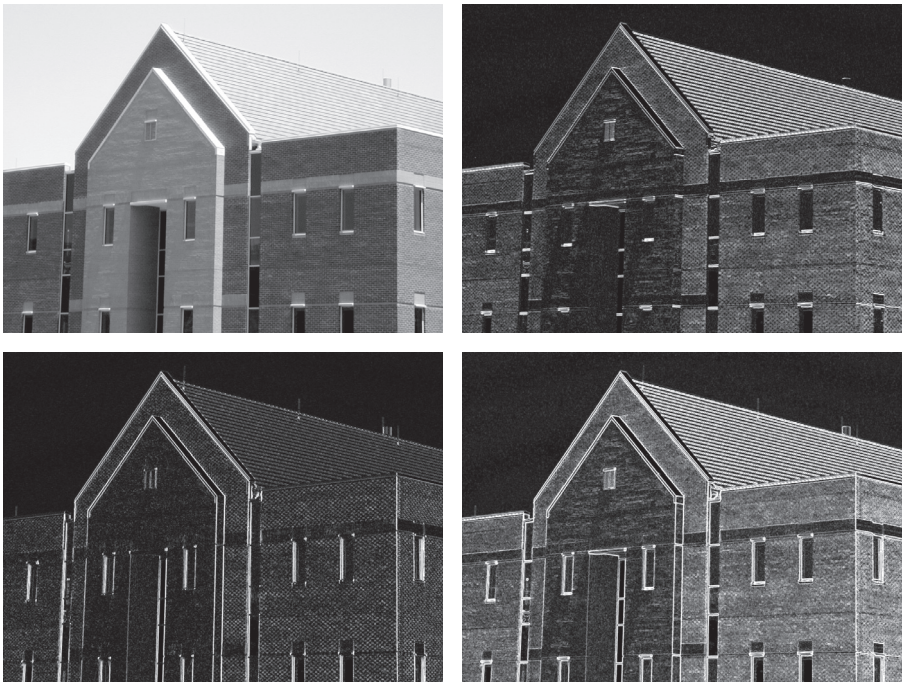


Рис. 10.16. (*а*) Исходное изображение с размерами 834×1114 пикселей, значения яркости которого преобразованы к интервалу [0, 1]. (*б*) Составляющая градиента вдоль оси x, $|g_x|$, полученная фильтрацией изображения по маске Собела на рис. 10.14(*е*). (*в*) Составляющая градиента вдоль оси y, $|g_y|$, полученная фильтрацией изображения по маске Собела на рис. 10.14(*ж*). (*г*) Изображение градиента $|g_x| + |g_y|$

однако они являются источником дополнительной информации. Например, области постоянной яркости на рис. 10.16(*а*), такие как передняя кромка ската крыши и горизонтальные полосы по верху фасадной стены, имеют постоянное значение на рис. 10.17, указывая тем самым, что во всех пикселях этих областей направление вектора градиента одно и то же.

Как будет показано в разделе 10.2.6, информация об угле играет важную вспомогательную роль при реализации алгоритма обнаружения контуров Кэнни — наиболее совершенного метода из рассматриваемых в данной главе. ■

Исходное изображение на рис. 10.16(*а*) (размерами 834×1114 пикселей) имеет сравнительно высокое разрешение, поэтому при выбранном масштабе съемки кирпичная кладка стен вносит существенный вклад в текстуру (детальность) изображения. Для выделения контуров такая степень детализации часто нежелательна, поскольку выступает как шум, который дополнительно усиливается при вычислении производных и тем самым затрудняет обнаружение основных контуров. Одним из способов снижения детализации является сглаживание изображения. На рис. 10.18 показана та же последовательность изображений, что и на рис. 10.16, но с предварительным сглаживанием исходного изображения усредняющим фильтром с окном 5×5 (сглаживающие фильтры обсужда-

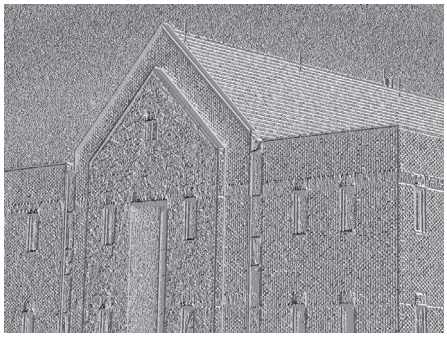


Рис. 10.17. Изображение угла градиента, вычисленное по формуле (10.2-11). Области постоянной яркости на этом изображении показывают, что направление вектора градиента одинаково для всех пикселей этих областей

лись в разделе 3.5). Как видно, сигнал, вносимый кирпичной кладкой стен в отклики масок, существенно ослаблен, и в результате основные контуры здания стали намного более заметными.

Максимальная величина перепада на сглаженном изображении снижается обратно пропорционально размеру маски сглаживания (задача 10.13).

На рис. 10.16 и 10.18 хорошо видно, что горизонтальная и вертикальная маски Собела дают примерно одинаковый отклик на контурах, имеющих направления ±45°. Если важно подчеркнуть контуры диагональных направлений, то следует воспользоваться одной из масок на рис. 10.15. Абсолютные величины откликов диагональных масок Собела для углов 45° и −45° показаны на рис. 10.19(*а*) и (*б*) соответственно, где хорошо виден более сильный отклик на перепадах, ориентированных вдоль диагональных направлений. Обе диагональные маски дают приблизительно одинаковые отклики на горизонтальных и вертикальных контурах, которые, как и можно было ожидать, слабее, чем отклики от горизонтальной и вертикальной масок, рассмотренных выше.

Градиент в сочетании с пороговой обработкой

Результаты на рис. 10.18 демонстрируют, что избирательность детектора контуров можно повысить путем сглаживания изображения перед вычислением градиента. Другой метод, направленный на достижение той же цели, состоит в пороговой обработке градиентного изображения. Например, на рис. 10.20(*а*) представлен вид градиентного изображения, ранее показанного на рис. 10.16(*г*), после следующей пороговой операции: если яркость пикселя больше или равна 33 % от максимального значения градиентного изображения, он отмечается белым цветом, в противном случае — черным. Сравнивая полученное изображение с рис. 10.18(*г*), мы видим, что после пороговой обработки осталось меньше контурных участков, а сами они стали более резкими (обратите внимание, например, на контуры черепицы). С другой стороны, на многих контурах появились разрывы, как, например, на линии среза дальнего края крыши, расположенного под углом 45°.

а б
в г

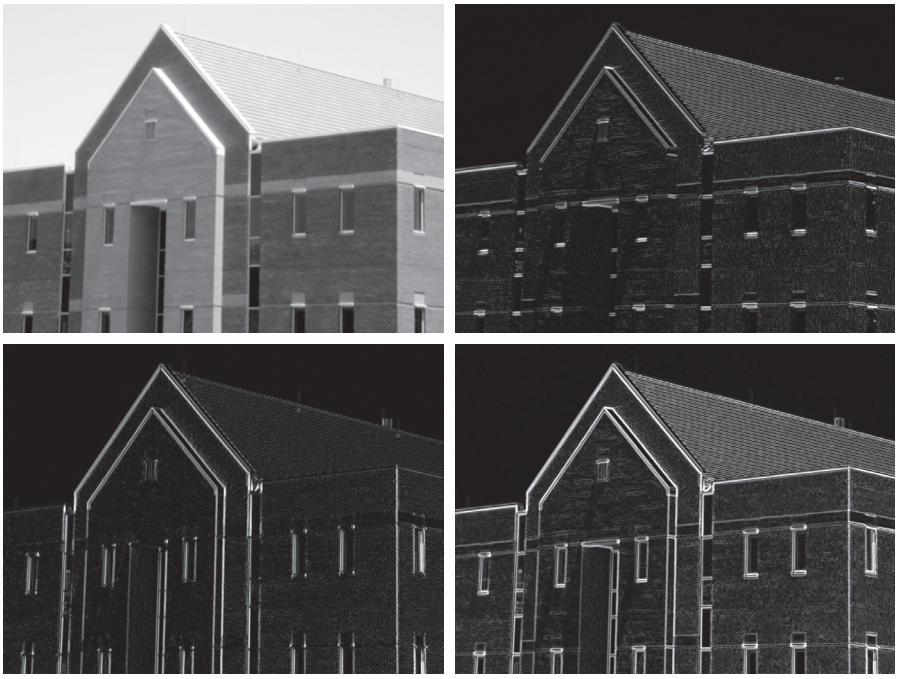


Рис. 10.18. Та же последовательность, что и на рис. 10.16, но исходное изображение предварительно сглажено с помощью усредняющего фильтра 5×5

Для получения изображения на рис. 10.20(*а*) порог выбирался так, чтобы пропало большинство мелких контуров от кирпичей. Вспомним, что такой же была изначальная цель сглаживания изображения на рис. 10.16 перед вычислением градиента.

Если интерес одновременно представляет как выделение основных контуров, так и сохранение как можно большей их связности, обычная практика состоит в совместном применении сглаживания с пороговым преобразованием.

а б

Рис. 10.19. Обнаружение диагональных контуров. (*а*) Результат использования маски на рис. 10.15(*в*). (*б*) Результат использования маски на рис. 10.15(*г*). Исходным для обоих случаев являлось изображение на рис. 10.18(*а*)

а б

Рис. 10.20. (*а*) Результат обработки изображения на рис. 10.16(*г*) с порогом величиной 33 % от максимального значения; этого достаточно для устранения большинства контуров от кирпичной кладки. (*б*) Результат обработки изображения на рис. 10.18(*г*) с порогом величиной 33 % от максимального значения

На рис. 10.20(*б*) показан результат пороговой обработки рис. 10.18(*г*), представляющего градиент предварительно сглаженного изображения. Из сравнения этого результата с рис. 10.20(*а*) видно, что число разрывов на контурах уменьшается, например на наклонных линиях. Конечно, те контуры, яркость которых значительно уменьшилась вследствие сглаживания, после пороговой обработки обычно полностью пропадают (как например края черепицы на крыше). Мы вернемся к проблеме разрывов на контурах в разделе 10.2.7.

10.2.6. Более совершенные методы обнаружения контуров

Рассмотренные в предыдущем разделе методы выделения контуров использовали просто фильтрацию изображения с помощью одной или более масок без каких-либо предположений касательно характеристик контура и содержания шума. В данном разделе мы обсудим более совершенные методы, в которых делается попытка улучшения простых методов обнаружения контуров благодаря учету таких факторов, как шум в изображении и особенности самих контуров.

Детектор контуров Марра — Хилдрета

Одна из первых успешных попыток внедрения более сложного анализа в процесс нахождения контуров принадлежит Марру и Хилдрету [Marr, Hildreth, 1980]. Применяемые в то время методы выделения контуров основывались на применении операторов с малой апертурой (таких как маски Собела), обсуждавшихся в предыдущем разделе. Марр и Хилдрет выдвинули следующие аргументы: (1) изменения яркости зависят от масштаба изображения, и поэтому для их обнаружения необходимо использовать операторы различных размеров; и (2) резкое изменение яркости приводит к появлению пика или провала первой производной или, что то же самое, к пересечению нулевого уровня второй производной (как мы видели на рис. 10.10).

Чтобы убедиться в зависимости контуров от масштаба, рассмотрите, например, контур крыши на рис. 10.8(*в*). Если уменьшить изображение, этот контур покажется тоньше.

Эти идеи приводят к мысли, что используемый для обнаружения контуров оператор должен обладать двумя характерными свойствами. Первое и главное, это должен быть дифференциальный оператор, способный вычислять приближенное значение первой или второй производной в каждой точке изображения. Во-вторых, он должен допускать настройку на любой желаемый масштаб, чтобы операторы большого размера можно было использовать для обнаружения размытых перепадов яркости, а операторы малого размера — для мелких деталей с высокой резкостью.

Марр и Хилдрет пришли к выводу, что самым подходящим оператором, который удовлетворяет этим условиям, является фильтр $\nabla^2 G$, где ∇^2 — оператор Лапласа ($\partial^2/\partial x^2 + \partial^2/\partial y^2$), а G — двумерная гауссова функция

$$G(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{10.2-21}$$

Использование в уравнении (10.2-21) выражения, отличающегося от обычного определения двумерной гауссовой функции плотности вероятности отсутствием множителя $1/2\pi\sigma 2$, является обычной практикой. Если для конкретной прикладной задачи требуется точное выражение, мультипликативную константу можно добавить к окончательному результату (10.2-23).

со стандартным отклонением σ (которое иногда называют *пространственным параметром*). Чтобы найти выражение для $\nabla^2 G$, выполним следующие дифференцирования:

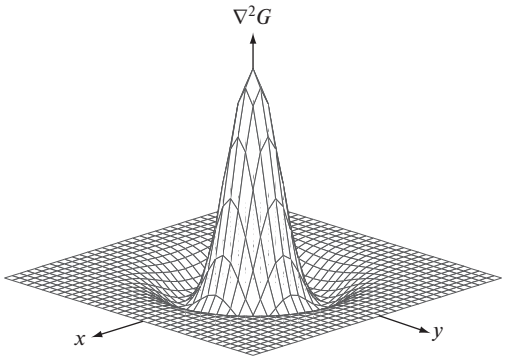
$$\begin{aligned}\nabla^2 G(x,y) &= \frac{\partial^2 G(x,y)}{\partial x^2} + \frac{\partial^2 G(x,y)}{\partial y^2} = \frac{\partial}{\partial x}\left(\frac{-x}{\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}\right) + \frac{\partial}{\partial y}\left(\frac{-y}{\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}\right) = \\ &= \left(\frac{x^2}{\sigma^4} - \frac{1}{\sigma^2}\right) e^{-\frac{x^2+y^2}{2\sigma^2}} + \left(\frac{y^2}{\sigma^4} - \frac{1}{\sigma^2}\right) e^{-\frac{x^2+y^2}{2\sigma^2}}.\end{aligned} \tag{10.2-22}$$

Приводя подобные члены, окончательно получаем:

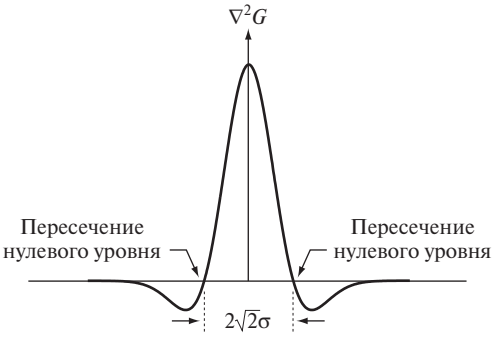
$$\nabla^2 G(x,y) = \left(\frac{x^2+y^2-2\sigma^2}{\sigma^4}\right) e^{-\frac{x^2+y^2}{2\sigma^2}}. \tag{10.2-23}$$

Эту формулу обычно называют *лапласианом гауссиана* (ЛГ).

На рис. 10.21(*а*)—(*в*) показаны трехмерный график, изображение и профиль ЛГ функции в перевернутом виде (т. е. с обратным знаком); обратите внимание, что ЛГ-функция пересекает нулевой уровень в точках, удовлетворяющих уравнению $x^2 + y^2 = 2\sigma^2$, что определяет окружность радиуса $\sqrt{2}\sigma$ с центром в начале координат. Из-за характерной формы графика на рис. 10.21(*а*) лапласиан гауссиана иногда называют оператором вида *мексиканская шляпа*. На рис. 10.21(*г*) приведена маска 5×5, используемая в качестве приближения функции на рис. 10.21(*а*) (на практике используют маску, имеющую значения с обратным знаком). Такое приближение не является единственно возможным; цель его — передать сущность *формы* ЛГ-функции, т. е. положительного всплеска в центре, окруженного областью отрицательных значений, которые, достигнув минимума, в дальнейшем по мере удаления от центра возрастают, приближаясь за пределами окрест-



а б
в г



0	0	−1	0	0
0	−1	−2	−1	0
−1	−2	16	−2	−1
0	−1	−2	−1	0
0	0	−1	0	0

Рис. 10.21. (*а*) Трехмерный график лапласиана гауссиана с *обратным знаком*. (*б*) Негативное изображение ЛГ-функции. (*в*) Поперечный профиль с отмеченными точками пересечения нулевого уровня. (*г*) Маска 5×5 для приближенного вычисления зависимости (*а*). На практике используются значения с обратным знаком

ности к нулевому значению. Сумма коэффициентов должна равняться нулю, чтобы маска давала нулевой отклик на области постоянной яркости.

> Обратите внимание на сходство профиля, представленного на рис. 10.21(*в*), с фильтром высоких частот на рис. 4.37(*г*). Тем самым, можно ожидать, что лапласиан гауссиана ведет себя подобно фильтру высоких частот.

Маски ЛГ можно строить любых желаемых размеров, выбирая шаг дискретизации уравнения (10.2-23) и масштабируя затем все коэффициенты так, чтобы их сумма равнялась нулю. Более эффективен следующий метод построения ЛГ-фильтра. Осуществляется дискретизация гауссовой функции (10.2-21) на сетке желаемых размеров $n \times n$, после чего вычисляется свертка[9] полученного массива с маской лапласиана, например представленной на рис. 10.4(*а*). Поскольку свертка изображения с маской, у которой сумма коэффициентов равна нулю,

[9] Лапласиан гауссиана является симметричным фильтром, поэтому пространственная фильтрация с помощью корреляции и свертки дает одинаковый результат. Мы здесь используем терминологию, относящуюся к свертке, чтобы подчеркнуть линейность фильтра и для большей согласованности с литературой по данной теме. Важно учитывать комментарии, сделанные на этот счет в конце раздела 3.4.2.

дает результат также с нулевой суммой элементов (см. задачи 3.16 и 10.14), это автоматически обеспечивает выполнение того требования, что сумма коэффициентов ЛГ-фильтра должна быть равна нулю. Вопрос выбора размера ЛГ-фильтра рассматривается позже в данном разделе.

Выбор оператора $\nabla^2 G$ основывается на двух фундаментальных идеях. Первая — то, что гауссова функция в лапласиане гауссиана обеспечивает сглаживание изображения, тем самым снижая интенсивность структур (в том числе шумовых) с размерами меньше σ. В отличие от сглаживания усреднением, которое рассматривалось в разделе 3.5 и использовалось на рис. 10.18, гауссова функция является гладкой не только в пространственной, но и в частотной областях (см. раздел 4.8.3), что снижает вероятность появления артефактов (типа «звона»), которых не было в исходном изображении. Вторая идея связана с ∇^2 — второй производной в составе фильтра. Хотя для обнаружения скачков яркости могут применяться первые производные, они являются направленными операторами. Напротив, лапласиан обладает тем важным достоинством, что он изотропен (инвариантен к повороту), что не только соответствует характеристикам зрительной системы человека ([Marr, 1982]), но и обеспечивает одинаковый отклик на изменения яркости в любом направлении, что позволяет избежать необходимости применения нескольких масок для вычисления максимального отклика в любой точке изображения.

Алгоритм Марра—Хилдрета состоит из свертки ЛГ-фильтра с исходным изображением

$$g(x,y) = \left[\nabla^2 G(x,y)\right] \star f(x,y) \tag{10.2-24}$$

и нахождения затем точек пересечения нулевого уровня функции $g(x,y)$, чтобы локализовать контуры на изображении $f(x,y)$. Поскольку взятие второй производной является линейной операцией, уравнение (10.2-24) можно переписать в виде

$$g(x,y) = \nabla^2 \left[G(x,y) \star f(x,y)\right], \tag{10.2-25}$$

как если бы изображение сначала сворачивалось с гауссовой сглаживающей функцией, а потом вычислялся лапласиан результата. Оба последних уравнения дают идентичные результаты.

> Выражение (10.2-25) вычисляется в пространственной области с помощью уравнения (3.4-2). Его также можно вычислять в частотной области с помощью (4.7-1).

Алгоритм выделения контуров Марра—Хилдрета можно кратко суммировать следующим образом:

1. Исходное изображение обрабатывается гауссовым фильтром низких частот с размерами $n \times n$, полученным посредством дискретизации уравнения (10.2-21).
2. Вычисляется лапласиан полученного изображения, используя, например, маску 3×3, приведенную на рис. 10.4(*a*). (Первый и второй шаги вместе реализуют уравнение (10.2-25).)
3. На полученном изображении находятся точки пересечения нулевого уровня.

Чтобы указать размеры гауссова фильтра, вспомним, что приблизительно 99,7 % объема под двумерной гауссовой поверхностью лежит в окрестности $\pm 3\sigma$ с центром в точке математического ожидания. Поэтому при выборе размеров $n \times n$ дискретного ЛГ-фильтра на практике рекомендуется в качестве n брать наименьшее нечетное целое число, равное или большее 6σ. Использование меньшего n приведет к «усечению» ЛГ-функции, причем степень такого усечения обратно пропорциональна размеру маски. Использование маски с бóльшими размерами не оказывает заметного влияния на результат.

Один из методов обнаружения того, что функция $g(x, y)$ пересекает нулевой уровень в некотором пикселе p, основан на использовании окрестности 3×3 с центром в p. Переход через ноль в точке p означает, что по меньшей мере два расположенных друг против друга соседних пикселя имеют значения разных *знаков*. Проверяются четыре пары соседей: верхний/нижний, левый/правый и две пары по диагонали. Если значения $g(x, y)$ сравниваются с некоторым порогом (что является обычным приемом), то требуется не только чтобы у противолежащих соседей p были разные знаки, но и чтобы абсолютная величина разности их значений превышала заданный порог; в таком случае p считается пикселем пересечения нулевого уровня. Этот метод иллюстрирует ниже пример 10.7.

> Попытки определять точки пересечения нулевого уровня как такие точки (x, y), в которых $g(x, y) = 0$, неприменимы для практических целей по причине шума и/или вычислительных погрешностей.

Нахождение точек пересечения нулевого уровня является отличительной чертой детектора контуров Марра—Хилдрета. Алгоритм из предыдущего абзаца привлекателен ввиду простоты его реализации и обычно хороших получаемых результатов. Если точность локализации точек перехода через ноль таким способом оказывается недостаточной для конкретной практической задачи, можно воспользоваться более сложным методом [Huertas, Medioni, 1986] для нахождения координат точек пересечения нулевого уровня с точностью выше одного пикселя.

Пример 10.7. Поиск контуров методом Марра—Хилдрета.

■ На рис. 10.22(*а*) приведено ранее использовавшееся исходное изображение здания, а на рис. 10.22(*б*) представлен результат шагов 1 и 2 алгоритма Марра—Хилдрета с $\sigma = 4$ (приблизительно 0,5 % наименьшего размера изображения) и $n = 25$ (наименьшее целое нечетное число, большее или равное 6σ, как говорилось выше). Как и раньше на рис. 10.5, это изображение показано после градационного преобразования. На рис. 10.22(*в*) представлены точки пересечения нулевого уровня, полученные, как описывалось выше, по окрестности 3×3 при нулевом пороге. Видно, что в состав контуров, найденных с помощью данного метода, входят многочисленные замкнутые петли. Этот так называемый «эффект спагетти» является серьезным недостатком данного метода в случае нулевого порога (задача 10.15). Во избежание этого эффекта воспользуемся положительным значением порога.

На рис. 10.22(*г*) показан результат, полученный при величине порога, приблизительно равной 4 % максимального значения ЛГ-изображения. Как видно

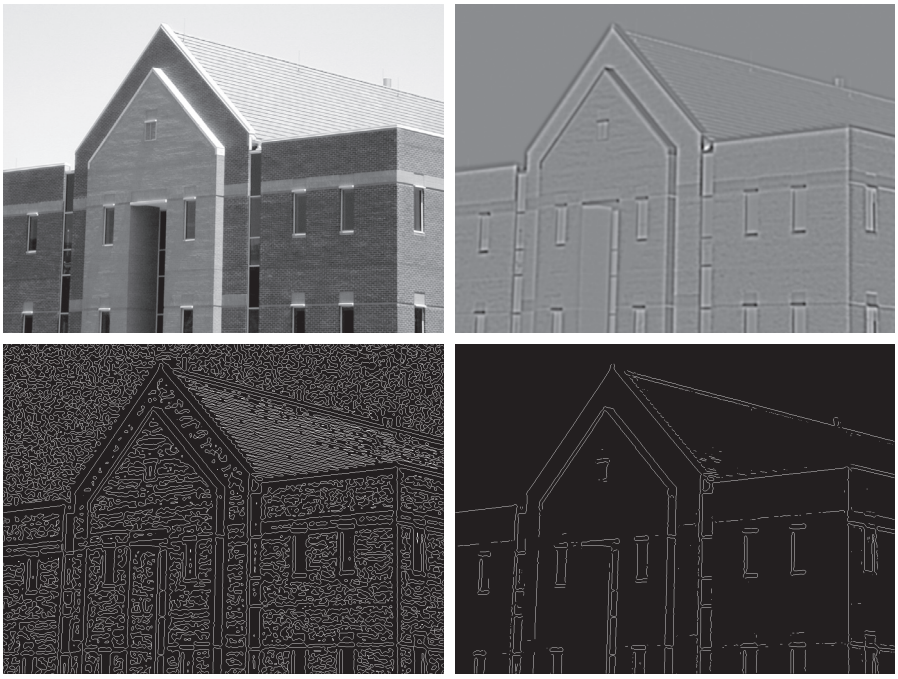


Рис. 10.22. (*а*) Исходное изображение с размерами 834×1114 пикселей, значения яркости которого преобразованы к интервалу [0, 1]. (*б*) Результаты шагов 1 и 2 алгоритма Марра—Хилдрета при $\sigma = 4$ и $n = 25$. (*в*) Пересечения нулевого уровня на изображении (*б*) при использовании порога 0 (обратите внимание на замкнутые петли контуров). (*г*) Пересечения нулевого уровня при использовании порога величиной 4% максимального значения изображения (*б*). Обратите внимание на малую толщину контуров

из рисунка, большинство основных контуров легко выделяются, а «несущественные» детали, такие как контуры кирпичной кладки и черепицы, удалены фильтрацией. Как будет показано в следующем разделе, подобного качества практически невозможно добиться, используя чисто градиентные методы выделения контуров из предыдущего раздела. Использование пересечений нулевого уровня для обнаружения контуров имеет и другое важное следствие: получаемые контуры имеют толщину 1 пиксель. Это свойство упрощает последующие стадии цифровой обработки, в частности связывание контуров. ■

Иногда применяют процедуру, которая учитывает отмечавшуюся выше зависимость скачков яркости от масштаба изображения и состоит в фильтрации исходного изображения с различными значениями σ. Затем полученные карты точек пересечения нулевого уровня комбинируются так, что остаются только контуры, присутствующие на всех картах. Такой подход способствует получению полезной информации, но из-за своей сложности применяется на практике главным образом для выбора подходящего значения σ, которое будет использовано в одиночном фильтре.

Марр и Хилдрет заметили [Marr, Hildreth, 1980], что ЛГ-фильтр в уравнении (10.2-23) можно аппроксимировать *разностью гауссианов* (РГ):

$$\Delta G(x,y) = \frac{1}{2\pi\sigma_1^2} e^{-\frac{x^2+y^2}{2\sigma_1^2}} - \frac{1}{2\pi\sigma_2^2} e^{-\frac{x^2+y^2}{2\sigma_2^2}}, \qquad (10.2\text{-}26)$$

Как отмечалось в разделе 4.7.4, разность гауссовых функций является фильтром высоких частот.

где $\sigma_1 > \sigma_2$. Данные экспериментов говорят о том, что определенные «каналы» в зрительной системе человека обладают избирательностью по ориентации и частоте, и их можно моделировать уравнением (10.2-26) с отношением стандартных отклонений 1,75:1. Марр и Хилдрет рекомендовали использовать отношение 1,6:1, которое соответствует основным характеристикам, полученным в экспериментах, и также обеспечивает более близкую «технически» аппроксимацию ЛГ-функции. Чтобы осмысленно сравнивать функции ЛГ и РГ, значение σ для лапласиана гауссиана необходимо выбирать в соответствии со следующим соотношением, чтобы точки пересечения нулевого уровня ЛГ и РГ были одинаковыми (задача 10.17):

$$\sigma^2 = \frac{\sigma_1^2\sigma_2^2}{\sigma_1^2 - \sigma_2^2} \ln\left(\frac{\sigma_1^2}{\sigma_2^2}\right). \qquad (10.2\text{-}27)$$

Хотя точки пересечения нулевого уровня функций ЛГ и РГ при таком значении σ совпадут, по амплитуде функции будут отличаться. Их можно совместить с помощью такого масштабирования, чтобы значения функций в начале координат были одинаковыми.

Профили на рис. 10.23(*а*) и (*б*) были построены при отношении стандартных отклонений 1:1,75 и 1:1,6 соответственно (по соглашению кривые показаны в перевернутом виде, как на рис. 10.21). Профили лапласиана гауссиана показаны сплошными линиями, а разности гауссианов — пунктирными. Профили яркости проходят через центр масок ЛГ и РГ, сформированных путем дискретизации уравнений (10.2-23) (с добавлением коэффициента $1/2\pi\sigma^2$) и (10.2-26) соответственно. Амплитуда всех кривых нормализована так, чтобы в начале координат достигалось значение 1. Как показывает рис. 10.23(*б*), при отношении 1:1,6 функция РГ точнее аппроксимирует лапласиан гауссиана.

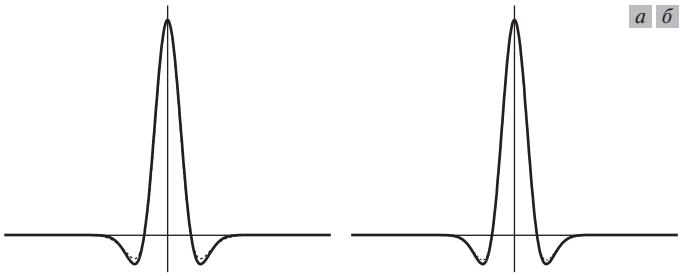


Рис. 10.23. (*а*) Показанные с противоположным знаком профили функций ЛГ (сплошная линия) и РГ (пунктир) в случае отношения стандартных отклонений 1,75:1. (*б*) Вид профилей в случае отношения 1,6:1

Оба фильтра, ЛГ и РГ, можно реализовать с помощью одномерной свертки, не выполняя прямо операцию вычисления двумерной свертки (задача 10.19). Если размеры изображения $M \times N$, а размеры фильтра $n \times n$, это позволяет уменьшить число умножений и сложений при вычислении каждой свертки с $O(n^2MN)$ в случае двумерной свертки до $O(nMN)$ при одномерных свертках, что существенно для реализации. Например, при $n = 25$ одномерная реализация требует в 12 раз меньше операций умножения и сложения по сравнению с двумерной сверткой.

Детектор контуров Кэнни

Хотя рассматриваемый в данном разделе детектор контуров Кэнни [Canny, 1986] использует более сложный алгоритм, качество его работы в общем случае выше, чем у рассмотренных до сих пор алгоритмов выделения контуров. Метод Кэнни преследует три основные цели:

1. *Низкая частота ошибок*. Должны обнаруживаться все контуры с минимумом ложных срабатываний. Иначе говоря, обнаруженный контурный рисунок должен быть как можно ближе к истинному.
2. *Хорошая локализация контурных точек*. Найденное пространственное положение контуров должно быть как можно ближе к истинному, т. е. расстояние от точки, которую детектор опознает как точку контура, до середины истинного контура должно быть минимальным.
3. *Одиночный отклик на точку контура*. Для каждой точки истинного контура детектор должен обнаруживать только одну точку. Другими словами, количество локальных максимумов в окрестности истинной контурной точки должно быть минимальным. Это значит, что детектор не должен обнаруживать несколько контурных пикселей, если имеется контур, состоящий только из одной точки.

Сущность работы Кэнни состояла в том, чтобы выразить три перечисленных критерия математически и затем попытаться найти оптимальное решение в такой постановке. Вообще говоря, трудно (или даже невозможно) найти в общем виде решение, удовлетворяющее всем поставленным целям. Однако применяя численную оптимизацию к одномерным ступенчатым перепадам с искажениями в форме аддитивного белого гауссова шума, можно прийти к заключению, что хорошим приближением[10] к оптимальному детектору ступенчатых перепадов является *первая производная гауссиана*:

$$\frac{d}{dx}e^{-\frac{x^2}{2\sigma^2}} = \frac{-x}{\sigma^2}e^{-\frac{x^2}{2\sigma^2}}. \qquad (10.2\text{-}28)$$

Напомним, что белый шум обладает непрерывным и равномерным частотным спектром в заданном диапазоне частот. Белый гауссов шум — это такой белый шум, значения которого имеют нормальное распределение. Белый гауссов шум является хорошим приближением для многих ситуаций реального мира и приводит к построению моделей, допускающих математическую трактовку. Его свойством является то, что принимаемые значения статистически независимы.

[10] Кэнни [Canny, 1986] показал, что использование гауссовой аппроксимации лишь незначительно уступает оптимальному решению с применением численных методов (около 20 %). Подобные отличия обычно неощутимы в большинстве практических задач.

Обобщение этого результата на двумерный случай использует тот факт, что в направлении нормали к контуру *по-прежнему применим* одномерный подход (см. рис. 10.12). Поскольку направление нормали заранее неизвестно, для этого потребовалось бы применять одномерный детектор контуров во всевозможных направлениях. Приближенное решение такой задачи можно получить, сначала сглаживая изображение с помощью *круговой* двумерной гауссовой функции и вычисляя градиент результата, а уже затем по модулю и направлению вектора градиента оценить величину и направление перепада в каждой точке.

Обозначим $f(x, y)$ исходное изображение, а $G(x, y)$ — гауссову функцию

$$G(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}}. \qquad (10.2\text{-}29)$$

Сглаженное изображение $f_s(x, y)$ формируется сверткой G с f:

$$f_s(x,y) = G(x,y) \bigstar f(x,y). \qquad (10.2\text{-}30)$$

Модуль и направление (угол) градиента вычисляются согласно описанию в разделе 10.2.5:

$$M(x,y) = \sqrt{g_x^2 + g_y^2} \qquad (10.2\text{-}31)$$

и

$$\alpha(x,y) = \operatorname{arctg}\left(\frac{g_y}{g_x}\right), \qquad (10.2\text{-}32)$$

где $g_x = \partial f_s / \partial x$ и $g_y = \partial f_s / \partial y$. Для вычисления g_x и g_y может использоваться любая пара масок фильтра из представленных на рис. 10.14. Формула (10.2-30) вычисляется с помощью гауссовой маски $n \times n$, размер которой обсуждается ниже. Напомним, что $M(x, y)$ и $\alpha(x, y)$ являются массивами тех же размеров, что и исходное изображение, по которому они вычисляются.

Поскольку изображение $M(x, y)$ формируется как модуль градиента, оно обычно содержит широкие гребни вдоль линии локальных максимумов (напомним проведенное в разделе 10.2.1 обсуждение характера контуров, выделяемых с помощью градиента). Следующий шаг состоит в утончении таких гребней. Один из применяемых методов использует *подавление немаксимальных точек*, которое может выполняться несколькими способами, но сущность метода в том, что выбирается несколько дискретных направлений нормали к контуру (т. е. направлений вектора градиента). Например, в области размерами 3×3 пикселя для контура, проходящего через центральный элемент области, можно определить четыре направления[11]: горизонтальное, вертикальное, +45° и –45°. На рис. 10.24(*а*) показаны две возможные ориентации горизонтального контура. Поскольку требуется привести все возможные направления контура к четырем фиксированным значениям, необходимо указать диапазон для каждого из выбранных направлений, например в каком из случаев контур будет считаться горизонтальным. Направление контура в точке определяется по направлению нормали к нему, которое определяется непосредственно из изображения, по-

[11] Следует учитывать, что контур каждого направления может иметь две ориентации. Например, контуры с направлениями нормали 0° и 180° — это один и тот же *горизонтальный* контур.

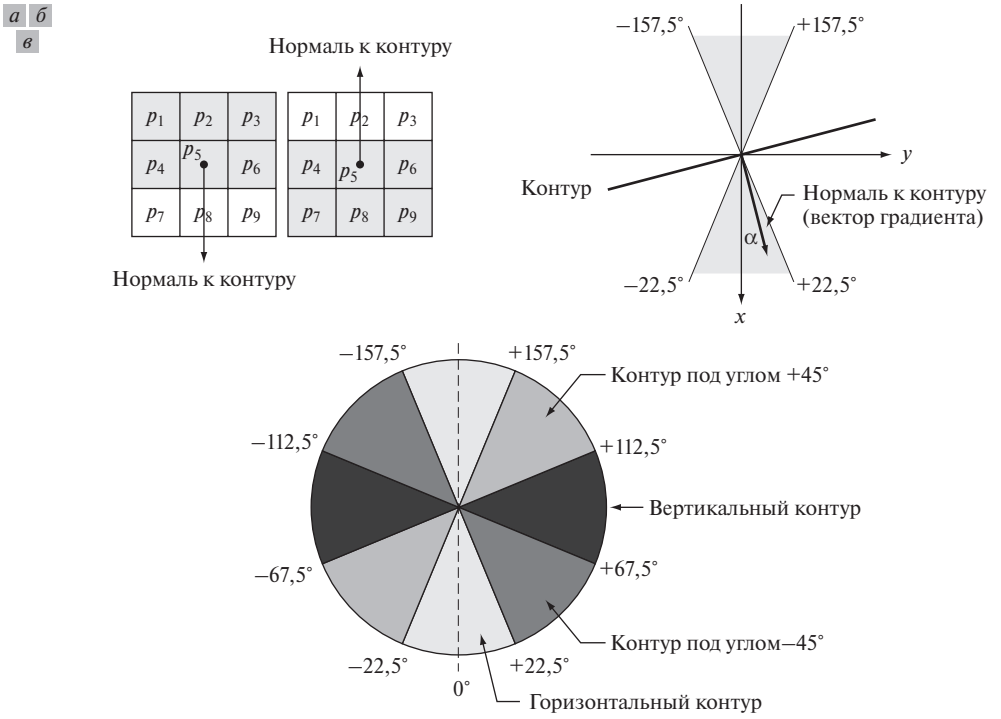


Рис. 10.24. Две возможные ориентации горизонтального контура (темные пиксели) в окрестности 3×3. (*б*) Темным цветом обозначен диапазон углов α *нормали к контуру*, когда контур считается горизонтальным. (*в*) Диапазоны углов нормали для четырех направлений контура в окрестности 3×3. Каждому направлению соответствуют два диапазона, обозначенные одинаковым цветом

лученного с помощью формулы (10.2-32). Как видно на рис. 10.24(*б*), если угол нормали к контуру находится в интервале от –22,5° до +22,5° или от –157,5° до +157,5°, такой контур считается горизонтальным. На рис. 10.24(*в*) показаны диапазоны углов, соответствующих всем четырем рассматриваемым направлениям контуров.

Обозначим через d_1, d_2, d_3 и d_4 четыре только что определенных базовых направления контура в окрестности 3×3: соответственно горизонтальное, −45°, вертикальное и +45°. Можно сформулировать следующую схему подавления немаксимальных точек для окрестности 3×3 с центром в *каждой* точке (x, y) изображения $\alpha(x, y)$ и формирования изображения $g_N(x, y)$ с утоненными контурами:

1. Находим направление d_k, ближайшее к $\alpha(x, y)$.
2. Если значение $M(x, y)$ меньше, чем хотя бы у одного из соседей в направлении d_k, принимаем $g_N(x, y) = 0$ (подавление); в противном случае $g_N(x, y) = M(x, y)$.

Здесь $g_N(x, y)$ — изображение с подавленными немаксимальными точками. Например, на рис. 10.24(*а*) в качестве (x, y) выступает элемент p_5, и, если предполагать, что через него проходит горизонтальный контур, на шаге 2 выше мы долж-

ны проверять значения пикселей p_2 и p_8. Изображение $g_N(x,y)$ содержит только утонченные контуры; оно отличается от $M(x,y)$ тем, что все немаксимальные точки контуров подавлены.

Заключительная операция состоит в пороговой обработке изображения $g_N(x,y)$ с целью уменьшения числа ложных контурных точек. В разделе 10.2.5 использовалось пороговое преобразование с одним порогом, при котором устанавливались в 0 значения во всех точках, где они меньше этого порога. Если установлен слишком низкий порог, то останутся некоторые ложные контуры (в задачах идентификации такие ошибки называют *ложная тревога*[12]). Если установить слишком высокий порог, то исчезнут истинные контурные точки (так называемые ошибки *пропуска*). В алгоритме Кэнни делается попытка улучшить такое положение с помощью порогового преобразования с *гистерезисом*, в котором, как мы увидим в разделе 10.3.6, используются два порога — нижний T_L и верхний T_H. Кэнни предложил использовать пороги с отношением величины верхнего порога к величине нижнего от 2—3 до 1.

Результаты порогового преобразования можно визуализировать, сформировав еще два изображения

$$g_{NH}(x,y) = g_N(x,y) \geq T_H \tag{10.2-33}$$

и

$$g_{NL}(x,y) = g_N(x,y) \geq T_L, \tag{10.2-34}$$

где первоначально все элементы $g_{NH}(x,y)$ и $g_{NL}(x,y)$ имеют нулевые значения. После такого порогового преобразования в изображении $g_{NH}(x,y)$ будет в общем случае меньше ненулевых пикселей, чем в $g_{NL}(x,y)$, но все ненулевые пиксели $g_{NH}(x,y)$ повторятся и в $g_{NL}(x,y)$, так как последнее изображение сформировано с меньшим значением порога. Удалим на изображении $g_{NL}(x,y)$ все пиксели с ненулевыми значениями в $g_{NH}(x,y)$, полагая

$$g_{NL}(x,y) = g_{NL}(x,y) - g_{NH}(x,y). \tag{10.2-35}$$

Теперь ненулевые пиксели изображений $g_{NH}(x,y)$ и $g_{NL}(x,y)$ можно рассматривать как соответственно «сильные» и «слабые» контурные точки.

После описанных пороговых преобразований все сильные пиксели в $g_{NH}(x,y)$ можно считать точками истинных контуров и сразу их пометить соответствующим образом. В зависимости от величины порога T_H контуры на изображении $g_{NH}(x,y)$ обычно будут прерывистыми. Более длинные контуры формируются по следующему алгоритму:

(а) Найти очередной необработанный контурный пиксель p на изображении $g_{NH}(x,y)$.

(б) Пометить как истинные контурные пиксели все слабые пиксели в изображении $g_{NL}(x,y)$, которые являются связными с p, скажем, в смысле 8-связности.

(в) Если обработаны все ненулевые пиксели $g_{NH}(x,y)$, перейти к шагу (г), в противном случае перейти к шагу (а).

---

[12] В оригинале использованы термины *false positive* и *false negative*, которые мы переводим, следуя терминологии, принятой в задачах идентификации (также называются ошибками 1 и 2 рода). — *Прим. перев.*

(г) Установить в 0 значения всех пикселей $g_{NL}(x,y)$, которые не были помечены как истинные.

По окончании изложенной процедуры окончательным результатом алгоритма Кэнни будет изображение, получаемое добавлением к $g_{NH}(x,y)$ всех оставшихся ненулевых пикселей $g_{NL}(x,y)$.

Мы использовали два добавочных изображения $g_{NH}(x,y)$ и $g_{NL}(x,y)$ для простоты изложения. На практике пороговое преобразование с гистерезисом можно выполнять одновременно с подавлением немаксимальных точек и реализовать его прямо для изображения $g_N(x,y)$, формируя списки сильных пикселей и связных с ними слабых пикселей.

В итоге алгоритм выделения контуров Кэнни состоит из следующих основных шагов:

1. Сгладить исходное изображение гауссовым фильтром.
2. Сформировать изображения модуля и направления градиента.
3. Применить подавление немаксимальных точек к изображению модуля градиента.
4. Выполнить преобразование с двойным порогом и анализ связности для обнаружения и связывания контуров.

Хотя контуры, полученные после подавления немаксимальных точек, тоньше найденных непосредственно на основании градиента, контуры с шириной более одного пикселя все же могут оставаться. Для получения контуров шириной 1 пиксель вслед за шагом 4 обычно выполняют один проход алгоритма утончения (см. раздел 9.5.5).

Как уже говорилось, сглаживание осуществляется путем свертки исходного изображения с гауссовой маской, размеры которой $n \times n$ необходимо задать. Для выбора значения n можно использовать метод, обсуждавшийся в предыдущем разделе в связи с алгоритмом Марра—Хилдрета. В соответствии с этим методом маска фильтра, полученная дискретизацией соотношения (10.2-29) со значением n, равным наименьшему нечетному целому числу, равному или больше 6σ, достаточна для обеспечения практически «полной» характеристики сглаживания гауссова фильтра. Если практические соображения вынуждают применять для фильтрации маску меньших размеров, за это приходится платить снижением качества сглаживания.

Несколько заключительных замечаний касательно реализации. Как отмечалось выше при обсуждении детектора контуров Марра—Хилдрета, двумерная гауссова функция в формуле (10.2-29) разлагается в произведение двух одномерных гауссианов. Следовательно, шаг 1 алгоритма Кэнни можно переформулировать как вычисление одномерных сверток отдельно для каждой строки (или столбца) изображения и затем повторное вычисление одномерных сверток соответственно по столбцам (или строкам) полученного результата. Более того, если используются приближения согласно формулам (10.2-12) и (10.2-13), можно также с помощью одномерных сверток вычислять градиент на шаге 2 (задача 10.20).

Пример 10.8. Иллюстрация метода обнаружения контуров Кэнни.

■ На рис. 10.25(*а*) показано уже знакомое изображение здания. Для сравнения на рис. 10.25(*б*) и (*в*) представлены результаты выделения контуров, ранее

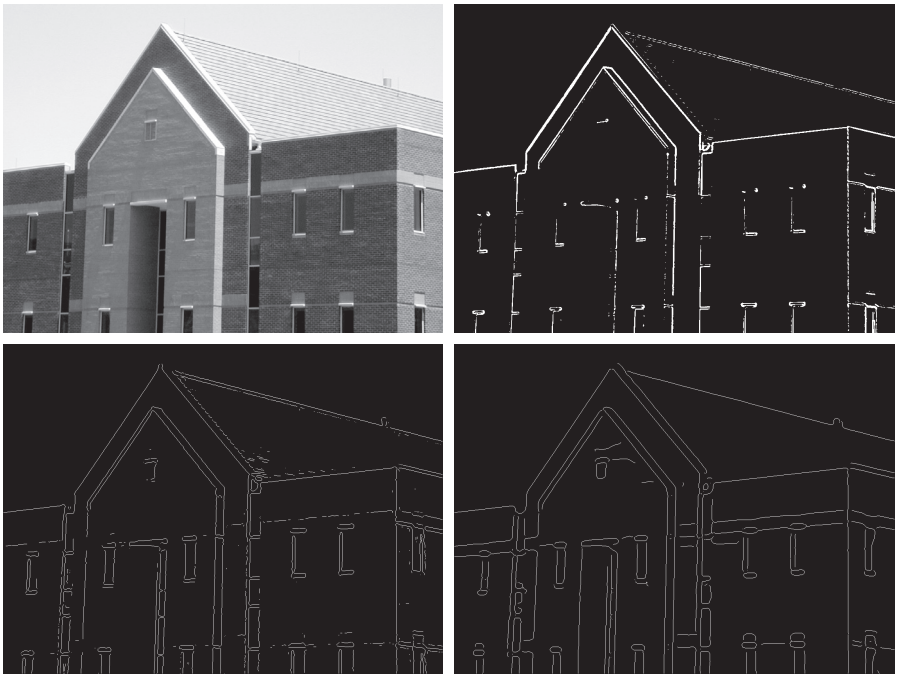


Рис. 10.25. (*а*) Исходное изображение размерами 834×1114 пикселей, значения яркости которого приведены к интервалу [0, 1]. (*б*) Результат порогового преобразования градиента сглаженного изображения. (*в*) Результат работы алгоритма Марра—Хилдрета. (*г*) Результат работы алгоритма Кэнни. Обратите внимание на значительное улучшение по сравнению с (*б*) и (*в*)

приведенные соответственно на рис. 10.20(*б*) (получен сочетанием градиента с пороговым преобразованием) и рис. 10.22(*г*) (получен детектором контуров Марра—Хилдрета). Напомним, что параметры указанных преобразований выбирались так, чтобы выделить основные контуры на изображении, одновременно стараясь ослабить «несущественные» детали, такие как контуры кирпичей кладки и черепицы крыши.

На рис. 10.25(*г*) представлен результат, полученный с помощью алгоритма Кэнни при значениях параметров $T_L = 0{,}04$, $T_H = 0{,}10$ (в 2,5 раза больше нижнего порога), $\sigma = 4$ и маски размерами 25×25, что соответствует наименьшему нечетному целому числу, превышающему 6σ. Эти параметры подбирались в интерактивном режиме, чтобы достичь тех же целей, что указаны в предыдущем абзаце для градиента и алгоритма Марра—Хилдрета. Сравнение результатов алгоритма Кэнни с двумя другими изображениями показывает, что значительно улучшились детали основных контуров и одновременно с этим сильнее подавлены несущественные детали. Обратите внимание, например, что алгоритм Кэнни выделил оба контура бетонной ленты в кирпичной кладке в верхней части стены, тогда как при пороговом градиентном преобразовании исчезли оба контура, а детектор Марра—Хилдрета выделил только верхний контур. Что касается несущественных деталей, на изображении от детектора Кэнни нет ни одного контура от черепицы, а для двух других изображений это неверно. Качество вы-

деленных линий в смысле толщины, непрерывности и прямолинейности также лучше всего для изображения, полученного алгоритмом Кэнни. Благодаря подобным результатам, этот алгоритм стал наиболее часто применяемым методом выделения контуров. ■

Приведенные здесь значения параметров следует рассматривать лишь как ориентировочные. Реализация большинства алгоритмов включает в себя различные шаги масштабирования, например приведение значений яркости исходного изображения к интервалу [0, 1]. Для других схем масштабирования, очевидно, потребуются величины порогов, отличающиеся от указанных в данном примере.

Пример 10.9. Другая иллюстрация трех основных методов выделения контуров.

■ Для еще одного сравнения трех основных методов выделения контуров, рассмотренных в данном разделе, рассмотрим КТ-изображение головы на рис. 10.26(*а*) размерами 512×512 пикселей. В этом примере задача состоит том, чтобы выделить контуры внешних очертаний головного мозга (область серого цвета), об-

а б
в г

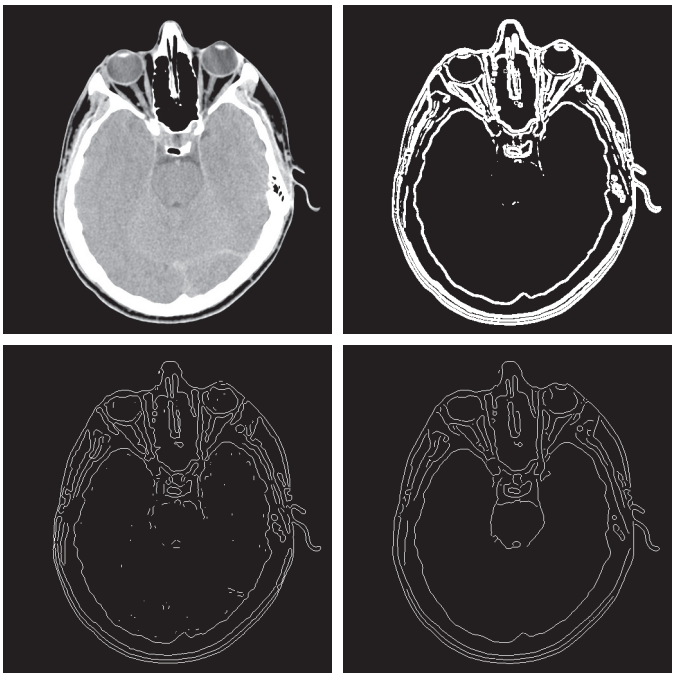


Рис. 10.26. (*а*) Исходное томографическое изображение головы с размерами 512×512 пикселей, значения яркости которого преобразованы к интервалу [0, 1]. (*б*) Результат порогового преобразования градиента сглаженного изображения. (*в*) Результат работы алгоритма Марра—Хилдрета. (*г*) Результат работы алгоритма Кэнни. (Исходное изображение предоставил д-р Дэвид Р. Пикенс, университет Вандербилта)

ласти спинного мозга (находится прямо за носовой полостью в переднем отделе мозга) и внешний контур головы. Мы хотим получить как можно более непрерывные и тонкие линии контуров, которые бы не содержали деталей в областях серого цвета, соответствующих глазам и мозгу.

На рис. 10.26(*б*) приведен результат пороговой обработки градиентного изображения, предварительно сглаженного усредняющим фильтром 5×5; порог был выбран как 15% от максимального значения на градиентном изображении. На рис. 10.26(*в*) показан результат работы алгоритма выделения контуров Марра — Хилдрета с порогом величиной 0,002, $\sigma = 3$ и маской размерами 19×19 пикселей. Рис. 10.26(*г*) был получен алгоритмом Кэнни при $T_L = 0{,}05$, $T_H = 0{,}15$ (в 3 раза больше нижнего порога), $\sigma = 2$ и с маской размерами 13×13, что, как и в случае детектора Марра — Хилдрета, соответствует наименьшему нечетному целому числу, превышающему 6σ.

В плане качества выделенных контуров и способности устранять несущественные детали результаты на рис. 10.26 хорошо согласуются с результатами и обсуждением в предыдущем примере. Отметим также, что только алгоритм Кэнни оказался способен получить тонкий непрерывный контур задней границы мозга. Он также стал единственной процедурой, которая смогла выделить контуры лучшего качества и при этом избежать появления ложных контуров из-за неравномерностей в области серого вещества на исходном изображении. ■

Как того можно было ожидать, лучшее качество работы алгоритма Кэнни достигается ценой более сложной реализации по сравнению с двумя ранее рассмотренными методами и требует также заметно большего времени счета. В некоторых практических задачах, например при обработке промышленных изображений в реальном времени, соображения стоимости и быстродействия обычно вынуждают применять более простые методы, главным образом градиентный с пороговым преобразованием. Если основное значение придается качеству контуров, то алгоритмы Марра — Хилдрета и Кэнни, особенно последний, являются превосходными альтернативами.

10.2.7. Связывание контуров и нахождение границ

В идеале методы обнаружения контуров должны выделять в изображении множества пикселей, лежащих только на контурах. Однако на практике выделяемые пиксели редко отображают контур достаточно точно, и причина этого — шумы, разрывы контуров из-за неоднородности освещения, а также влияние прочих эффектов, нарушающих непрерывность яркостной картины. Поэтому алгоритмы обнаружения контуров обычно дополняются алгоритмами связывания, чтобы сформировать из множества контурных точек содержательные контуры и/или границы областей. В этом разделе мы рассмотрим три фундаментальных подхода к связыванию контуров, на которых обычно строятся используемые на практике методы. Первый подход основан на знании контурных точек в локальной окрестности (например размерами 3×3), при втором подходе требуется знать точки на границе области, а третий представляет собой глобальный подход, применяемый к контурному изображению в целом.

Локальная обработка

Один из простейших подходов к связыванию точек контура состоит в анализе характеристик пикселей в небольшой окрестности каждой точки (x, y) изображения, которая была отмечена как контурная точка с помощью какого-либо из рассмотренных в предыдущем разделе методов. Все точки, являющиеся сходными в соответствии с некоторыми заранее заданными критериями, связываются и образуют контур, состоящий из пикселей с общими свойствами согласно этим критериям.

При таком анализе используются следующие два основных параметра для установления сходства пикселей контура: (1) величина (модуль) градиента и (2) направление вектора градиента. Первый параметр задается выражением (10.2-10). Обозначим S_{xy} множество координатных точек окрестности с центром в точке (x, y). Пиксель контура, имеющий координаты (s, t), расположенный внутри S_{xy}, считается сходным по *модулю градиента* с пикселем (x, y), если

$$|M(s,t) - M(x,y)| \le E, \tag{10.2-36}$$

где E — заданный положительный порог.

Направление (угол) вектора градиента задается выражением (10.2-11). Пиксель контура с координатами (s, t), расположенный внутри S_{xy}, считается сходным по *направлению градиента* с пикселем (x, y), если

$$|\alpha(s,t) - \alpha(x,y)| \le A, \tag{10.2-37}$$

где A — заданный положительный угловой порог. Как уже отмечалось в разделе 10.2.5, направление контура в точке (x, y) *перпендикулярно* направлению вектора градиента в этой точке.

Пиксель с координатами (s, t) внутри окрестности S_{xy} объединяется с центральным пикселем (x, y), если выполнены критерии сходства и по величине, и по направлению. Этот процесс повторяется в каждой точке изображения с одновременным запоминанием найденных связанных пикселей при движении центра окрестности. Простой способ учета данных состоит в том, что каждому множеству связываемых пикселей контура присваивается свой идентификатор (например значение яркости, если известно, что число контуров невелико).

Изложенный метод обработки является трудоемким в плане вычислений, так как необходимо исследовать все соседние элементы каждой точки изображения. Упрощенная процедура, особенно хорошо подходящая для обработки в реальном времени, состоит из следующих шагов:

1. По исходному изображению $f(x, y)$ вычислить массивы модуля $M(x, y)$ и направления $\alpha(x, y)$ градиента.
2. Сформировать двоичное изображение g, значение которого в любой координатной точке (x, y) задается правилом

$$g(x,y) = \begin{cases} 1 & \text{если } M(x,y) > T_M \text{ И } \alpha(x,y) = A \pm T_A \\ 0 & \text{в противном случае,} \end{cases}$$

   где T_M — порог, A — заданное направление, а $\pm T_A$ определяет диапазон допустимых отклонений от A.
3. Построчно сканируя изображение g, заполнить (установить значение 1) все имеющиеся в строке промежутки с нулевыми значениями, длина

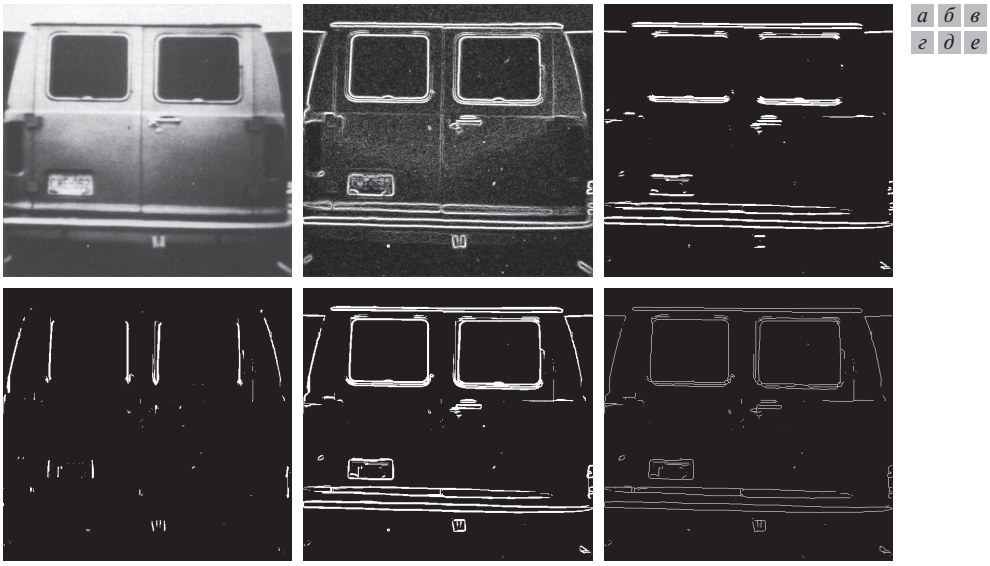
которых не превышает заданной величины K. Заметим, что промежуток по определению ограничен с обеих сторон одним или более пикселей с единичным значением. Строки обрабатываются независимо, без сохранения переходной информации между ними.

4. Для заполнения промежутков в любом другом направлении θ повернуть изображение g на угол θ и повторить процедуру построчной обработки из шага 3, после чего повернуть результат обратно на угол $-\theta$.

Если для конкретной задачи требуется связывать контуры только горизонтального и вертикального направлений, шаг 4 превращается в простую процедуру поворота g на 90°, затем полученное изображение обрабатывается построчно и результат поворачивается обратно. Этот метод чаще всего применяется на практике и дает хорошие результаты, как и показывает следующий пример.

Пример 10.10. Связывание контуров с помощью локальной обработки.

■ На рис. 10.27(*а*) показан снимок автомобиля сзади. Цель этого примера состоит в том, чтобы проиллюстрировать использование изложенного выше алгоритма для отыскания прямоугольников, которые по своим размерам могут являться подходящими кандидатами на роль номерного знака. Такие прямоугольники могут формироваться на основе обнаружения строго вертикальных и горизонтальных контуров. На рис. 10.27(*б*) показано изображение модуля градиента $M(x,y)$, а рис. 10.27(*в*) и (*г*) иллюстрируют результаты шагов 3 и 4 алгоритма, полученные при величине порога T_M, равной 30% максимального



а б в
г д е

Рис. 10.27. (*а*) Изображение автомобиля сзади, имеющее размеры 534×566 пикселей. (*б*) Изображение модуля градиента. (*в*) Горизонтально связанные контурные пиксели. (*г*) Вертикально связанные контурные пиксели. (*д*) Результат применения логической операции ИЛИ к (*в*) и (*г*). (*е*) Окончательный результат, полученный с помощью операции морфологического утончения. (Исходное изображение предоставлено компанией Perceptics Corporation)

значения модуля градиента, $A = 0°/90°$, $T_A = 45°$, с заполнением всех пропусков длиной 25 пикселей или менее (что составляет приблизительно 5 % от ширины изображения). Использование большого диапазона допустимых направлений ($T_A = 45°$) потребовалось для того, чтобы успешно обнаруживать скругленные углы номерного знака и задних окон автомобиля. На рис. 10.27(*д*) представлен результат применения логической операции ИЛИ к двум предыдущим изображениям, а рис. 10.27(*е*) был получен из него с помощью процедуры утончения, рассмотренной в разделе 9.5.5. Как видно из рис. 10.27(*е*), прямоугольник, отвечающий номерному знаку, является одним из нескольких обнаруженных на этом изображении прямоугольников. Зная их все, выбор нужного становится несложной задачей, поскольку соотношение ширины и высоты номерного знака в США равно 2:1. ■

Обработка в интересующей области

Часто встречаются ситуации, когда области изображения, которые представляют интерес, известны заранее или могут быть как-то определены. Это значит, что известно, какие пиксели соответствующего контурного изображения принадлежат интересующей области. В таких ситуациях методы связывания контуров могут основываться на знаниях об области, и желаемый результат состоит в построении аппроксимации границы этой области. Одним из таких методов обработки является аппроксимация функциями заданного вида, когда строится аппроксимация плоской кривой по известным точкам. Как правило, интерес представляют методы быстрой обработки, позволяющие приближенно найти существенные особенности границы области, например экстремальные точки и вогнутости. Особенно привлекательна аппроксимация ломаной линией, потому что она позволяет сравнительно просто описать существенные признаки

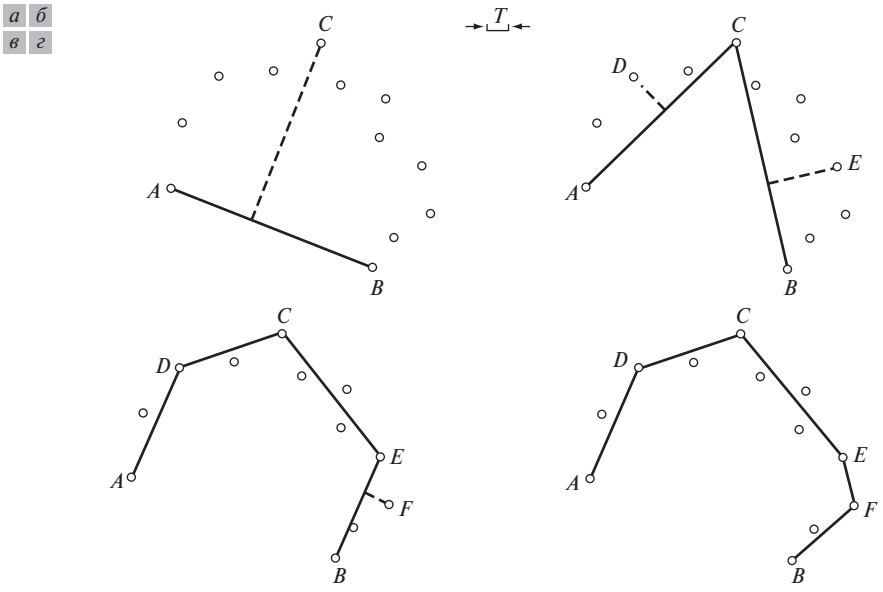


Рис. 10.28. Иллюстрация итерационного алгоритма аппроксимации ломаной линией

формы границы области (в виде вершин многоугольника). В данном разделе мы изложим и проиллюстрируем алгоритм, подходящий для такой цели.

Перед изложением алгоритма рассмотрим принцип его работы на простом примере. На рис. 10.28 изображено множество точек, представляющее разомкнутую кривую с отмеченными концевыми точками *A* и *B*. Эти две точки по определению являются вершинами аппроксимирующей ломаной. Сначала вычисляем параметры прямой, проходящей через точки *A* и *B*, после чего находим точку множества, расположенную на максимальном расстоянии от этой прямой (при неоднозначном выборе берем любую из точек). Если величина расстояния превышает заданный порог *T*, принимаем эту точку за вершину ломаной, обозначенную *C*, как показано на рис. 10.28(*а*). Затем проводятся линии от *A* до *C* и от *C* до *B* и определяются расстояния от прямой *AC* до всех точек множества, расположенных между точками *A* и *C*. Точка, находящаяся на максимальном расстоянии, объявляется вершиной *D*, если это расстояние превышает порог *T*; в противном случае для этого отрезка не определяется новых вершин ломаной. Аналогичная процедура применяется к точкам между *C* и *B*. Результаты этого и следующего шагов представлены на рис. 10.28(*б*) и (*в*). Описанная процедура продолжается до тех пор, пока больше не остается точек, отвечающих условию порога. На рис. 10.28(*г*) показан окончательный результат аппроксимации формы кривой, отвечающей заданным точкам.

В описанной процедуре неявно подразумеваются два требования. Во-первых, необходимо указать две концевые точки; во-вторых, все точки должны быть упорядочены (например пронумерованы по часовой стрелке или в противоположном направлении). В тех ситуациях, когда множество точек на плоскости не образует связный путь (что типично для контурных изображений), не всегда просто определить, относятся ли эти точки к участку границы (разомкнутая кривая) или к целой границе (замкнутая кривая). Если точки упорядочены, то, анализируя расстояния между ними, можно выяснить, является ли кривая замкнутой или разомкнутой. Ситуация, когда расстояние между какими-то двумя соседними точками упорядоченной последовательности велико по сравнению с расстояниями между остальными соседними точками, уверенно указывает на то, что кривая является разомкнутой в этом месте, и найденные таким способом концы кривой далее могут использоваться как начальные точки для вышеизложенного алгоритма. Если же расстояния между соседними точками мало различаются, то скорее всего мы имеем дело с замкнутой кривой. В этом случае имеется несколько вариантов выбора двух начальных точек. В одном из них выбираются самая левая и самая правая точки множества. Другой метод состоит в нахождении экстремальных точек кривой (один из способов, как это сделать, рассматривается в разделе 11.2.1). Итак, алгоритм построения аппроксимирующей ломаной для разомкнутых и замкнутых кривых можно сформулировать следующим образом.

1. Пусть *P* — множество упорядоченных несовпадающих точек двоичного изображения, имеющих значение 1. Выбрать две начальные точки *A* и *B* в качестве вершин ломаной.

> Один из алгоритмов, формирующих упорядоченную последовательность точек, изложен в разделе 11.1.1.

2. Задать порог T и завести два пустых стека, OPEN и CLOSED.

Выбор таких имен для стеков не связан с тем, является ли кривая разомкнутой или замкнутой, а указывает лишь на то, используется ли стек для хранения окончательных (CLOSED) или промежуточных (OPEN) вершин ломаной.

3. Если точки множества P соответствуют замкнутой кривой, поместить в OPEN B, затем A, *а также* поместить B в CLOSED. Если точки P соответствуют разомкнутой кривой, поместить A в OPEN, а B в CLOSED.
4. Вычислить параметры прямой, соединяющей последнюю вершину в CLOSED с последней вершиной в OPEN.
5. Вычислить расстояния от прямой, построенной на шаге 4, до всех точек последовательности, находящихся между вершинами, используемыми на шаге 4. Выбрать точку V_{max} с максимальным расстоянием D_{max} (в случае неоднозначности — любую из таких точек).
6. Если $D_{max} > T$, поместить V_{max} в стек OPEN в качестве новой вершины и перейти к шагу 4.
7. Иначе переместить последнюю вершину из OPEN в CLOSED в качестве новой вершины.
8. Если OPEN не пустой, перейти к шагу 4.
9. Иначе выход. Вершины в CLOSED суть вершины ломаной, аппроксимирующей множество точек P.

Принцип действия алгоритма иллюстрируют следующие два примера.

Пример 10.11. Связывание контуров с помощью аппроксимации ломаной линией.

■ Рассмотрим множество точек P на рис. 10.29(*а*). Предположим, что эти точки принадлежат замкнутой кривой, упорядочены в направлении по часовой стрелке (обратите внимание, что некоторые из соседних точек последовательности не являются смежными) и что в качестве A и B выбраны соответственно самая левая и самая правая точки P. Это начальные вершины ломаной, как указано в табл. 10.1. Пусть первой в последовательности является самая левая точка A. На рис. 10.29(*б*) показана единственная точка (обозначенная C) в верхнем отрезке кривой между A и B, удовлетворяющая условию шага 6 алгоритма, поэтому она взята в качестве новой вершины и добавлена в стек OPEN. Вторая строка табл. 10.1 фиксирует нахождение точки C, а третья строка — занесение точки в OPEN. Величина порога T на рис. 10.29(*б*) составляет приблизительно 1,5 шага сетки.

Заметим, что на рис. 10.29(*б*) есть точка ниже линии AB, также удовлетворяющая условию шага 6. Однако поскольку точки упорядочены, каждый раз анализируется только одно подмножество точек, расположенных между двумя рассматриваемыми вершинами. Другая точка в нижнем отрезке кривой будет обнаружена позже, как показывает рис. 10.29(*д*). Важно всегда соблюдать заданный порядок нумерации точек.

В табл. 10.1 приведены детали отдельных шагов, которые привели к решению на рис. 10.29(*з*). Были обнаружены четыре вершины, после соединения которых отрезками образовался многоугольник, аппроксимирующий данные

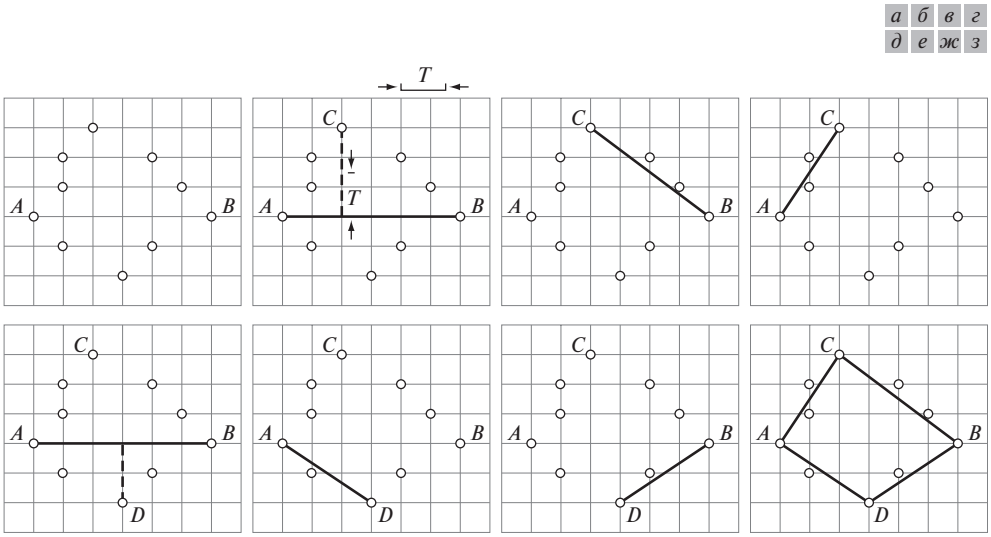


Рис. 10.29. (*а*) Множество точек, упорядоченных в направлении по часовой стрелке (в качестве начальных вершин выбраны точки *A* и *B*). (*б*) Расстояние от точки *C* до прямой *AB* является максимальным по сравнению с другими точками между *A* и *B* и превышает порог. (*в*) Поэтому *C* становится новой вершиной. (*г*)—(*ж*) Последующие стадии работы алгоритма. (*з*) Окончательный набор соединенных отрезками вершин, образующих многоугольник. Подробности выполнения шагов представлены в табл. 10.1

граничные точки. Обратите внимание, что порядок обнаружения вершин *B*, *C*, *A*, *D*, *B* соответствует направлению против часовой стрелки, несмотря на то, что для нахождения вершин точки обходились в противоположном направлении. Если бы исходные данные относились к разомкнутой кривой, то вершины бы обнаруживались по часовой стрелке. Причина такого расхождения кроется в способе инициализации стеков OPEN и CLOSED. Разница в заполнении стека CLOSED для разомкнутой и замкнутой кривых также приводит к тому, что последняя найденная вершина замкнутой ломаной повторяет первую. Это согласуется с тем, как человек различает разомкнутую и замкнутую ломаную, если даны только ее вершины. ■

Таблица 10.1. Подробности работы алгоритма в примере 10.11 по шагам

Стек CLOSED	Стек OPEN	Обработанный отрезок кривой	Построенная вершина
B	*B*, *A*	—	*A*, *B*
B	*B*, *A*	(*BA*)	*C*
B	*B*, *A*, *C*	(*BC*)	—
B, *C*	*B*, *A*	(*CA*)	—
B, *C*, *A*	*B*	(*AB*)	*D*
B, *C*, *A*	*B*, *D*	(*AD*)	—
B, *C*, *A*, *D*	*B*	(*DB*)	—
B, *C*, *A*, *D*, *B*	Пусто	—	—

Пример 10.12. Аппроксимация ломаной линией границы на изображении. ■ Рис. 10.30 демонстрирует более близкий к практике пример аппроксимации ломаной линией. На рис. 10.30(*а*) приведено исходное рентгеновское изображение человеческого зуба, которое имеет размеры 550×566 пикселей и диапазон яркостей, преобразованный к интервалу [0, 1]. В этом примере задача состоит в том, чтобы выделить границу зуба для сравнения ее с базой данных при проведении криминалистической экспертизы. На рис. 10.30(*б*) показано градиентное изображение, полученное с помощью масок Собела и последующего порогового преобразования с величиной порога T = 0,1 (10 % максимального значения яркости). Как и можно было ожидать для рентгеновского изображения, градиент содержит значительную шумовую составляющую, поэтому первый шаг состоит в снижении шума. Поскольку изображение двухградационное, для этой цели

а *б* *в* *г*
д *е* *ж* *з*
и *к* *л* *м*

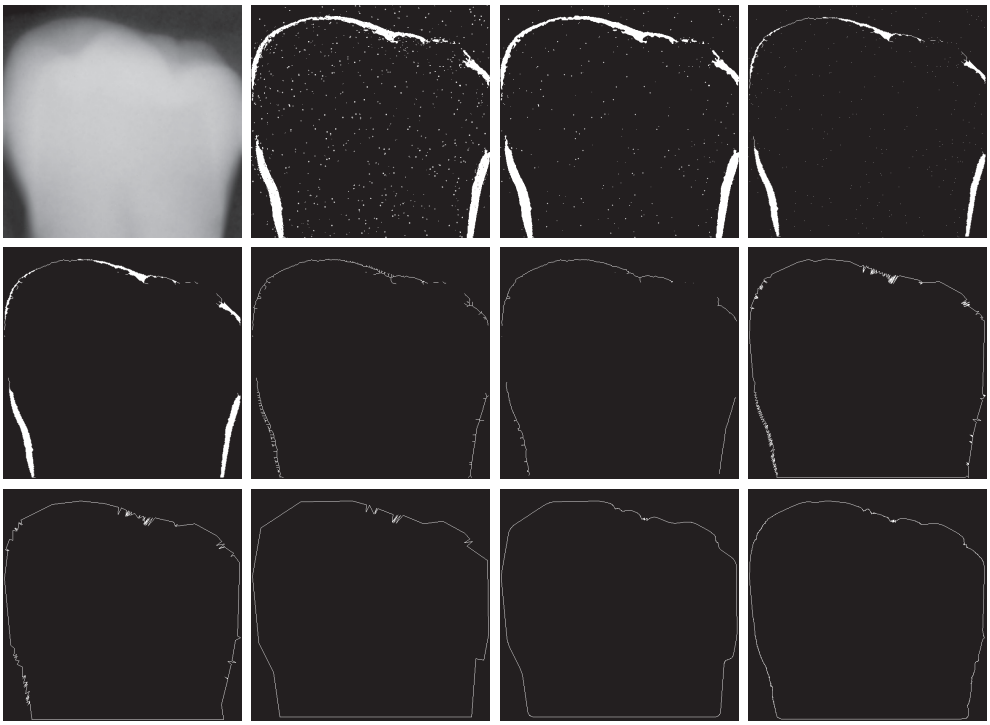


Рис. 10.30. (*а*) Рентгеновское изображение зуба человека, имеющее размеры 550×566 пикселей. (*б*) Градиентное изображение. (*в*) Результат мажоритарной фильтрации. (*г*) Результат морфологического утончения. (*д*) Результат морфологической очистки. (*е*) Построение остова. (*ж*) Усечение отростков. (*з*)—(*к*) Результаты аппроксимации границы ломаной линией при величинах порога приблизительно 0,5 %, 1 % и 2 % от ширины изображения (T = 3, 6 и 12). (*л*) Результат сглаживания границы (*к*) с помощью одномерного усредняющего фильтра размерами 1×31 (приблизительно 5 % от ширины изображения). (*м*) Результат сглаживания границы (*з*) тем же фильтром

хорошо подходят морфологические методы. Рис. 10.30(*в*) демонстрирует результат *мажоритарной фильтрации*, при которой пикселю присваивается значение 1, если в его окрестности 3×3 содержится пять и более единичных пикселей, и 0 в противоположном случае. Хотя шума стало меньше, некоторые шумовые точки все еще хорошо видны. На рис. 10.30(*г*) представлен результат операции морфологического утончения, которая привела к дальнейшему снижению шума до изолированных точек. На рис. 10.30(*д*) эти точки устраняются способом морфологической фильтрации, как описывалось в примере 9.4. К этому моменту изображение состоит из утолщенных границ, которые утончаются с помощью морфологической операции построения остова, как показывает рис. 10.30(*е*). Наконец, на рис. 10.30(*ж*) приведен результат последнего шага предварительной обработки — усечения отростков методом из раздела 9.5.8.

Далее будем аппроксимировать ломаной линией точки на рис. 10.30(*ж*). Рис. 10.30(*з*)—(*к*) демонстрируют результаты работы алгоритма аппроксимации с помощью ломаной линии при величинах порога 0,5%, 1% и 2% от ширины изображения (T = 3, 6 и 12). Первые два дают хорошее приближение, но третий является мало подходящим. Избыточная зазубренность краев во всех трех случаях ясно указывает на необходимость сглаживания границы. На рис. 10.30(*л*) и (*м*) приведены результаты свертки усредняющей маски с изображениями (*к*) и (*з*) соответственно. Маска имеет размеры 1×31 пиксель (приблизительно 5% от ширины изображения) со всеми элементами, равными 1. Как можно было ожидать, результат на рис. 10.30(*л*) по-прежнему не удовлетворителен в смысле сохранения важных характеристик формы границы (например ее правая сторона сильно искажена). Напротив, результат на рис. 10.30(*м*) демонстрирует существенно сглаженную границу и в достаточной степени сохраняет признаки формы. Например, форма округлости левого верхнего кончика зуба и детали правого верхнего кончика переданы достаточно правильно. ■

Предыдущий пример демонстрирует типичные результаты, которых можно достичь с помощью рассмотренного в данном разделе алгоритма аппроксимации ломаной линией. Достоинство этого алгоритма в том, что он прост в реализации и дает в общем вполне приемлемые результаты. В разделе 11.1.3 мы рассмотрим более сложный алгоритм, дающий более точную аппроксимацию за счет вычисления ломаной с минимальной длиной.

Глобальный анализ с помощью преобразования Хафа

Рассмотренные в двух предшествующих разделах методы применимы в тех ситуациях, когда имеется хотя бы частичная информация о пикселях, принадлежащих отдельным объектам. Например, при обработке в области интереса имеет смысл связывать данный набор пикселей, только если известно, что они являются частью границы осмысленной области. Часто приходится работать в условиях отсутствия какой-либо структуры, когда все, что имеется — это контурное изображение и нет сведений о том, где могут находиться интересующие объекты. В таких ситуациях все контурные пиксели являются кандидатами для связывания и должны приниматься или отвергаться, исходя из заранее определенных *глобальных* свойств. В этом разделе излагается метод, основанный на выяснении того, лежат ли множества пикселей на некоторой кривой заданной формы. Будучи обнаруженными, такие кривые образуют контуры границы интересующей области.

Пусть задано n точек на изображении (плоскости). Предположим, что требуется найти подмножества этих точек, лежащие на прямых линиях. Одно из возможных решений этой задачи состоит в том, что вначале находятся все прямые, определяемые каждой парой точек, а затем ищутся все подмножества точек, близких к конкретным прямым. Такая процедура связана с необходимостью нахождения $n(n-1)/2 \sim n^2$ прямых, а затем выполнения $n(n(n-1))/2 \sim n^3$ сравнений каждой точки со всеми прямыми. Вычислительная сложность данного подхода позволяет применять его лишь в самых тривиальных прикладных задачах.

Хаф [Hough, 1962] предложил альтернативный подход (который принято называть *преобразованием Хафа*). Возьмем точку (x_i, y_i) из заданного множества n точек и рассмотрим общее уравнение прямой на плоскости xy в форме с угловым коэффициентом $y = ax + b$. Очевидно, что через точку (x_i, y_i) проходит бесконечно много прямых, удовлетворяющих уравнению $y_i = ax_i + b$ при различных значениях a и b. Однако если переписать это уравнение в виде $-b = -x_i a + y_i$ и рассмотреть плоскость ab, называемую *пространством параметров*, то для заданной пары (x_i, y_i) получаем уравнение *единственной* прямой. Более того, другой точке (x_j, y_j) также соответствует своя прямая в пространстве параметров, и если они не параллельны друг другу, эти две прямые пересекаются в некоторой точке (a', b'), такой, что a' есть угловой коэффициент, а b' — точка пересечения с осью y прямой, проходящей через точки (x_i, y_i) и (x_j, y_j) в плоскости xy. На самом деле каждой точке прямой, проходящей через точки (x_i, y_i) и (x_j, y_j), в пространстве параметров соответствует своя прямая линия, причем все они пересекаются в точке (a', b'). Это иллюстрирует рис. 10.31.

В принципе, можно изобразить в пространстве параметров прямые, соответствующие всем точкам (x_k, y_k) заданного множества на плоскости xy, и основные линии на исходной плоскости можно найти по тем точкам в пространстве параметров, где пересекается большое число линий. Однако использовать на практике этот подход затруднительно, поскольку если прямая близка к вертикали, ее угловой коэффициент a стремится к бесконечности. Один из способов обойти эту трудность состоит в представлении прямой с помощью нормали

$$x\cos\theta + y\sin\theta = \rho. \tag{10.2-38}$$

Рис. 10.32(*а*) дает геометрическую интерпретацию параметров ρ и θ. Для горизонтальной прямой $\theta = 0°$, а ρ будет равно координате точки ее пересечения с осью x. Аналогично для вертикальной прямой ρ есть координата точки ее пе-

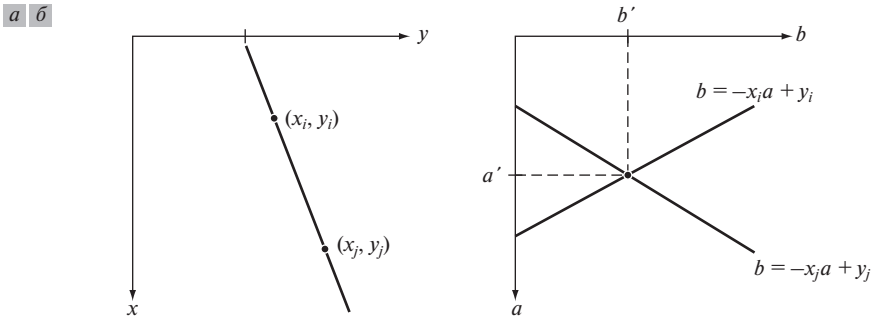


Рис. 10.31. (*а*) Плоскость xy. (*б*) Пространство параметров

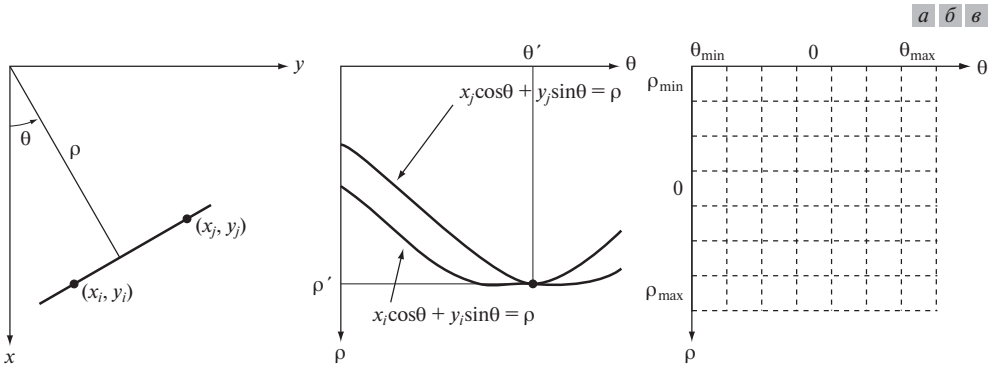


Рис. 10.32. (*а*) Представление прямой на плоскости *xy* с помощью параметров нормали (ρ_p, θ_q). (*б*) Синусоиды на плоскости ρθ; точка пересечения (ρ′, θ′) соответствует прямой, проходящей через точки (x_i, y_i) и (x_j, y_j) на плоскости *xy*. (*в*) Разбиение плоскости ρθ на ячейки накопления

ресечения с осью *y*, а $\theta = 90°$ при $\rho \geq 0$ или $\theta = -90°$ при $\rho < 0$. Каждая синусоида на рис. 10.32(*б*) представляет семейство прямых, проходящих через конкретную точку (x_k, y_k) на плоскости *xy*. Точка пересечения (ρ′, θ′) на рис. 10.32(*б*) соответствует прямой, проходящей через точки (x_i, y_i) и (x_j, y_j) на рис. 10.32(*а*).

Привлекательность преобразования Хафа с точки зрения вычислений проистекает из возможности разбиения пространства параметров ρθ на так называемые *ячейки накопления*, как показано на рис. 10.32(*в*), где (ρ_{min}, ρ_{max}) и (θ_{min}, θ_{max}) — предполагаемые диапазоны возможных значений параметров: $-90° \leq \theta \leq 90°$ и $-D \leq \rho \leq D$, где D — расстояние между углами изображения по диагонали. В ячейке с координатами (p, q) накапливается значение $A(p, q)$ для прямоугольника в пространстве параметров, соответствующего точке (ρ_p, θ_q). Первоначально значения во всех ячейках накопления равны нулю. Затем для каждой точки (x_k, y_k) из заданного множества точек в плоскости *xy* полагаем параметр θ равным поочередно каждому разрешенному дискретному значению на оси θ и находим соответствующее ему значение ρ, решая уравнение $\rho = x_k\cos\theta + y_k\sin\theta$. После этого найденное значение ρ округляется до ближайшего разрешенного дискретного значения на оси ρ. Если выбор значения θ_q приводит к решению ρ_p, увеличиваем накопленное значение в соответствующей ячейке: $A(p, q) = A(p, q) + 1$. После выполнения описанной процедуры для всех анализируемых точек (x_i, y_i) записанное в ячейке (p, q) значение $A(p, q) = P$ означает, что в плоскости *xy* имеется P точек, лежащих на прямой $\rho_p = x\cos\theta_q + y\sin\theta_q$. Точность попадания точек на эту прямую определяется размерами ячеек накопления на плоскости ρθ. Можно показать (задача 10.24), что вычислительная сложность изложенного метода линейна по n, числу точек в заданном множестве на плоскости *xy*.

Пример 10.13. Иллюстрация основных свойств преобразования Хафа.

■ Рис. 10.33 иллюстрирует выполнение преобразования Хафа, основанного на уравнении (10.2-38). На рис. 10.33(*а*) показано изображение размерами 101×101 пиксель, на котором отмечены пять точек. Для каждой из них построен образ (отрезок синусоиды) в плоскости ρθ с дискретными целочисленными отсчетами по обеим осям, как показано на рис. 10.33(*б*). Диапазон значений θ составляет

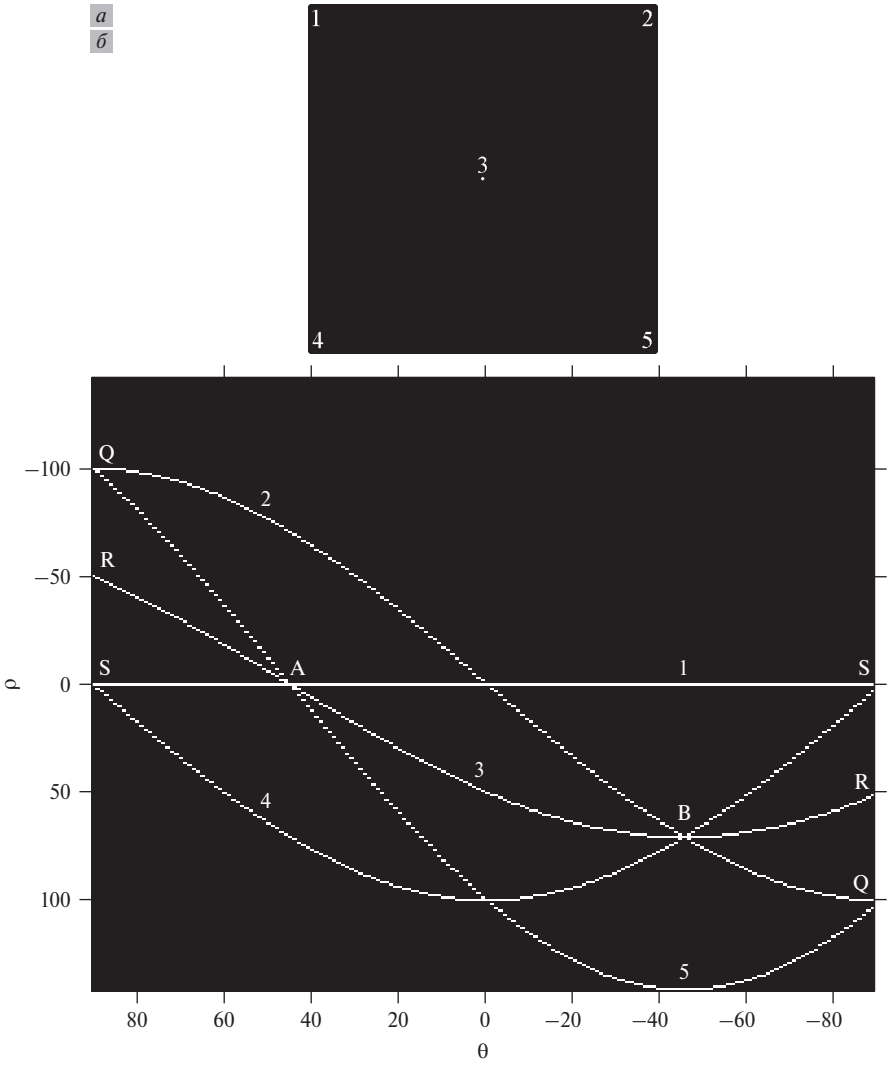


Рис. 10.33. (*а*) Изображение размерами 101×101 пиксель с пятью отмеченными точками (точки увеличены для наглядности). (*б*) Соответствующее пространство параметров

[–90°, +90°], а диапазон значений ρ равен $[-\sqrt{2}D, \sqrt{2}D]$, где D — расстояние между углами изображения по диагонали. Как видно из рис. 10.33(*б*), каждая из синусоид имеет свою амплитуду и фазу. Горизонтальная линия, полученная при отображении точки 1, представляет собой синусоиду с нулевой амплитудой.

Точки, обозначенные *A* и *B* на рис. 10.33(*б*), демонстрируют свойство преобразования Хафа обнаруживать точки, лежащие на одной прямой. В точке *A* пересекаются кривые, которые соответствуют точкам 1, 3 и 5 в плоскости *xy* исходного изображения. Координаты точки *A* показывают, что упомянутые три точки лежат на прямой линии, проходящей через начало координат (ρ = 0) под углом 45° (см. рис. 10.32(*а*)). Аналогично пересечение кривых в точке B простран-

ства параметров указывает, что точки 2, 3 и 4 лежат на прямой, проходящей под углом $-45°$ на расстоянии от начала координат $\rho = 71$ (округленная до ближайшего целого значения половина длины диагонали от начала координат к правому нижнему углу изображения). Наконец, точки *Q*, *R* и *S* на рис. 10.33(*б*) иллюстрируют тот факт, что используемое преобразование Хафа на левом и правом краях области значений в пространстве параметров ведет себя зеркально симметричным образом. Это свойство является следствием того, как меняются знаки параметров θ и ρ на границах $\pm 90°$. ■

Хотя до сих пор в центре внимания были прямые линии на изображении, преобразование Хафа можно применить к любой функции вида $g(\mathbf{v}, \mathbf{c}) = 0$, где $\mathbf{v}$ — вектор координат, а $\mathbf{c}$ — вектор коэффициентов. Например, точки, лежащие на окружности,

$$(x - c_1)^2 + (y - c_2)^2 = c_3^2, \tag{10.2-39}$$

также могут обнаруживаться описанным выше методом. Основное отличие состоит в увеличении числа параметров до трех (c_1, c_2 и c_3), что приводит к трехмерному пространству параметров с кубическими ячейками, накапливаемые значения в которых имеют вид $A(p, q, r)$. Процедура состоит в последовательном присваивании параметрам c_1 и c_2 всех сочетаний допустимых дискретных значений, нахождением для каждой пары значения c_3, которое бы удовлетворяло уравнению (10.2-39), и увеличением накопленного значения в ячейке, отвечающей тройке (c_1, c_2, c_3). Ясно, что сложность преобразования Хафа пропорциональна числу координат и коэффициентов в данном функциональном представлении[13]. Возможны дальнейшие обобщения преобразования Хафа на случаи обнаружения кривых, не имеющих простого аналитического представления, как в случае применения этого преобразования к полутоновым изображениям. Некоторые ссылки, относящиеся к подобным расширениям, приводятся в конце данной главы.

Вернемся теперь к задаче связывания контуров. Подход, основанный на преобразовании Хафа, заключается в следующем.

1. Формируется *двоичное* изображение контуров с помощью одного из методов, рассмотренных выше в данном разделе.
2. Выполняется разбиение (дискретизация) пространства параметров $\rho\theta$ на ячейки накопления.
3. Для всех ненулевых пикселей двоичного изображения, полученного в п. 1, находятся образы в пространстве параметров $\rho\theta$ и осуществляется процедура накопления[14].
4. Анализируются накопленные значения и отыскиваются ячейки с наибольшей концентрацией точек.
5. Исследуются отношения между пикселями изображения, отвечающими выбранным ячейкам накопления (в основном на предмет их связности).

Понятие связности в данном случае обычно базируется на вычислении расстояний между несвязными пикселями, соответствующими данной ячейке

---

[13] Можно показать, что сложность преобразования Хафа пропорциональна nK^{P-1}, где P — число параметров, а K — число дискретных значений (одинаковое) для каждого из параметров. — *Прим. перев.*

[14] Данный пункт добавлен при переводе. — *Прим. ред. перев.*

накопления. Разрыв в линии, отвечающей данной ячейке, заполняется, если длина промежутка меньше заданного порога. Отметим, что сама возможность группировки линий по их направлению является *глобальным* принципом, применяемым ко всему изображению, для чего требуется проверять только пиксели, соответствующие определенной ячейке накопления. Это значительное преимущество, если сравнивать с методами, которые рассматривались в двух предыдущих разделах. Данную идею иллюстрирует следующий пример.

Пример 10.14. Использование преобразования Хафа для связывания контуров.

■ На рис. 10.34(*а*) представлен аэрофотоснимок аэропорта. Цель данного примера состоит в применении преобразования Хафа для выделения границ главной взлетно-посадочной полосы. Решение подобной задачи может представлять интерес, например, для систем автономной навигации летательных аппаратов.

Первый шаг состоит в формировании контурного изображения. Рис. 10.34(*б*) демонстрирует изображение контуров, построенное алгоритмом Кэнни с помощью тех же параметров и процедуры, что и в примере 10.9. Для целей вычисления преобразования Хафа аналогичные результаты можно получить с помощью любого из методов выделения контуров, рассмотренных в разделе 10.2.5 или 10.2.6. На рис. 10.34(*в*) представлено пространство параметров Хафа, полученное с дискретностью 1° по θ и 1 пиксель по ρ.

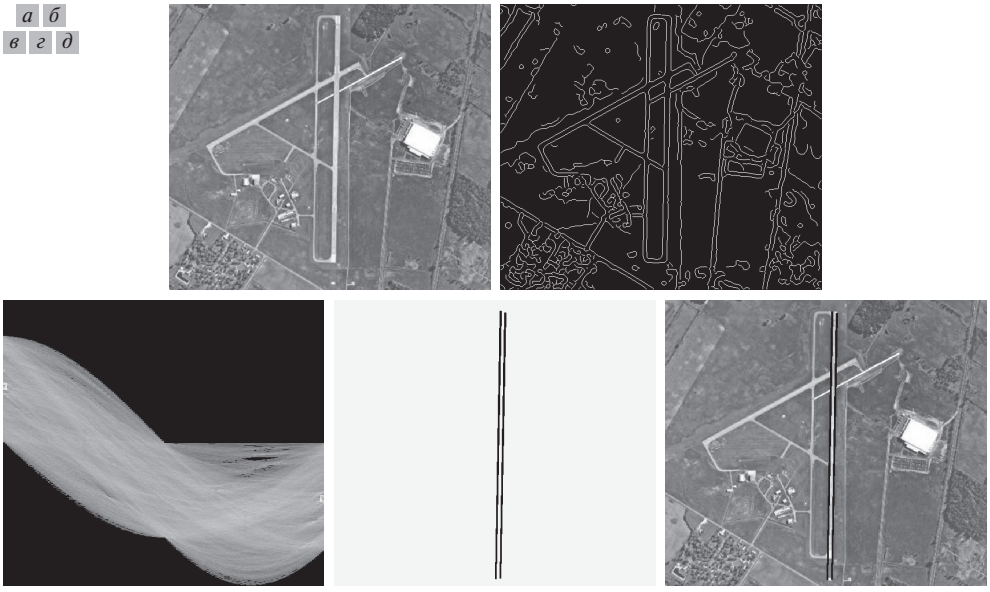


Рис. 10.34. (*а*) Аэрофотоснимок аэропорта размерами 502×564 пикселей. (*б*) Контурное изображение, полученное с помощью алгоритма Кэнни. (*в*) Пространство параметров Хафа (прямоугольниками отмечены точки, связанные с длинными вертикальными линиями). (*г*) Линии в плоскости изображения, соответствующие выделенным прямоугольникам. (*д*) Те же линии, наложенные на исходное изображение

Интересующая взлетно-посадочная полоса на изображении отклоняется приблизительно на 1° от вертикального направления. В связи с этим выбираем ячейки, соответствующие углам нормали ±90° и содержащие наибольшие накопленные значения, поскольку полосе отвечают самые длинные линии такого направления. Небольшие белые прямоугольники справа и слева на краях рис. 10.34(*в*) отмечают эти ячейки. Как отмечалось выше в связи с рис. 10.33(*б*), преобразование Хафа демонстрирует смежность на краях. Другая интерпретация данного свойства состоит в том, что прямые линии с направлением нормали +90° и –90° эквивалентны (т. е. в обоих случаях имеют вертикальную ориентацию). На рис. 10.34(*г*) показаны линии, соответствующие двум упомянутым ячейкам накопления, а на рис. 10.34(*д*) эти линии наложены на исходное изображение. При построении этих линий соединялись любые разрывы с длиной, не превышающей 20% высоты изображения (приблизительно 100 пикселей). Ясно видно, что найденные линии отвечают краям интересующей взлетно-посадочной полосы.

Заметим, что единственная существенная информация, потребовавшаяся для решения данной задачи, — это ориентация взлетно-посадочной полосы и ориентация наблюдателя относительно нее. Другими словами, автономная система навигации летательного аппарата должна знать, что если полоса ориентирована на север и направление полета также на север, то полоса будет расположена вертикально на изображении. Другие взаимные ориентации обрабатываются аналогичным образом. Направления взлетно-посадочных полос в аэропортах по всему миру указываются на полетных картах, а направление полета легко определяется по данным системы глобального позиционирования (GPS). Эта информация также может использоваться для расчета расстояния до полосы, что позволит оценить параметры (например ожидаемую длину линий относительно размеров изображения), как мы и делали в данном примере. ■

10.3. Пороговая обработка

Пороговые преобразования занимают центральное место в прикладных задачах сегментации изображений благодаря интуитивно понятным свойствам, простоте реализации и скорости вычислений. Пороговое преобразование в первый раз упоминалось в разделе 3.1.1, и в последующих главах мы использовали его в самых различных обсуждениях. В этом разделе дается более формальное определение порогового преобразования и на его основе строятся значительно более общие методы пороговой обработки, чем излагались до сих пор.

10.3.1. Обоснование

В предыдущем разделе идентификация областей выполнялась в два этапа: сначала находились отрезки контуров, а затем делалась попытка связать эти отрезки в протяженные границы. В данном разделе мы рассмотрим методы прямого разбиения изображений на области, опираясь на сами значения яркости или их свойства.

Основы порогового преобразования яркости

Предположим, что показанная на рис. 10.35(*а*) гистограмма соответствует некоторому изображению $f(x, y)$, содержащему светлые объекты на темном фоне, так что яркости пикселей объекта и фона сосредоточены вблизи двух преобладающих значений. Очевидный способ выделения объектов из окружающего фона состоит в выборе значения порога T, разграничивающего моды распределения яркостей. Тогда любая точка (x, y), в которой $f(x, y) > T$, называется *точкой объекта*, а в противном случае — *точкой фона*. Другими словами, сегментированное изображение $g(x, y)$ задается соотношением

$$g(x,y) = \begin{cases} 1, & \text{если } f(x,y) > T \\ 0, & \text{если } f(x,y) \le T. \end{cases} \qquad (10.3\text{-}1)$$

Хотя мы следуем общепринятой традиции, используя значение яркости 0 для пикселей фона и 1 для пикселей объекта, в формуле (10.3-1) можно использовать любые два отличающихся значения.

Если значение T есть константа, применяемая для всего изображения, то эта формула описывает так называемое *глобальное пороговое преобразование*. Если величина порога не постоянна на изображении, говорят о *преобразовании с переменным порогом*. Говоря о пороговом преобразовании, иногда используют термины *«локальное»* или «*в окрестности*», чтобы указать, что величина переменного порога T в любой точке изображения (x, y) зависит от свойств окрестности (x, y) (например от средней яркости пикселей в этой окрестности). Если порог T зависит от характеристик обрабатываемого изображения и меняется с изменением пространственных координат x и y, то такое преобразование с переменным порогом часто называют *динамическим* или *адаптивным*. Использование этих терминов не является общепринятым, и в литературе по цифровой обработке изображений их можно встретить и в ином значении.

Рис. 10.35(*б*) демонстрирует более сложный случай выбора порога, когда гистограмма изображения характеризуется наличием трех мод распределения (например если на темном фоне изображения имеются два вида светлых объектов). Здесь с помощью *многоуровневого порогового преобразования* точка (x, y) классифицируется как принадлежащая фону, если $f(x, y) \le T_1$, объекту одного

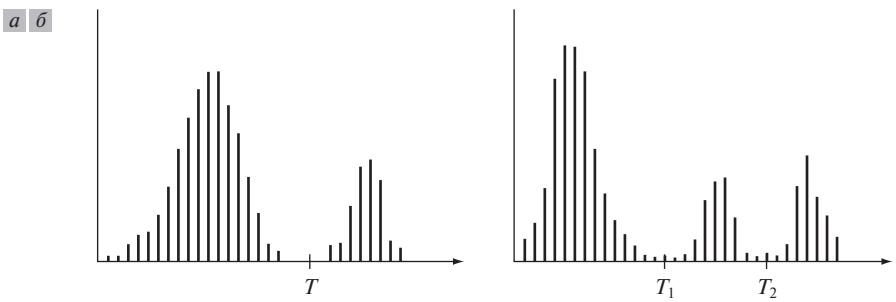


Рис. 10.35. Гистограммы яркости, допускающие разделение с помощью (*а*) одного порога; (*б*) двух порогов

класса, если $T_1 < f(x,y) \le T_2$, и объекту другого класса, если $f(x,y) > T_2$. То есть сегментированное изображение задается соотношением

$$g(x,y) = \begin{cases} a & \text{если } f(x,y) > T_2 \\ b & \text{если } T_1 < f(x,y) \le T_2 \\ c & \text{если } f(x,y) \le T_1, \end{cases} \qquad (10.3\text{-}2)$$

где a, b, c — любые три различных значения яркости. Пороговое преобразование с двумя порогами будет рассматриваться в разделе 10.3.6. Задачи сегментации, требующие применения более двух порогов, трудно поддаются решению (а часто вообще не решаются), и к лучшим результатам обычно приводят другие методы, такие как обработка с переменным порогом, обсуждаемая в разделе 10.3.7, или выращивание областей, обсуждаемое в разделе 10.4.

Исходя из обсуждения выше, можно сделать интуитивный вывод, что успех порогового преобразования напрямую связан с шириной и глубиной впадин между модами распределения яркости. В свою очередь основные факторы, влияющие на характеристики впадин, это: (1) расстояние между пиками на гистограмме (чем оно больше, тем больше шансов разделить моды); (2) уровень шума в изображении (по мере увеличения шума моды распределения становятся шире); (3) соотношение размеров объектов и области фона; (4) равномерность освещения и (5) однородность коэффициента отражения объектов и фона.

Влияние шума при пороговой обработке

В качестве иллюстрации того, как наличие шума влияет на гистограмму изображения, рассмотрим рис. 10.36(*а*). Это простое синтезированное изображение

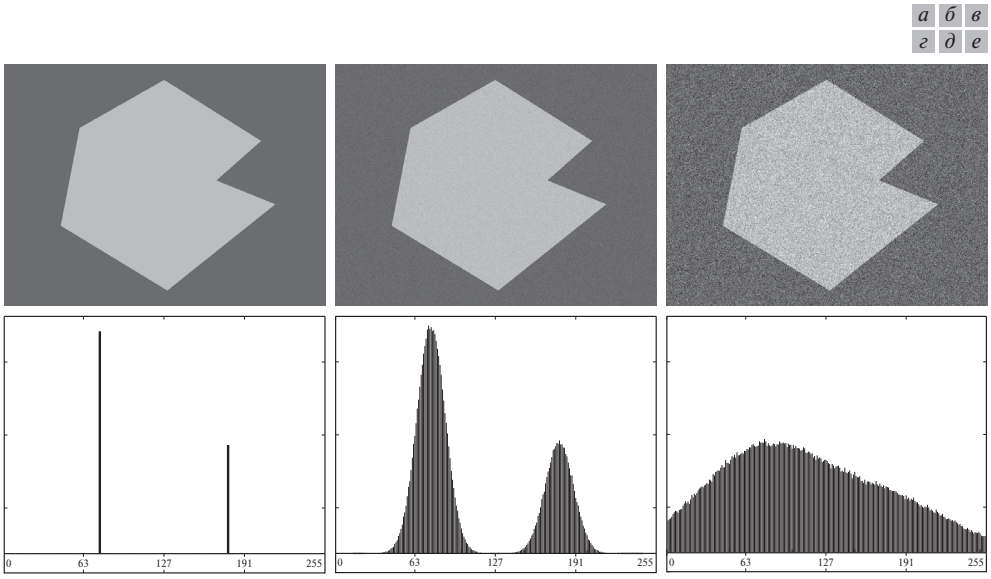


Рис. 10.36. (*а*) 8-битовое изображение, не содержащее шума. (*б*) Изображение с аддитивным гауссовым шумом, имеющим среднее 0 и стандартное отклонение 10 градаций яркости. (*в*) Изображение с аддитивным гауссовым шумом, имеющим среднее 0 и стандартное отклонение 50 градаций яркости. (*г*)—(*е*) Соответствующие гистограммы

не содержит шума, поэтому его гистограмма состоит из двух мод в форме точечных импульсов, как показывает рис. 10.36(*г*). Сегментация такого изображения на две области является тривиальной задачей, которая решается установкой порога в любой точке между модами. На рис. 10.36(*б*) показано исходное изображение, искаженное гауссовым шумом с нулевым средним и стандартным отклонением величиной 10 градаций яркости. Хотя на соответствующей гистограмме (рис. 10.36(*д*)) моды стали шире, расстояние между ними достаточно велико, и глубина впадины на гистограмме позволяет легко разграничить моды. Задача сегментации такого изображения решается, если поместить порог посередине между пиками. На рис. 10.36(*в*) представлен результат искажения исходного изображения гауссовым шумом с нулевым средним и стандартным отклонением величиной 50 градаций яркости. Как показывает гистограмма на рис. 10.36(*е*), ситуация значительно усложнилась, так что не видно способа разграничить две моды. Без дополнительной обработки (например, с помощью методов, рассматриваемых в разделах 10.3.4 и 10.3.5) трудно найти подходящий порог для сегментации данного изображения.

Роль освещения и отражения

Рис. 10.37 иллюстрирует влияние освещения на гистограмму изображения. На рис. 10.37(*а*) опять приведено зашумленное изображение, ранее показанное на рис. 10.36(*б*), а на рис. 10.37(*г*) — его гистограмма. Это изображение легко сегментируется с помощью одиночного порога. Можно проиллюстрировать влияние неравномерного освещения, умножая исходное изображение на некоторую функцию изменения яркости, например линейную, как на рис. 10.37(*б*) (ее гистограмма представлена на рис. 10.37(*д*)). При этом будет получено изображение, показанное на рис. 10.37(*в*). Как видно из его гистограммы (рис. 10.37(*е*)), прежняя

а *б* *в*
г *д* *е*

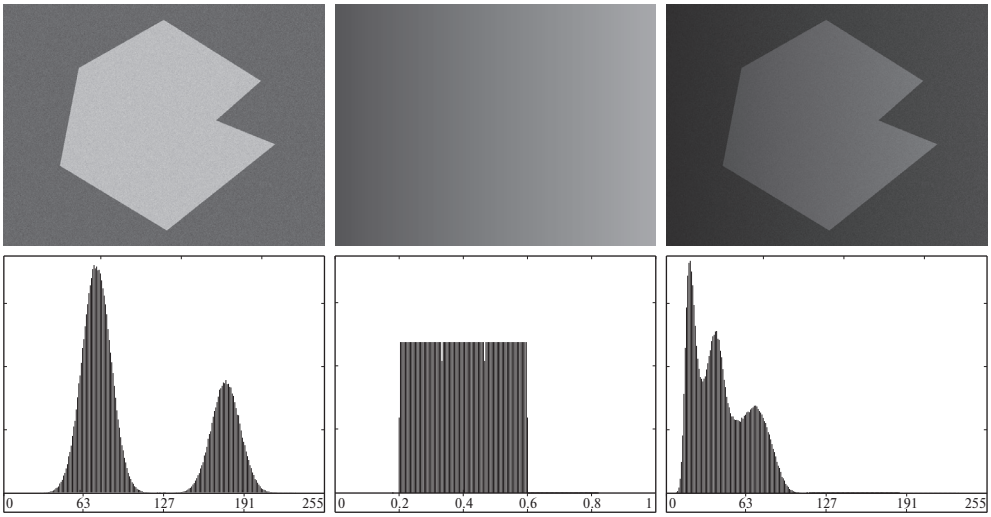


Рис. 10.37. (*а*) Зашумленное изображение. (*б*) Изображение с линейным изменением яркости в диапазоне [0,2, 0,6]. (*в*) Произведение (*а*) и (*б*). (*г*)—(*е*) Соответствующие гистограммы

глубокая впадина исказилась до такой степени, что разграничение двух мод стало невозможным без дополнительной обработки (см. разделы 10.3.4 и 10.3.5). К аналогичным результатам приводят ситуации, когда функция освещения идеально равномерная, а функция отражения — нет, например, вследствие естественных отклонений коэффициента отражения поверхности объектов и/или фона.

> Теоретически гистограмма изображения с линейно изменяющейся по горизонтали яркостью должна быть равномерной. На практике равномерность гистограммы зависит от размеров изображения и яркостного разрешения. Например, изображение размерами 256×256 с линейно изменяющейся яркостью, квантованной на 256 градаций, будет иметь равномерную гистограмму, а аналогичное изображение с размерами 256×257 — нет.

Ключевой пункт предыдущего абзаца — это то, что функции освещения и отражения играют центральную роль в успехе сегментации изображения с использованием пороговых преобразований и других методов. Поэтому в качестве первого шага на пути решения задачи сегментации следует рассматривать управление указанными факторами, когда это возможно. Если эти факторы находятся вне контроля, для решения задачи сегментации существуют три основных подхода. Первый состоит в непосредственной компенсации неравномерности освещения. Например, неравномерное (но постоянное) освещение можно скорректировать путем деления исходного изображения на полученное в тех же условиях изображение отражательной поверхности ровного белого цвета[15]. При втором подходе пытаются добиться глобальной коррекции неравномерности путем обработки изображения с помощью, например, преобразования «выступ» из раздела 9.6.3. Третий подход состоит в том, чтобы обойти неравномерность, применяя преобразование с переменным порогом, как описано в разделе 10.3.7.

10.3.2. Обработка с глобальным порогом

Как отмечалось в предыдущем разделе, если распределения яркости пикселей объектов и фона достаточно хорошо разделяются, можно применить единый (*глобальный*) порог для всего изображения. В большинстве практических задач изображения характеризуются значительной изменчивостью, поэтому даже в случае возможности обработки с глобальным порогом необходим алгоритм, позволяющий автоматически выбрать значение порога для каждого конкретного изображения. Для этой цели может применяться следующий итерационный алгоритм:

1. Выбирается некоторая начальная оценка значения порога T.
2. Проводится сегментация изображения согласно (10.3-1) с помощью порога T. В результате образуются две группы пикселей: G_1, состоящая из пикселей с яркостью больше T, и G_2, состоящая из пикселей с яркостью меньше или равной T.

---

[15] Отметим здесь еще раз, что если в преобразованиях участвуют не сами яркости в своем точном физическом понимании, а их логарифмированные значения (как это в большинстве случаев и происходит), то операцию деления следует заменить операцией вычитания. — *Прим. перев.*

3. Вычисляются значения m_1 и m_2 средних яркостей пикселей по областям G_1 и G_2 соответственно.
4. Вычисляется новое значение порога:

$$T = \frac{1}{2}(m_1 + m_2).$$

5. Повторяются шаги со 2-го по 4-й до тех пор, пока разница значений T при соседних итерациях не окажется меньше значения наперед заданного параметра ΔT.

Этот простой алгоритм хорошо работает в ситуациях, когда на гистограмме имеется ярко выраженная впадина между модами, которые соответствуют объекту и фону. Параметр ΔT используется для управления числом итераций алгоритма в ситуациях, когда важным соображением является скорость вычислений. В общем случае чем больше значение ΔT, тем меньше итераций алгоритма потребуется. Начальное значение порога T должно выбираться между минимальным и максимальным значениями яркости на изображении (задача 10.28). Хорошим начальным приближением для T является средний уровень яркости изображения.

Пример 10.15. Глобальная пороговая обработка.

■ На рис. 10.38 демонстрируется пример сегментации по порогу, вычисленному вышеописанным алгоритмом. На рис. 10.38(*а*) приведено исходное изображение, а на рис. 10.38(*б*) — его гистограмма, имеющая ярко выраженную впадину. В результате применения итерационного алгоритма, начиная со среднего значения яркости и параметра $\Delta T = 0$, после трех итераций было найдено значение порога 125,4. Результат сегментации исходного изображения с использованием порога $T = 125$ приводится на рис. 10.38(*в*). Как и следовало ожидать, благодаря четко разделяющимся модам гистограммы, сегментация объектов и фона оказывается весьма эффективной. ■

а б в

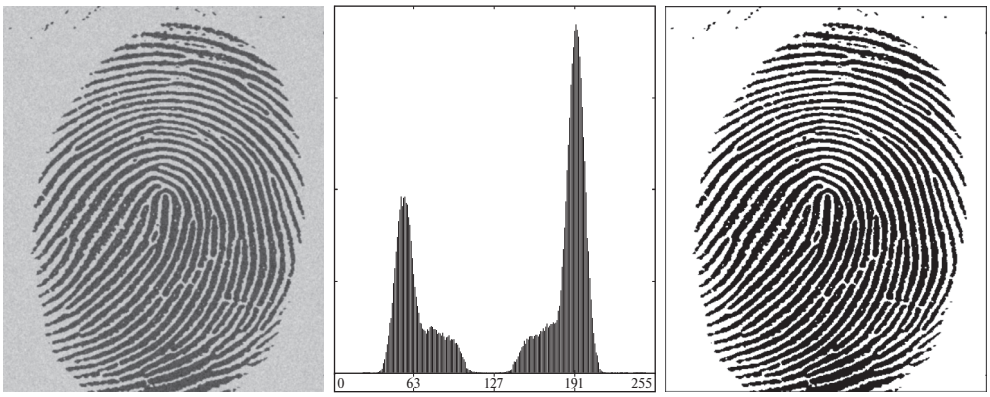


Рис. 10.38. (*а*) Зашумленное изображение отпечатка пальца. (*б*) Гистограмма изображения. (*в*) Результат сегментации с порогом, полученным итерационным методом (рамка добавлена для наглядности). (Исходное изображение предоставлено Национальным институтом стандартов и технологии США)

Чтобы легче было понять вышеизложенный алгоритм, он формулировался в терминах последовательного порогового преобразования исходного изображения с вычислением на каждом шаге новых средних значений. Однако можно предложить более эффективную процедуру, в которой все операции выполняются по гистограмме изображения, вычисляемой только один раз (задача 10.26).

10.3.3. Метод Оцу оптимального глобального порогового преобразования

Пороговую обработку можно рассматривать как задачу теории статистических решений, в которой цель состоит в минимизации средней ошибки разбиения пикселей на две или более группы (также называемые *классами*). Как известно, эта задача имеет элегантное решение в замкнутом виде, известное как *байесовское решающее правило* (см. раздел 12.2.2). Данное решение строится на основе всего двух параметров: функции плотности вероятности значений яркости каждого класса и априорной вероятности появления пикселей каждого класса в рассматриваемой прикладной задаче. К сожалению, оценка распределения представляет трудность, и задачу обычно упрощают, делая рабочие предположения о форме функций плотности вероятности, например считая их гауссовыми функциями. Даже с подобными упрощениями процедура нахождения решения может оказаться сложной и не всегда подходящей для практических целей.

Рассматриваемый в данном разделе *метод Оцу* [Otsu, 1979] предлагает привлекательную альтернативу. Этот метод является оптимальным в том смысле, что он максимизирует *межклассовую дисперсию* — хорошо известную характеристику, применяемую в статистическом дискриминантном анализе. Основная идея состоит в том, что если классы хорошо разделяются по порогу, они должны отличаться значениями яркости своих пикселей, и наоборот, порог, который хорошо разграничивает классы по яркости, будет наилучшим. Помимо оптимальности, метод Оцу имеет то важное свойство, что он целиком основан на вычислениях, выполняемых над гистограммой изображения, — легко вычисляемым одномерным массивом.

Обозначим {0, 1, 2,..., $L-1$} L различаемых градаций яркости цифрового изображения размерами $M \times N$ пикселей, и пусть n_i обозначает число пикселей с яркостью i. Общее число пикселей в изображении $MN = n_0 + n_1 + n_2 + ... + n_{L-1}$. Нормированная гистограмма (см. раздел 3.3) состоит из компонент $p_i = n_i / MN$, откуда следует, что

$$\sum_{i=0}^{L-1} p_i = 1, \; p_i \geq 0 . \qquad (10.3\text{-}3)$$

Предположим теперь, что выбран порог $T(k) = k$, $0 \leq k < L-1$, с помощью которого исходное изображение разбивается на два класса C_1 и C_2, такие, что C_1 состоит из всех пикселей изображения, имеющих значения яркости в интервале $[0, k]$, а C_2 состоит из пикселей со значениями яркости в интервале $[k+1, L-1]$. При таком пороге вероятность $P_1(k)$ того, что некоторый пиксель будет отнесен к классу C_1, задается накопленной суммой

$$P_1(k) = \sum_{i=0}^{k} p_i \,. \tag{10.3-4}$$

Иначе говоря, это вероятность появления класса C_1. Аналогично вероятность появления класса C_2 задается формулой

$$P_2(k) = \sum_{i=k+1}^{L-1} p_i = 1 - P_1(k). \tag{10.3-5}$$

Согласно формуле (3.3-18) среднее значение яркости пикселей, относящихся к классу C_1, равно

$$m_1(k) = \sum_{i=0}^{k} iP(i \mid C_1) = \sum_{i=0}^{k} iP(C_1 \mid i)P(i) / P(C_1) = \frac{1}{P_1(k)} \sum_{i=0}^{k} ip_i \,, \tag{10.3-6}$$

где $P_1(k)$ задано соотношением (10.3-4). Член $P(i|C_1)$ в первом равенстве (10.3-6) есть вероятность значения i при условии, что i относится к классу C_1. Второе равенство следует из формулы Байеса:

$$P(A \mid B) = P(B \mid A)P(A)/P(B).$$

Третье равенство вытекает из того факта, что $P(C_1|i)$, вероятность C_1 при условии i, равна 1, поскольку здесь мы имеем дело только с i из класса C_1. Кроме того, $P(i)$ есть вероятность значения яркости i, т. е. просто i-я компонента гистограммы, выше обозначенная p_i. Наконец, $P(C_1)$ есть вероятность класса C_1, которая равна $P_1(k)$ в соответствии с соотношением (10.3-4).

Аналогичным образом среднее значение яркости пикселей, относящихся к классу C_2, равно

$$m_2(k) = \sum_{i=k+1}^{L-1} iP(i \mid C_2) = \frac{1}{P_2(k)} \sum_{i=k+1}^{L-1} ip_i \,. \tag{10.3-7}$$

Накопленная сумма до уровня яркости k включительно задается соотношением

$$m(k) = \sum_{i=0}^{k} ip_i \,, \tag{10.3-8}$$

а средняя яркость по всему изображению (т. е. *глобальное* среднее значение) равна

$$m_G = \sum_{i=0}^{L-1} ip_i \,. \tag{10.3-9}$$

Справедливость следующих двух формул можно проверить прямой подстановкой вычисленных выше результатов:

$$P_1 m_1 + P_2 m_2 = m_G \tag{10.3-10}$$

и

$$P_1 + P_2 = 1 \,, \tag{10.3-11}$$

где для краткости обозначений всюду временно опущена зависимость от k.

Для оценки «качества» порога уровня k используется нормированный безразмерный показатель

$$\eta = \frac{\sigma_B^2}{\sigma_G^2}, \tag{10.3-12}$$

где σ_G^2 — *глобальная дисперсия* (т. е. дисперсия яркости всех пикселей изображения согласно формуле (3.3-19))

$$\sigma_G^2 = \sum_{i=0}^{L-1} (i - m_G)^2 p_i , \tag{10.3-13}$$

а σ_B^2 — *межклассовая дисперсия*, которая определяется как

$$\sigma_B^2 = P_1 (m_1 - m_G)^2 + P_2 (m_2 - m_G)^2 . \tag{10.3-14}$$

Это равенство также можно переписать в виде

$$\sigma_B^2 = P_1 P_2 (m_1 - m_2)^2 = \frac{(m_G P_1 - m)^2}{P_1 (1 - P_1)}, \tag{10.3-15}$$

> Второй шаг в выражении (10.3-15) имеет смысл только в том случае, если P_1 строго больше 0 и строго меньше 1, откуда с учетом (10.3-11) следует, что P_2 должно удовлетворять тем же условиям.

где выражения для m_G и m выписаны выше — выражения (10.3-9) и (10.3-8). Первое равенство в (10.3-15) вытекает из соотношений (10.3-14), (10.3-10) и (10.3-11). Второе следует из (10.3-5) с учетом (10.3-9). Такая форма записи несколько эффективнее для вычислений, поскольку глобальное среднее значение m_G вычисляется только один раз, так что для каждого значения k необходимо вычислять только два параметра — m и P_1.

Из первой строки равенства (10.3-15) видно, что чем дальше друг от друга два математических ожидания m_1 и m_2, тем больше будет межклассовая дисперсия σ_B^2, которая, таким образом, является мерой *разделимости* классов. Поскольку глобальная дисперсия σ_G^2 есть константа, отсюда следует, что η также является показателем разделимости, максимизация которого эквивалентна максимизации σ_B^2. Задача, таким образом, состоит в определении такого значения порога k, для которого межклассовая дисперсия максимальна, как и было заявлено в начале данного раздела. Заметим, что в формуле (10.3-12) неявно предполагается, что $\sigma_G^2 > 0$. Эта дисперсия может равняться нулю только в том случае, когда все пиксели изображения имеют одну и ту же яркость, что означает наличие только одного класса. В свою очередь, в таком случае и $\sigma_B^2 = 0$, поскольку показатель разделимости одного класса от самого себя равен нулю.

Возвращая назад зависимость от k, получаем окончательные результаты:

$$\eta(k) = \frac{\sigma_B^2(k)}{\sigma_G^2} \tag{10.3-16}$$

и

$$\sigma_B^2(k) = \frac{\left[m_G P_1(k) - m(k)\right]^2}{P_1(k)\left[1 - P_1(k)\right]}. \tag{10.3-17}$$

Тогда оптимальным будет такое значение порога k^*, которое максимизирует $\sigma_B^2(k)$:

$$\sigma_B^2(k^*) = \max_{0 \le k < L-1} \sigma_B^2(k). \tag{10.3-18}$$

Другими словами, для нахождения k^* мы просто оцениваем правую часть равенства (10.3-17) для всех *целых* значений k (таких, что справедливо условие $0 < P_1(k) < 1$) и выбираем то значение k, которое доставляет максимум $\sigma_B^2(k)$. Если максимум достигается более чем при одном k, обычно берут среднее по всем значениям k, для которого межклассовая дисперсия $\sigma_B^2(k)$ максимальна. Можно доказать (задача 10.33), что при соблюдении условия $0 < P_1(k) < 1$ максимум всегда существует. Оценка по формулам (10.3-17) и (10.3-18) для всех значений k является относительно несложной процедурой с вычислительной точки зрения, поскольку k принимает не более чем L целочисленных значений.

После того, как найдено значение k^*, исходное изображение $f(x, y)$ сегментируется, как и прежде:

$$g(x,y) = \begin{cases} 1, & \text{если } f(x,y) > k^* \\ 0, & \text{если } f(x,y) \le k^*, \end{cases} \tag{10.3-19}$$

для $x = 0, 1, 2, ..., M - 1$ и $y = 0, 1, 2, ..., N - 1$. Обратите внимание, что все величины, необходимые для вычисления оценки (10.3-17), находятся по одной лишь гистограмме изображения $f(x, y)$. Помимо оптимального порога, из гистограммы можно извлечь и другую информацию о сегментированном изображении. Например, оценки вероятностей классов при оптимальном значении порога, $P_1(k^*)$ и $P_2(k^*)$, указывают доли площади изображения, занимаемые каждым классом после сегментации. Аналогично математические ожидания $m_1(k^*)$ и $m_2(k^*)$ являются оценками средней яркости классов в исходном изображении.

Значение нормированного показателя η при оптимальном пороге, $\eta(k^*)$, можно использовать для количественной оценки разделимости классов, которая говорит о сложности пороговой обработки данного изображения. Эта характеристика принимает значения в диапазоне

$$0 \le \eta(k^*) \le 1. \tag{10.3-20}$$

> Хотя мы интересуемся величиной η при оптимальном пороге k^*, это неравенство справедливо для любого значения k в интервале $[0, L - 1]$.

Как отмечалось выше, нижняя граница достигается только для изображений с одинаковой яркостью по всему полю. Верхняя граница достигается только для двухградационных изображений с уровнями яркости 0 и $L - 1$ (задача 10.34).

Алгоритм Оцу можно кратко суммировать следующим образом:

1. Вычислить нормированную гистограмму исходного изображения, обозначая ее компоненты p_i, $i = 0, 1, 2, ..., L - 1$.
2. Вычислить накопленные суммы $P_1(k)$ для $k = 0, 1, 2, ..., L - 1$ по формуле (10.3-4).
3. Вычислить накопленные суммы $m(k)$ для $k = 0, 1, 2, ..., L - 1$ по формуле (10.3-8).
4. Вычислить глобальную среднюю яркость m_G по формуле (10.3-9).

5. Вычислить межклассовую дисперсию $\sigma_B^2(k)$ для $k = 0, 1, 2, ..., L - 1$ по формуле (10.3-17).
6. Получить порог Оцу k^* — такое значение k, при котором величина $\sigma_B^2(k)$ максимальна. Если максимум неоднозначен, взять в качестве k^* среднее значение k по всем найденным максимумам.
7. Оценить показатель разделимости η^* путем вычислений по формуле (10.3—16) при $k = k^*$.

Следующий пример иллюстрирует изложенные выше концепции.

Пример 10.16. Обработка с оптимальным глобальным порогом по методу Оцу.

■ На рис. 10.39(*а*) показано изображение клеток-полимеросом, полученное оптическим микроскопом, а на рис. 10.39 представлена его гистограмма. Задача сегментации в этом примере состоит в том, чтобы отделить молекулы от фона. На рис. 10.39(*в*) представлен результат применения простого алгоритма глобаль-

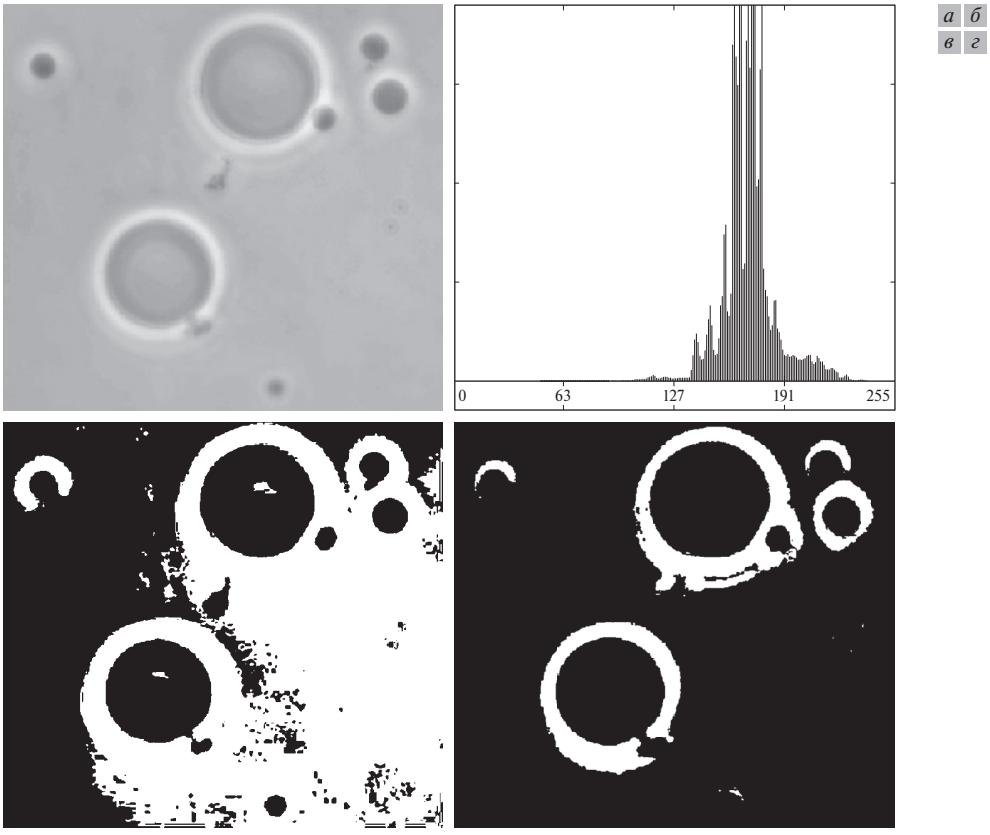


Рис. 10.39. (*а*) Исходное изображение. (*б*) Гистограмма (высокие пики обрезаны, чтобы были видны детали в области меньших значений). (*в*) Результат сегментации с помощью простого алгоритма обработки с глобальным порогом из раздела 10.3.2. (*г*) Результат обработки методом Оцу. (Исходное изображение предоставил проф. Дэниел А. Хаммер, университет Пенсильвании)

ного порогового разделения, описанного в предыдущем разделе. Поскольку на гистограмме нет явно выраженных впадин и различия яркостей объектов и фона небольшие, алгоритм не способен достичь желаемой сегментации. На рис. 10.39(*г*) показаны результаты, полученные методом Оцу. Как легко видеть, они значительно превосходят представленные на рис. 10.39(*в*). Значение порога, вычисленное простым алгоритмом, было 169, тогда как с помощью метода Оцу было найден порог 181, который ближе к яркости светлых областей изображения, соответствующих клеткам. Значение показателя разделимости η составило 0,467.

> Полимеросомы — это искусственные клетки, синтезированные из полимеров. Полимеросомы не вызывают реакции иммунной системы человека и могут применяться, например, для доставки лекарств в необходимые участки тела.

Интересно отметить, что если применить метод Оцу к изображению отпечатка пальца в примере 10.15, будет получено значение порога 125 с показателем разделимости 0,944. Это значение совпадает с тем порогом, который был найден простым алгоритмом (с точностью до округления до ближайшего целого числа). Этого можно было ожидать, учитывая вид гистограммы. Такое высокое значение показателя разделимости обусловлено прежде всего относительно большим расстоянием между модами распределения и наличием глубокой впадины между ними. ■

10.3.4. Применение сглаживания изображения для улучшения обработки с глобальным порогом

Как демонстрировалось на рис. 10.36, с увеличением шума простая задача порогового разделения может превратиться в неразрешимую. Если невозможно снизить уровень шума в исходном изображении и в качестве метода сегментации выбрана пороговая обработка, для улучшения качества результатов часто применяется сглаживание исходного изображения перед сегментацией. Проиллюстрируем этот метод на примере.

На рис. 10.40(*а*) воспроизведено изображение, ранее показанное на рис. 10.36(*в*), а на рис. 10.40(*б*) — его гистограмма. Рис. 10.40(*в*) демонстрирует результат пороговой обработки методом Оцу. Каждая черная точка в белой области и каждая белая точка в черной области — это ошибки порогового разделения, так что данная сегментация является в большой степени неуспешной. На рис. 10.40(*г*) показан результат сглаживания зашумленного изображения с помощью усредняющей маски размерами 5×5 (размеры изображения составляют 651×814 пикселей), а на рис. 10.40(*д*) — его гистограмма. В результате сглаживания форма гистограммы несомненно улучшилась, и можно ожидать, что пороговое преобразование сглаженного изображения даст близкие к идеальным результаты. Рис. 10.40(*е*) подтверждает это. Небольшие искажения формы границы между объектом и фоном на сглаженном сегментированном изображении вызваны размыванием границы. Фактически чем сильнее сглаживается изображение, тем бóльших ошибок на границах следует ожидать при сегментации.

Рассмотрим, что будет происходить, если уменьшать размер области, показанной на рис. 10.40(*а*), по отношению к площади фона. Результатом такого уменьшения является изображение на рис. 10.41(*а*). Шумовая составляющая

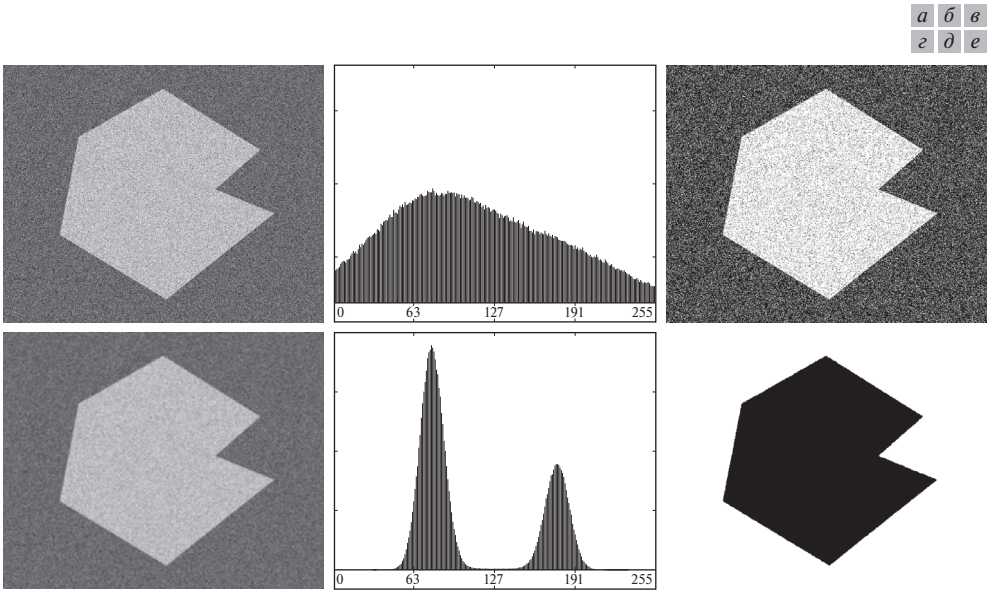


Рис. 10.40. (*а*) Зашумленное изображение, ранее показанное на рис. 10.36(*в*), и (*б*) его гистограмма. (*в*) Результат сегментации методом Оцу. (*г*) Результат сглаживания усредняющей маской 5×5 и (*д*) его гистограмма. (*е*) Результат сегментации методом Оцу

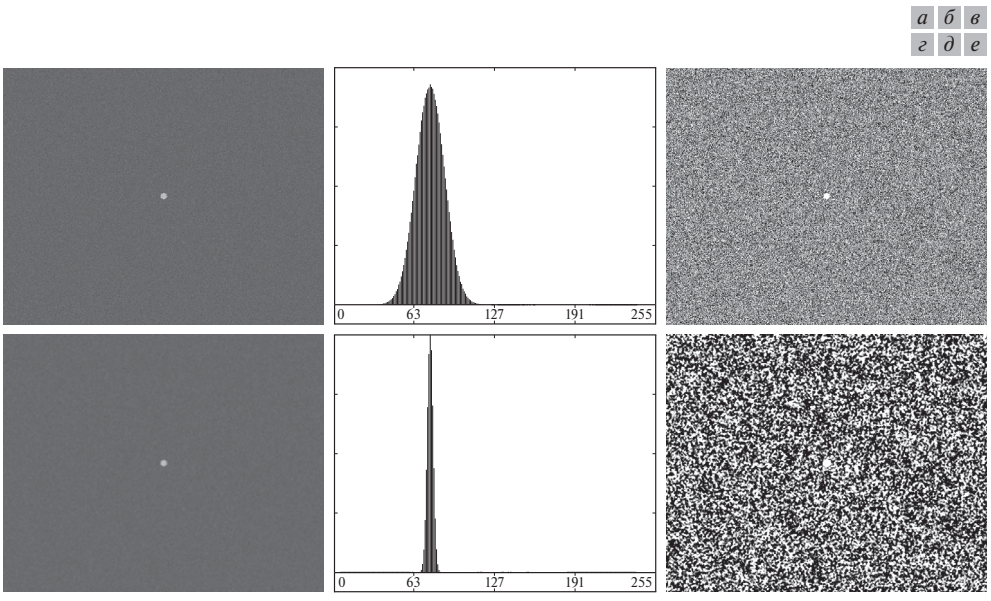


Рис. 10.41. (*а*) Зашумленное изображение и (*б*) его гистограмма. (*в*) Результат сегментации методом Оцу. (*г*) Результат сглаживания усредняющей маской 5×5 и (*д*) его гистограмма. (*е*) Результат сегментации методом Оцу. В обоих случаях сегментация неудовлетворительная

в этом изображении — это аддитивный гауссов шум с нулевым средним и стандартным отклонением 10 градаций яркости (в отличие от 50 в предыдущем примере). Как видно на рис. 10.41(*б*), на гистограмме нет выраженной впадины, так что трудно ожидать успеха сегментации, как и подтверждают результаты на рис. 10.41(*в*). На рис. 10.41(*г*) приведено изображение после сглаживания усредняющей маской размерами 5×5, а на рис. 10.41(*д*) — соответствующая гистограмма. Как и следовало ожидать, общий эффект сглаживания состоит в уменьшении рассеяния на гистограмме, но распределение по-прежнему унимодальное. Как показывает рис. 10.41(*е*), сегментация опять не приводит к успеху. Причина неудачи кроется в том, что объект слишком мал и поэтому его вклад в гистограмму незначителен по сравнению с разбросом яркости из-за шума. В подобных ситуациях более успешных результатов можно ожидать от метода, рассматриваемого в следующем разделе.

10.3.5. Использование контуров для улучшения обработки с глобальным порогом

Из обсуждения, проводившегося на протяжении предшествующих четырех разделов, можно заключить, что шансов на выбор «хорошего» глобального порога значительно больше, если пики на гистограмме являются высокими, узкими, симметричными, а также разделены глубокими впадинами. Один из возможных подходов к «улучшению» формы гистограммы состоит в том, чтобы рассматривать в изображении только те пиксели, которые лежат вблизи перепадов между объектами и фоном либо на самих перепадах. Непосредственное и очевидное улучшение заключается в том, что гистограммы станут меньше зависеть от относительных размеров объектов и областей фона. Например, на гистограмме изображения, состоящего из маленького объекта на фоне с большой площадью (или наоборот), будет доминировать один большой пик, потому что имеется большое число пикселей одного из видов. Мы видели в предыдущем разделе, что это может стать причиной неудачи пороговой обработки.

Если учитывать только пиксели, лежащие на перепаде между объектом и фоном или вблизи перепада, то в получаемой гистограмме будут присутствовать пики примерно равной высоты. К тому же вероятность того, что любой из таких пикселей принадлежит объекту, будет приблизительно равна вероятности его принадлежности фону, что улучшает симметричность пиков гистограммы. Наконец, как будет ясно из следующего абзаца, при использовании пикселей, удовлетворяющих некоторым простым метрическим соотношениям, основанным на операторах градиента и лапласиана, наблюдается тенденция к углублению впадины между пиками гистограммы.

Главная трудность при реализации описанного подхода состоит в неявном предположении, что известны места перепадов между объектами и фоном. Ясно, что во время сегментации этих данных нет, поскольку разделение изображения на объекты и фоновую область как раз и является целью сегментации. Однако, как упоминалось в разделе 10.2, указание на то, находится ли некоторый пиксель на перепаде, можно получить, вычисляя градиент или лапласиан в этой точке. Полученное значение градиента или лапласиана можно использовать как параметр для порогового выбора: брать ли значение пикселя

для последующего анализа или нет. На склоне перепада среднее значение лапласиана равно нулю (см. рис. 10.10) , поэтому такие точки будут отброшены пороговым критерием, а значит, можно ожидать, что интервалы промежуточных значений на гистограммах, которые строятся из пикселей, отбираемых порогом по лапласиану, будут заполнены слабо. Это и означает наличие глубокой впадины — столь желательное свойство, обсуждавшееся выше. На практике использование изображений градиента или лапласиана обычно дает сопоставимые результаты, но последний вариант предпочтительнее, поскольку проще с вычислительной точки зрения и к тому же лапласиан является изотропным детектором перепадов.

Проведенное выше рассмотрение можно кратко суммировать в виде следующего алгоритма обработки исходного изображения $f(x, y)$:

> Этот алгоритм можно модифицировать так, чтобы использовать изображения модуля градиента и абсолютного значения лапласиана совместно. Для этого нужно указать порог отдельно для каждого из этих изображений и формировать дополнительное изображение-маркер с помощью логической функции ИЛИ от двух результатов. Такой подход полезен, если требуется углубленный контроль точек, которые, как предполагается, принадлежат истинным контурам.

1. По исходному изображению $f(x, y)$ вычислить изображение контуров либо как модуль градиента, либо как абсолютное значение лапласиана с помощью любого из методов, изложенных в разделе 10.2.
2. Задать значение порога T.
3. Преобразовать изображение, полученное на шаге 1, по порогу, заданному на шаге 2, в двоичное изображение $g_T(x, y)$. Это изображение будет использоваться на следующем шаге как *маска* для выделения из $f(x, y)$ пикселей, соответствующих отчетливо выраженным контурам.
4. Вычислить гистограмму по тем пикселям $f(x, y)$, которые соответствуют единицам в изображении $g_T(x, y)$.
5. С помощью полученной гистограммы выполнить глобальную сегментацию $f(x, y)$, например методом Оцу.

Если выбрать величину порога T меньше минимального значения контурного изображения, то в соответствии с (10.3-1) изображение $g_T(x, y)$ будет состоять из одних единиц; это означает, что гистограмма будет строиться по всем пикселям изображения $f(x, y)$. В этом случае вышеизложенный алгоритм превращается в глобальное пороговое преобразование, основанное на гистограмме исходного изображения. Обычно устанавливают значение T, соответствующее большому квантилю (например 0,95 и выше), так что в вычислениях участвует лишь небольшая часть пикселей изображения градиента или лапласиана. Этот принцип иллюстрируют следующие примеры, в первом из которых используется градиент, а во втором — лапласиан. В этих примерах оба метода дают похожие результаты. Важным вопросом является построение подходящего изображения производной.

> Для заданной случайной величины x с функцией распределения $\Phi(x)$ квантиль уровня n — это такое значение x_n, для которого $P\{x \le x_n\} = \Phi(x_n) = n$.

Пример 10.17. Использование контурной информации на основе градиента для улучшения обработки с глобальным порогом.

■ На рис. 10.42(*а*) и (*б*) показаны изображение и его гистограмма, ранее приводившиеся на рис. 10.41. Как мы видели, это изображение не удается сегментировать применением сглаживания перед пороговым разделением. Цель данного примера — решить задачу сегментации, используя контурную информацию. На рис. 10.42(*в*) приведено изображение модуля градиента, разделенное по квантилю уровня 0,997. Рис. 10.42(*г*) демонстрирует результат умножения этого изображения-маски на исходное изображение. На рис. 10.42(*д*) показана гистограмма значений ненулевых элементов изображения на рис. 10.42(*г*). Обратите внимание, что эта гистограмма обладает упомянутыми выше важными качествами: на ней имеются относительно симметричные моды, разделенные глубокой впадиной. Таким образом, хотя гистограмма исходного зашумленного изображения не оставляла надежд на успешную бинаризацию, гистограмма на рис. 10.42(*д*) указывает, что выделить небольшой объект из фона изображения с помощью глобального порога действительно возможно. Это подтверждает результат на рис. 10.42(*е*), который был получен методом Оцу, с определением значения порога по гистограмме на рис. 10.42(*д*) и затем применением этого порога глобально к зашумленному изображению на рис. 10.42(*а*). Полученный результат близок к идеальному. ■

Пример 10.18. Использование контурной информации на основе лапласиана для улучшения обработки с глобальным порогом.

■ В этом примере рассматривается более сложная задача порогового разделения. На рис. 10.43(*а*) показано 8-битовое изображение дрожжевых клеток, к которому мы хотим применить преобразование с глобальным порогом, чтобы выделить области, соответствующие ярким пятнам. Для начала на рис. 10.43(*б*) приведена гистограмма, а на рис. 10.43(*в*) — результат применения метода Оцу непосредственно к исходному изображению с такой гистограммой. Как видим, метод Оцу не позволяет достичь желаемой цели, т. е. выделить яркие пятна. Хотя с его помощью удалось изолировать часть областей, соответствующих самим клеткам, некоторые из выделенных областей в правой части изображения слились. Значение порога, вычисленного методом Оцу, равнялось 42 при показателе разделимости, равном 0,636.

На рис. 10.43(*г*) приведено изображение $g_T(x, y)$, полученное вычислением абсолютного значения лапласиана с последующим разделением по порогу $T = 115$ на шкале яркостей [0, 255], который соответствует приблизительно квантилю уровня 0,995 распределения значений пикселей изображения модуля лапласиана. Преобразование с таким порогом дает сильно разреженное множество пикселей, как показывает рис. 10.43(*г*). Обратите внимание, что на этом изображении точки группируются вблизи краев ярких пятен, как того можно было ожидать из рассмотрения выше. На рис. 10.43(*д*) приведена гистограмма ненулевых значений пикселей в произведении изображений (*а*) и (*г*). Наконец, рис. 10.43(*е*) демонстрирует результат глобальной сегментации исходного изображения с помощью порога, найденного методом Оцу по гистограмме на рис. 10.43(*д*). Этот результат согласуется с расположением ярких пятен на исходном изображении. Вычисленное значение порога равнялось 115 при показателе разделимости 0,762, что больше значений, найденных по исходной гистограмме.

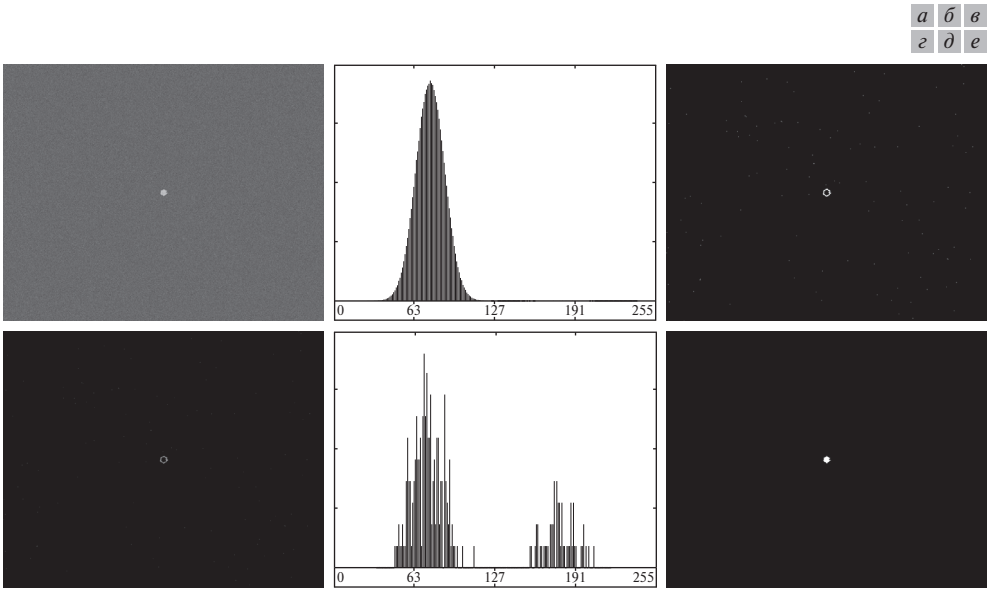


Рис. 10.42. (*а*) Зашумленное изображение, ранее показанное на рис. 10.41(*а*), и (*б*) его гистограмма. (*в*) Изображение модуля градиента после порогового разделения по квантилю уровня 0,997. (*г*) Результат перемножения (*а*) и (*в*). (*д*) Гистограмма ненулевых пикселей изображения (*г*). (*е*) Результат сегментации изображения (*а*) с порогом, найденным методом Оцу по гистограмме (*д*). Порог равен 134, что соответствует приблизительно середине между пиками на этой гистограмме

Варьируя уровень квантиля, на котором устанавливается порог, можно также улучшить и качество сегментации клеточных областей. Например, на рис. 10.44 приведен результат применения той же процедуры со значением порога 55, что составляет приблизительно 5% максимального значения модуля лапласиана и соответствует квантилю уровня 0,539 распределения значений пикселей этого изображения. Очевидно, что такой результат значительно превосходит ранее показанный на рис. 10.43(*в*), где метод Оцу применялся к гистограмме исходного изображения. ■

10.3.6. Обработка с несколькими порогами

До сих пор наше внимание было обращено на сегментацию изображения с использованием одиночного глобального порога. Метод пороговой обработки, изложенный в разделе 10.3.3, допускает обобщение для произвольного числа порогов, поскольку показатель разделимости, на котором он основан, также обобщается для произвольного числа классов [Fukunaga, 1972]. В случае K классов C_1, C_2,..., C_K выражение для межклассовой дисперсии принимает общий вид

$$\sigma_B^2 = \sum_{k=1}^{K} P_k (m_k - m_G)^2 , \qquad (10.3\text{-}21)$$

а б в
г д е

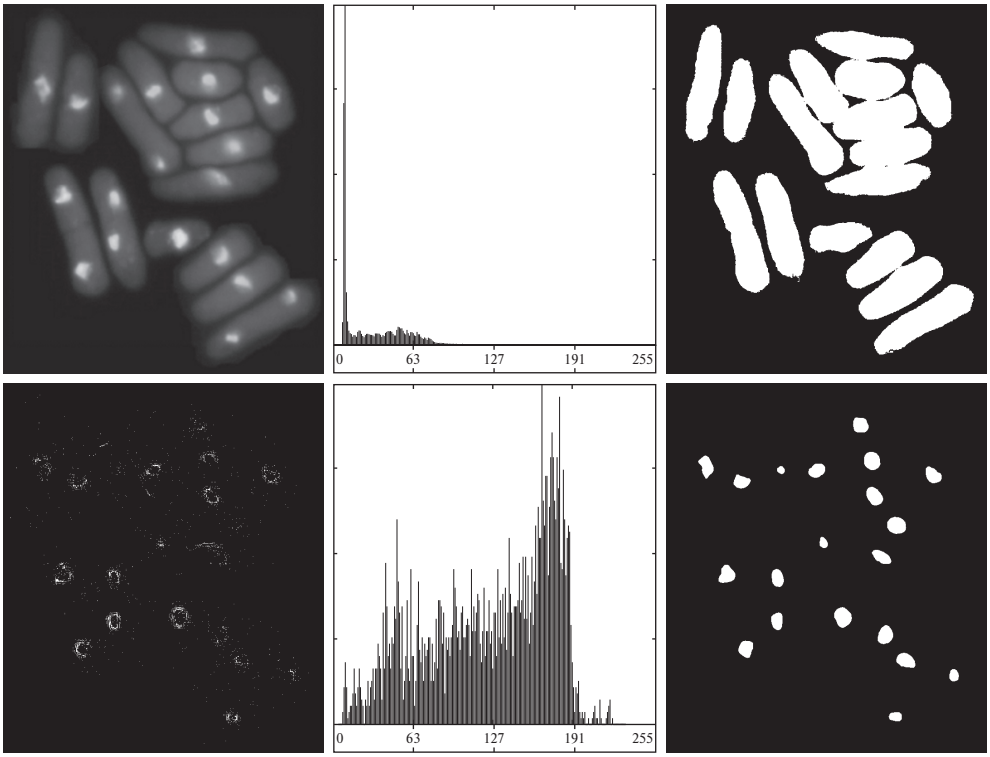


Рис. 10.43. (*а*) Изображение дрожжевых клеток. (*б*) Гистограмма исходного изображения. (*в*) Результат сегментации изображения (*а*) с порогом, найденным методом Оцу по гистограмме (*б*). (*г*) Бинаризованное изображение модуля лапласиана. (*д*) Гистограмма ненулевых значений пикселей произведения изображений (*а*) и (*г*). (*е*) Результат пороговой обработки исходного изображения методом Оцу с использованием гистограммы (*д*). (Исходное изображение предоставила проф. Сьюзен Л. Форсбург, университет Южной Калифорнии)

Рис. 10.44. Результат сегментации изображения на рис. 10.43(*а*) с помощью той же процедуры, что и на рис. 10.43(*г*)—(*е*), но с более низким порогом бинаризации изображения модуля лапласиана

где

$$P_k = \sum_{i \in C_k} p_i \,, \tag{10.3-22}$$

$$m_k = \frac{1}{P_k} \sum_{i \in C_k} i p_i \tag{10.3-23}$$

и m_G — глобальное среднее значение согласно формуле (10.3-9). K классов разделяются $K-1$ порогом, значения которых $k_1^*, k_2^*, \ldots, k_{K-1}^*$ максимизируют величину (10.3-21):

$$\sigma_B^2(k_1^*, k_2^*, \ldots k_{K-1}^*) = \max_{0 \le k_1 < k_2 < \ldots < k_{K-1} < L-1} \sigma_B^2(k_1, k_2, \ldots, k_{K-1}) \,. \tag{10.3-24}$$

Хотя этот результат является совершенно общим, он начинает терять смысл по мере увеличения числа классов, потому что рассматривается только одна переменная (яркость). Фактически межклассовая дисперсия обычно выражается в терминах многочисленных векторных переменных ([Fukunaga, 1972]). На практике подход к решению, предусматривающий использование нескольких глобальных порогов, оправдан лишь в случае, если есть основания полагать, что задачу можно эффективно решить с помощью двух порогов. Прикладные задачи, для решения которых требуется больше двух порогов, обычно решают с привлечением других признаков, а не только значений яркости. Вместо пороговых преобразований используется подход, основанный на дополнительных дескрипторах (например цвете), и прикладная задача формулируется в виде задачи распознавания образов, как рассматривается в разделе 10.3.8.

В случае трех классов, представленных тремя диапазонами яркостей (которые разделяются двумя порогами), межклассовая дисперсия задается соотношением

$$\sigma_B^2 = P_1(m_1 - m_G)^2 + P_2(m_2 - m_G)^2 + P_3(m_3 - m_G)^2 \,, \tag{10.3-25}$$

где

$$P_1 = \sum_{i=0}^{k_1} p_i; \quad P_2 = \sum_{i=k_1+1}^{k_2} p_i; \quad P_3 = \sum_{i=k_2+1}^{L-1} p_i \tag{10.3-26}$$

и

$$m_1 = \frac{1}{P_1} \sum_{i=0}^{k_1} i p_i; \quad m_2 = \frac{1}{P_2} \sum_{i=k_1+1}^{k_2} i p_i; \quad m_3 = \frac{1}{P_3} \sum_{i=k_2+1}^{L-1} i p_i \,. \tag{10.3-27}$$

Подобно равенствам (10.3-10) и (10.3-11), справедливы следующие соотношения:

$$P_1 m_1 + P_2 m_2 + P_3 m_3 = m_G \tag{10.3-28}$$

и

$$P_1 + P_2 + P_3 = 1 \,. \tag{10.3-29}$$

Ясно, что члены P и m (а значит и σ_B^2) являются функциями, зависящими от k_1 и k_2. Два оптимальных порога, k_1^* и k_2^*, доставляют максимум величине $\sigma_B^2(k_1^*, k_2^*)$. Другими словами, как и в случае с одиночным порогом, который рассматривался в разделе 10.3.3, оптимальные пороги определяются путем нахождения

$$\sigma_B^2(k_1^*, k_2^*) = \max_{0 \le k_1 < k_2 < L-1} \sigma_B^2(k_1, k_2). \qquad (10.3\text{-}30)$$

Процедура начинается с выбора первого значения k_1 (принимаем его равным 0, т. к. это соответствует минимально возможному диапазону для первого класса — в одну градацию; также следует помнить, что приращения должны быть целочисленными, поскольку это градации яркости). Затем k_2 пробегает все значения, бóльшие k_1 и меньшие $L - 1$ (т. е. $k_2 = k_1 + 1, ..., L - 2$). После этого устанавливается следующее значение k_1 и снова перебираются все значения $k_2 > k_1$. Описанная процедура повторяется вплоть до значения $k_1 = L - 3$; в результате полного перебора будет построена треугольная матрица значений $\sigma_B^2(k_1, k_2)$. Последним шагом является поиск максимального элемента в этой матрице; отвечающие ему значения порогов k_1, k_2 и будут оптимальными порогами k_1^* и k_2^*. Если имеется несколько максимумов, соответствующие им значения k_1 и k_2 усредняются. После этого выполняется сегментация изображения по правилам

$$g(x,y) = \begin{cases} a & \text{если } f(x,y) \le k_1^* \\ b & \text{если } k_1^* < f(x,y) \le k_2^* \\ c & \text{если } f(x,y) > k_2^*, \end{cases} \qquad (10.3\text{-}31)$$

где a, b, c — три произвольных допустимых значения.

В заключение отметим, что показатель разделимости, введенный в разделе 10.3.3 для одного порога, допускает прямое обобщение на случай нескольких порогов:

$$\eta(k_1^*, k_2^*) = \frac{\sigma_B^2(k_1^*, k_2^*)}{\sigma_G^2}, \qquad (10.3\text{-}32)$$

где σ_G^2 — глобальная дисперсия изображения, вычисляемая по формуле (10.3-13).

Пример 10.19. Обработка с несколькими глобальными порогами.

■ На рис. 10.45(*а*) показано изображение айсберга. Задача сегментации в этом примере состоит в том, чтобы разделить изображение на области трех видов: темный фон, освещенная часть айсберга и часть айсберга, находящаяся в тени. Из ги-

а б в

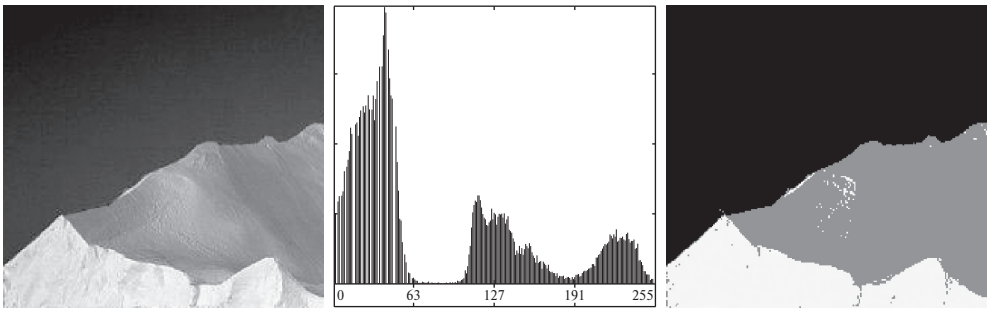


Рис. 10.45. (*а*) Изображение айсберга. (*б*) Гистограмма. (*в*) Результат сегментации на области трех видов с помощью двойного глобального порога по методу Оцу. (Исходное изображение предоставлено службой NOAA)

стограммы на рис. 10.45(*б*) очевидно, что для решения этой задачи необходимы два порога. Изложенная выше процедура приводит к значениям порогов $k_1^* = 80$ и $k_2^* = 177$, которые расположены вблизи середин двух впадин на гистограмме рис. 10.45(*б*). Рис. 10.45(*в*) демонстрирует результат сегментации изображения согласно (10.3-31) с такими двумя порогами. Значение показателя разделимости равно 0,954. Основная причина, почему в этом примере достигается такая хорошая сегментация, является следствием гистограммы, в которой имеются три отдельные моды, разделенные относительно широкими и глубокими впадинами. ■

10.3.7. Обработка с переменным порогом

Как говорилось в разделе 10.3.1, на качество работы алгоритма порогового разделения сильно влияют такие факторы, как шум и неравномерность освещения. В разделах 10.3.4 и 10.3.5 мы показали, что существенную помощь могут оказать сглаживание изображения и информация о контурах. Часто, однако, встречаются ситуации, когда подобная предварительная обработка либо практически невозможна, либо просто не дает достаточного эффекта для решения задачи любым из описанных выше методов. В подобных ситуациях следующим шагом усложнения пороговой обработки является использование переменного порога. В данном разделе будут рассмотрены различные методы выбора величины переменного порога.

Разбиение изображения

Один из простейших методов обработки с переменным порогом состоит в разбиении исходного изображения на непересекающиеся прямоугольные области, в каждой из которых для сегментации используется свое значение порога. Такой метод применяется для компенсации неравномерностей освещения и/или отражения. Прямоугольники выбираются достаточно малыми, чтобы в пределах каждого из них освещение было приблизительно равномерным[16]. Проиллюстрируем этот метод на примере.

Пример 10.20. Обработка с переменным порогом с помощью разбиения изображения.

■ На рис. 10.46(*а*) показано то же изображение, что и на рис. 10.37(*в*), а на рис. 10.46(*б*) приведена его гистограмма. При обсуждении рис. 10.37(*в*) мы пришли к выводу о невозможности его эффективной сегментации с помощью единого глобального порога. Действительно, рис. 10.46(*в*) и (*г*) демонстрируют, каковы были бы результаты сегментации этого изображения с помощью итерационной схемы из раздела 10.3.2 и метода Оцу. Оба метода дают сопоставимые результаты, в которых наблюдаются многочисленные ошибки сегментации.

На рис. 10.46(*д*) показано разбиение исходного изображения на шесть прямоугольных областей, а рис. 10.46(*е*) демонстрирует результат глобального порогового разделения методом Оцу каждой области по отдельности. Хотя можно заметить некоторые ошибки, благодаря сделанному разбиению удалось получить

[16] А также для уменьшения возникающих артефактов «ступенчатости» результата на границах прямоугольников. — *Прим. перев.*

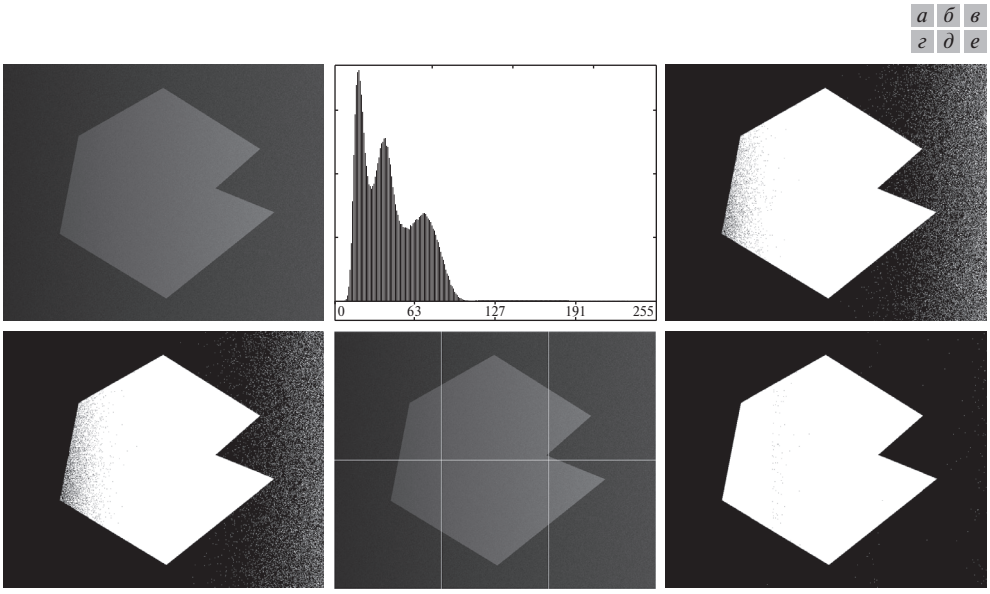


Рис. 10.46. (*а*) Зашумленное изображение с неравномерным освещением и (*б*) его гистограмма. (*в*) Результат сегментации (*а*) с помощью итерационного глобального алгоритма из раздела 10.3.2. (*г*) Результат сегментации методом Оцу. (*д*) Изображение, разбитое на шесть областей меньших размеров. (*е*) Результат применения метода Оцу отдельно для каждой области

приемлемые результаты для изображения, сегментация которого представляет значительные трудности. Причину улучшения легко понять, анализируя гистограммы для областей. Как видно на рис. 10.47, каждая подобласть характеризуется бимодальной гистограммой с глубокой впадиной между модами, что, как мы знаем, позволяет осуществить эффективную обработку с глобальным порогом для каждой из областей.

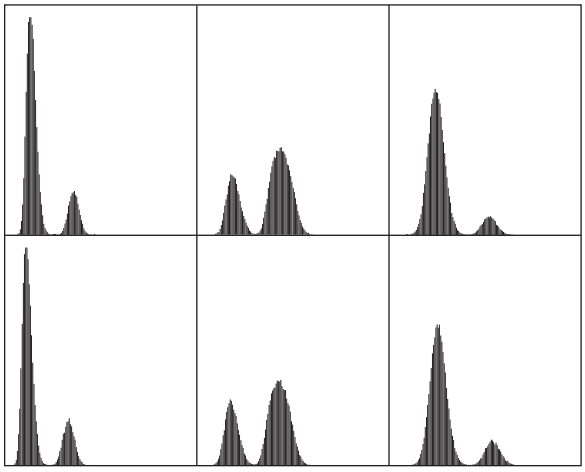


Рис. 10.47. Гистограммы для шести подобластей на рис. 10.46(*д*)

Разбиение на подобласти обычно хорошо помогает в ситуациях, когда интересующий объект и фоновая область имеют сопоставимые размеры, как на рис. 10.46. Если это не так, то данный метод, как правило, приводит к неудачам из-за вероятного появления подобластей, содержащих пиксели только из одной области — или объекта, или фона. Хотя такие случаи можно учесть, применяя дополнительные методы, обнаруживающие наличие в области пикселей обоих видов, подобная логика требует рассматривать различные сценарии и в общем случае приводит к усложнению обработки. В описанных ситуациях более предпочтительными обычно являются методы, описываемые в оставшейся части данного раздела. ■

Выбор переменного порога на основе локальных свойств изображения

Более общий метод, чем описанное в предыдущем разделе разбиение на подобласти, состоит в вычислении порога для каждой точки (x, y) исходного изображения на основании одной или более характеристик, рассчитанных по окрестности этой точки. Хотя это может показаться трудоемкой процедурой, современные алгоритмы и оборудование позволяют выполнять быструю обработку по окрестности, особенно для типовых функций, например логических и арифметических операций.

Для иллюстрации основного подхода, применяемого при локальной пороговой обработке, будем использовать стандартное отклонение и среднее значение пикселей в окрестности каждой точки исходного изображения. Эти две величины весьма полезны для определения локальных порогов, поскольку они характеризуют локальный контраст и локальную среднюю яркость изображения. Обозначим σ_{xy} и m_{xy} стандартное отклонение и среднее значение по множеству пикселей, содержащихся в окрестности S_{xy} с центром в точке изображения (x, y) (см. раздел 3.3.4 о вычислении локального среднего и стандартного отклонения). Следующие соотношения задают типичный вид переменных локальных порогов:

$$T_{xy} = a\sigma_{xy} + bm_{xy}, \tag{10.3-33}$$

где a и b — неотрицательные константы, и

$$T_{xy} = a\sigma_{xy} + bm_G, \tag{10.3-34}$$

где m_G — глобальное среднее значение по изображению. Результат сегментации исходного изображения $f(x, y)$ находится по правилу

$$g(x,y) = \begin{cases} 1 & \text{если } f(x,y) > T_{xy} \\ 0 & \text{если } f(x,y) \le T_{xy}. \end{cases} \tag{10.3-35}$$

Эти вычисления выполняются для всех пикселей изображения, и в каждой точке (x, y) находится свое значение порога с использованием пикселей окрестности S_{xy}.

Обработку с локальным порогом можно сделать значительно более мощной (при умеренном увеличении трудоемкости вычислений) за счет добавления предикатов, которые используют параметры, вычисленные по окрестности точки (x, y):

$$g(x,y) = \begin{cases} 1 & \text{если } Q(\text{локальные параметры}) = \text{TRUE} \\ 0 & \text{если } Q(\text{локальные параметры}) = \text{FALSE}, \end{cases} \tag{10.3-36}$$

где Q — *предикат*, основанный на некоторых параметрах, вычисленных с использованием значений пикселей в окрестности S_{xy}. Например, рассмотрим следующий предикат $Q(\sigma_{xy}, m_{xy})$, основанный на локальных среднем и стандартном отклонении:

$$Q(\sigma_{xy}, m_{xy}) = \begin{cases} \text{TRUE} & \text{если } f(x,y) > a\sigma_{xy} \text{ И } f(x,y) > bm_{xy} \\ \text{FALSE} & \text{в противном случае.} \end{cases} \qquad (10.3\text{-}37)$$

Заметим, что формула (10.3-35) является частным случаем формулы (10.3-36), который возникает, если предикат принимает истинное значение при $f(x,y) > T$ и ложное в противном случае. Здесь предикат основан просто на значении яркости в центре окрестности.

Пример 10.21. Обработка с переменным порогом на основе локальных характеристик изображения.

■ На рис. 10.48(*а*) показано то же изображение дрожжевых клеток, что и в примере 10.18. В этом изображении преобладают три уровня яркости, так что есть

а *б*
в *г*

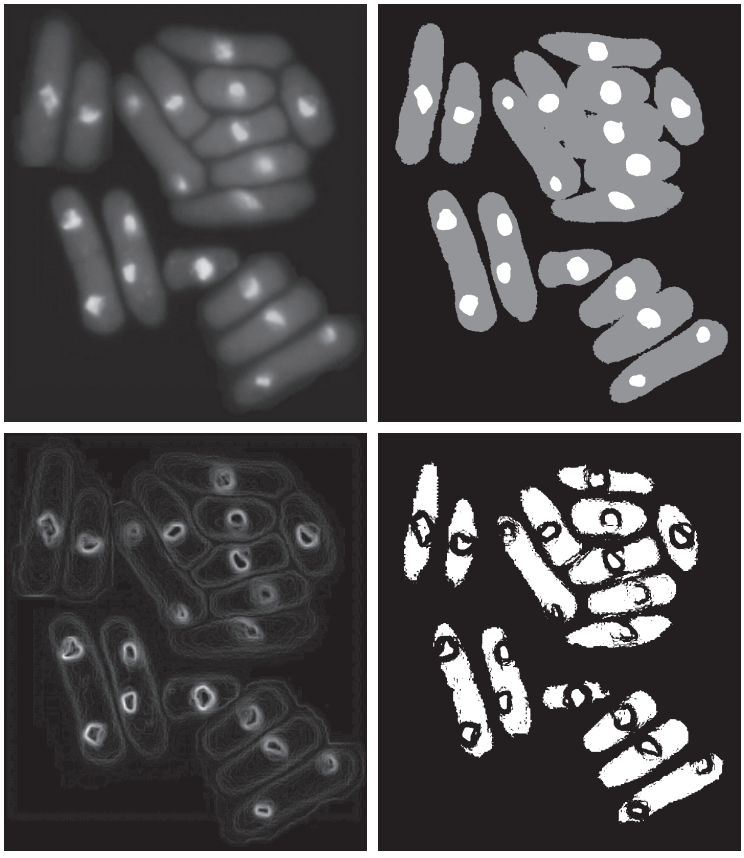


Рис. 10.48. (*а*) Изображение, ранее приведенное на рис. 10.43. (*б*) Результат сегментации с помощью метода с двойным порогом, описанного в разделе 10.3.6. (*в*) Изображение локальных стандартных отклонений. (*г*) Результат локальной пороговой обработки

основания полагать, что сегментация с двумя порогами может дать хорошие результаты. На рис. 10.48(*б*) приведен результат глобальной обработки с двойным порогом по методу, описанному в разделе 10.3.6. Как видно на рисунке, удалось отделить яркие области от темного фона, но области с промежуточным уровнем яркости в правой части изображения сегментируются с ошибками (вспомним, что мы уже сталкивались с похожей проблемой на рис. 10.43(*в*) в примере 10.18). В качестве иллюстрации обработки с локальным порогом вычислим в каждой точке (x, y) исходного изображения стандартное отклонение σ_{xy} по окрестности 3×3. Результат вычисления представлен на рис. 10.48(*в*). Обратите внимание, насколько точно слабые линии очерчивают границы клеток. Теперь построим предикат в форме, аналогичной (10.3-37), но с использованием глобального среднего значения вместо локального среднего m_{xy}. Выбор глобального среднего обычно дает лучшие результаты, когда значение фона близко к постоянному, а яркости всех объектов или больше, или меньше, чем яркость фона. Чтобы окончательно определить предикат, подставим значения $a = 30$ и $b = 1{,}5$ (как обычно в подобных приложениях, эти величины[17] были определены экспериментально). Затем выполняем сегментацию изображения с помощью формулы (10.3-36). Как показывает рис. 10.48(*г*), результаты сегментации хорошо согласуются с областями двух уровней яркости, которые преобладают в исходном изображении. Обратите внимание, в частности, на безошибочную сегментацию всех внешних областей и что большинство внутренних, более ярких областей также выделены правильно. ■

Использование скользящего среднего значения

Особый случай только что рассмотренного метода локальной пороговой обработки основан на вычислении скользящего среднего значения при обходе изображения вдоль строк. Такой алгоритм весьма полезен для обработки документов, когда предъявляются жесткие требования к быстродействию. При этом изображение сканируется строка за строкой со сменой направления на встречное по окончании обхода каждой строки (т. е. зигзагом), чтобы уменьшить разницу освещения. Обозначим z_{k+1} яркость точки, встретившейся в указанном порядке обхода на шаге $k + 1$. Скользящее среднее значение яркости в этой новой точке вычисляется по формуле

$$m(k+1) = \frac{1}{n}\sum_{i=k+2-n}^{k+1} z_i = m(k) + \frac{1}{n}\left(z_{k+1} - z_{k-n+1}\right), \qquad (10.3\text{-}38)$$

Оба выражения справедливы для $k \geq n - 1$, а начальные средние значения формируются только из имеющихся точек.

где n — число точек, используемых для вычисления среднего значения, а $m(1) = z_1/n$. Строго говоря, такое начальное значение не является правильным, потому что среднее значение по единственной точке есть само значение в этой точке, но мы используем указанное $m(1)$, чтобы выполнять однотип-

[17] Такие цифры приведены авторами, хотя значение $a = 30$ представляется чрезмерно большим. — *Прим. перев.*

ные вычисления по формуле (10.3-38) с самого начала. По-другому это можно воспринимать, как если бы первой точке изображения z_1 предшествовал $n - 1$ нулевой элемент. Данный алгоритм[18] инициализируется только один раз, не на каждой строке (в конце текущей строки алгоритм берет для продолжения точки следующей строки в порядке от ее конца к началу, затем точки следующей строки от начала к концу и т. д.). Поскольку скользящее среднее значение вычисляется для каждой точки изображения, сегментация выполняется с помощью формулы (10.3-35) и $T_{xy} = bm_{xy}$, где b — константа, а m_{xy} — величина скользящего среднего, вычисленного по формуле (10.3-38) в точке (x, y) исходного изображения.

Пример 10.22. Пороговая обработка документа с помощью скользящего среднего.

■ На рис. 10.49(*а*) приведено изображение рукописного текста с неравномерным освещением по типу точечного источника. Подобные неравномерности освещения типичны для изображений, полученных с использованием фотовспышки. Рис. 10.49(*б*) демонстрирует результаты сегментации с глобальным порогом по методу Оцу. Как того можно было ожидать, обработка с глобальным порогом оказалась неспособной скомпенсировать такие колебания яркости. На рис. 10.49(*в*) показана успешная сегментация с помощью локального порогового преобразования на основе скользящего среднего значения. В качестве эмпирического правила рекомендуется выбирать n равным пятикратной толщине штрихов. В данном случае средняя толщина штриха 4 пикселя, поэтому мы в формуле (10.3-38) полагали $n = 20$ и использовали константу $b = 0{,}5$.

В качестве еще одной иллюстрации эффективности этого метода сегментации воспользуемся теми же значениями параметров, что и в предыдущем абзаце, для сегментации изображения на рис. 10.50(*а*), в котором присутствует помеха в виде синусоидального колебания яркости. Подобные искажения могут

а б в

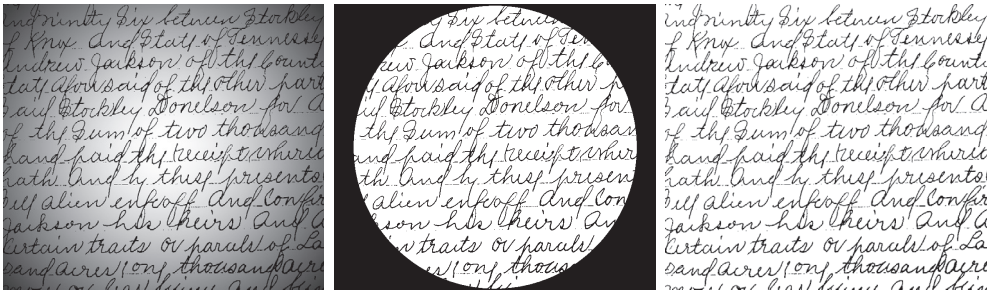


Рис. 10.49. (*а*) Изображение текста, неравномерно освещенного точечным источником. (*б*) Результат глобального порогового разделения методом Оцу. (*в*) Результат локальной пороговой обработки с использованием скользящего среднего значения

[18] Кроме приведенного алгоритма вычисления скользящего среднего для одномерной окрестности, известен также алгоритм вычисления скользящего среднего для двумерной окрестности за конечное число шагов независимо от размеров окрестности. — *Прим. перев.*

а б в

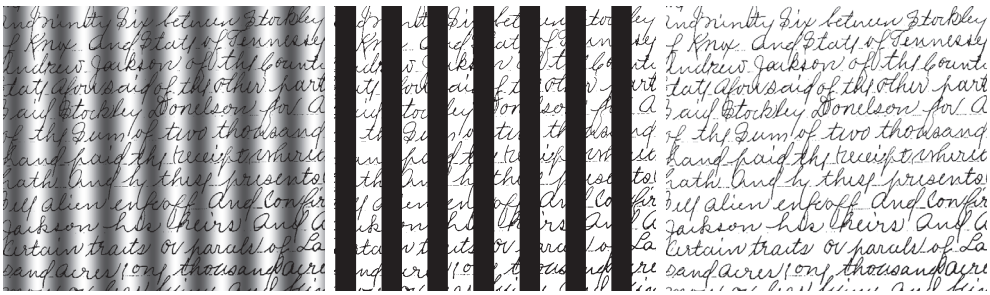


Рис. 10.50. (*а*) Изображение текста, искаженное синусоидальной помехой. (*б*) Результат глобального порогового разделения методом Оцу. (*в*) Результат локальной пороговой обработки с использованием скользящего среднего значения

возникать, например, при плохом качестве заземления блока питания в сканере документов. Как показывают рис. 10.50(*б*) и (*в*), результаты сегментации сопоставимы с приведенными на рис. 10.49.

Интересно отметить, что в обоих случаях успешная сегментация была достигнута с одними и теми же значениями параметров n и b; это говорит об относительной устойчивости данного метода. Вообще говоря, пороговая обработка на основе скользящего среднего значения хорошо работает, когда интересующие объекты маленькие (или имеют небольшую толщину) по сравнению с размерами изображения, что справедливо для изображений печатного или рукописного текста. ■

10.3.8. Пороги, основанные на нескольких переменных

До сих пор мы занимались пороговыми преобразованиями изображений, исходя из одной переменной — черно-белого сигнала яркости. В некоторых случаях для получения изображений могут применяться сенсоры, позволяющие характеризовать каждый элемент изображения более чем одной переменной, что в свою очередь позволяет выполнять *мультиспектральную пороговую обработку*. Хорошим примером такого рода являются цветные изображения, в которых красная (R), зеленая (G) и синяя (B) компоненты в совокупности составляют цветное изображение (см. главу 6). В этом случае каждый пиксель характеризуется тремя значениями и может быть представлен трехмерным вектором $\mathbf{z} = (z_1, z_2, z_3)^T$, компоненты которого суть цвета RGB в данной точке. Такую трехмерную точку часто называют *воксел*[19], указывая тем самым на ее *объемный* характер, в отличие от элементов *изображения*.

Как подробнее обсуждалось в разделе 6.7, пороговое преобразование, основанное на нескольких переменных, можно рассматривать как вычисление некоторого расстояния. Предположим, что стоит задача выделения на цветном изображении всех областей заданного диапазона цветов, скажем, имеющих красный оттенок. Обозначим **a** «средний» цвет интересующих оттенков красного. Один из способов сегментации цветного изображения на основе такого

[19] Название *voxel* происходит от *volumetric pixel* (*англ.*) — «элемент объемного изображения». — *Прим. перев.*

параметра состоит в вычислении расстояния $D(\mathbf{z}, \mathbf{a})$ между точками произвольного цвета $\mathbf{z}$ и желаемого среднего цвета $\mathbf{a}$. Тогда сегментация исходного изображения осуществляется по следующему правилу:

$$g = \begin{cases} 1 & \text{если } D(\mathbf{z},\mathbf{a}) < T \\ 0 & \text{в противном случае,} \end{cases} \tag{10.3-39}$$

где T — некоторый порог, и вычисление расстояния проводится для всех координатных точек исходного изображения, формируя тем самым соответственные элементы сегментированного изображения g. Отметим, что в данном случае знак неравенства противоположный по сравнению с формулой (10.3-1), использовавшейся при пороговом преобразовании одиночной переменной. Причина состоит в том, что уравнение $D(\mathbf{z}, \mathbf{a}) = T$ определяет объемную поверхность (см. рис. 6.43), и интуитивно понятнее представлять себе значения пикселей отсегментированного объекта заключенными внутри объема, а значения пикселей фона — лежащими вне этого объема или на его поверхности. Соотношение (10.3-39) сводится к формуле (10.3-1) после подстановки $D(\mathbf{z}, \mathbf{a}) = -f(x, y)$.

Заметим, что условие $f(x, y) > T$ по существу говорит, что евклидово расстояние между значением f и нулем действительной оси больше значения T. Таким образом, пороговое преобразование основано на вычислении меры расстояния, а форма соотношения (10.3-39) зависит от используемой меры. Если в общем случае $\mathbf{z}$ представляет собой n-мерный вектор, из раздела 2.6.6 мы знаем, что n-мерное евклидово расстояние определяется как

$$D(\mathbf{z},\mathbf{a}) = \|\mathbf{z}-\mathbf{a}\| = \left[(\mathbf{z}-\mathbf{a})^T(\mathbf{z}-\mathbf{a})\right]^{\frac{1}{2}}. \tag{10.3-40}$$

Уравнение $D(\mathbf{z}, \mathbf{a}) = T$ описывает сферу в n-мерном евклидовом пространстве (называемую *гиперсферой*) (на рис. 6.43 показан трехмерный пример). Более мощной мерой расстояния является так называемое *расстояние Махаланобиса*, определяемое как

$$D(\mathbf{z},\mathbf{a}) = \left[(\mathbf{z}-\mathbf{a})^T\mathbf{C}^{-1}(\mathbf{z}-\mathbf{a})\right]^{\frac{1}{2}}, \tag{10.3-41}$$

где $\mathbf{C}$ — ковариационная матрица векторов $\mathbf{z}$, как говорится в разделе 12.2.2. Для такого расстояния уравнение $D(\mathbf{z}, \mathbf{a}) = T$ описывает n-мерный гиперэллипс (на рис. 6.43 показан трехмерный пример). Формула (10.3-41) сводится к (10.3-40), если $\mathbf{C} = \mathbf{I}$ (единичная матрица).

В разделе 6.7 приводился подробный пример использования этих соотношений. Мы также обсудим в разделе 12.2 задачу сегментации областей на изображении с использованием методов распознавания образов, основанных на решающих функциях, которая также может рассматриваться как задача пороговой обработки со многими классами на основе нескольких переменных.

10.4. Сегментация на отдельные области

Как говорилось в разделе 10.1, конечной целью сегментации является разбиение изображения на области. В разделе 10.2 описывался подход к решению этой

проблемы, при котором границы между областями искались на основе разрывов яркости. В разделе 10.3 сегментация осуществлялась с помощью пороговых преобразований, в которых значения порогов определялись по распределению характеристик пикселей, таких как значение яркости или цвет. В этом разделе рассматриваются методы сегментации, основанные на непосредственном поиске самих областей.

> Перед тем, как двигаться дальше, рекомендуется освежить в памяти терминологию, введенную в разделе 10.1.

10.4.1. Выращивание областей

Как ясно из названия, *выращивание областей* представляет собой процедуру, которая группирует пиксели или подобласти в более крупные области по заранее заданным критериям укрупнения. Основной подход состоит в том, что вначале берется множество точек, играющих роль «центров кристаллизации», а затем на них наращиваются области путем присоединения к каждому центру тех пикселей из числа соседей, которые по своим заранее заданным свойствам близки к центру кристаллизации (например имеют яркость или цвет в определенном диапазоне).

Выбор множества, состоящего из одной или более начальных точек, часто может основываться на существе задачи, как будет показано в примере 10.23. При отсутствии априорной информации процедура состоит в вычислении одного и того же набора свойств для каждого пикселя. В конечном счете эти свойства будут использоваться для отнесения пикселя к той или иной области в процессе выращивания. Если в результате вычислений обнаруживаются кластеры значений, то пиксели, близкие по своим свойствам к центроидам таких кластеров, могут выбираться в качестве центров кристаллизации.

Выбор критериев сходства зависит не только от конкретной рассматриваемой задачи, но и от вида данных, из которых состоит имеющееся изображение. Например, задача анализа данных спутниковой съемки земной поверхности существенно основана на использовании цвета, и она стала бы значительно труднее, вплоть до полной невозможности ее решить, при отсутствии такой цветовой информации. Если изображения монохромные, анализ областей проводится с использованием дескрипторов, основанных на значениях яркости и пространственных характеристиках (таких как текстура или статистические моменты). Дескрипторы, применяемые для описания областей изображения, обсуждаются в главе 11.

Использование при выращивании областей одних лишь дескрипторов может привести к ошибочным результатам, если это делается в отрыве от информации о связности областей. Например, представим себе случайную схему расположения пикселей, принимающих только три отличающихся значения яркости. Если строить «области», группируя вместе пиксели с одинаковой яркостью и не обращая внимания на их связность, то это приведет к сегментации, лишенной смысла в контексте нашего обсуждения.

Другая проблема при выращивании областей состоит в формулировании правила остановки этого процесса. Выращивание некоторой области следует

прекратить, когда в изображении больше нет пикселей, удовлетворяющих критериям присоединения к данной области. Такие критерии, как яркость, текстура и цвет, являются по своей природе локальными и не учитывают «историю» выращивания области. Мощность алгоритма выращивания областей можно повысить за счет привлечения дополнительных критериев, использующих, например, такие понятия, как размеры и форма выращиваемой области, а также сходство между пикселем-кандидатом и пикселями, объединенными к данному моменту (скажем, путем сравнения значений яркости нового кандидата и средней яркости уже выращенной области). Использование дескрипторов такого типа основано на предположении, что имеется хотя бы грубая модель ожидаемых результатов.

Пусть $f(x, y)$ — исходное изображение. Обозначим $S(x, y)$ массив, в котором единицами отмечено положение центров кристаллизации, а остальные элементы нулевые. Также обозначим Q предикат, который будет применяться в каждой точке (x, y). Предполагается, что массивы f и S имеют одинаковые размеры. Можно предложить следующий простой алгоритм выращивания областей на основе 8-связности.

1. Найти все компоненты связности $S(x, y)$ и применять к каждой из них операцию эрозии до тех пор, пока не останется один пиксель. Пометить все такие пиксели значением 1; остальные пиксели S помечаются нулевым значением.

См. разделы 2.5.2 и 9.5.3 о связных компонентах и раздел 9.2.1 об эрозии.

2. Сформировать изображение f_Q по следующему правилу: если в точке (x, y) исходное изображение удовлетворяет заданному предикату Q, принимаем $f_Q(x, y) = 1$, иначе принимаем $f_Q(x, y) = 0$.
3. Сформировать изображение g путем присоединения к каждому центру кристаллизации в S всех единичных элементов f_Q, которые являются 8-связными с этим центром кристаллизации.
4. Пометить пиксели каждой связной компоненты g уникальной пометкой данной области (например числами 1, 2, 3...). Это и будет сегментированное изображение, полученное выращиванием областей.

Проиллюстрируем действие этого алгоритма на примере.

Пример 10.23. Сегментация методом выращивания областей.

■ На рис. 10.51(*a*) приводится 8-битовое рентгеновское изображение сварного шва (горизонтальная темная область), в котором имеются несколько трещин и раковин (яркие полосы, идущие горизонтально посередине изображения). Продемонстрируем использование метода выращивания областей для сегментации участков изображения с дефектами сварки. Эти выделенные области могут применяться для контроля качества сварки, включаться в базу исследовательских данных или использоваться для управления автоматическим сварочным оборудованием.

Первоочередной задачей является определение начальных точек для выращивания. Из физики задачи известно, что трещины и раковины пропускают рентгеновские лучи намного лучше, чем сплошной сварочный шов, поэто-

му можно ожидать, что такие дефекты будут выглядеть значительно ярче, чем другие части рентгеновского изображения. Можно выделить начальные точки пороговым преобразованием исходного изображения, устанавливая значение порога равным квантилю большого уровня. На рис. 10.51(*б*) приведена гистограмма исходного изображения, а рис. 10.51(*в*) демонстрирует результат разделения с величиной порога, равной квантилю уровня 0,999 от максимального значения яркости на изображении, что в данном случае составило 254 (о квантилях см. раздел 10.3.5). На рис. 10.51(*г*) показан результат морфологической эрозии каждой связной компоненты на рис. 10.51(*в*) до одиночного пикселя.

Теперь необходимо задать предикат. В данном примере мы хотим присоединить к каждому центру кристаллизации все пиксели, удовлетворяющие двум критериям: (а) наличие 8-связности с данным центром кристаллизации и (б) «сходство» пикселя с этим центром. Используя в качестве меры сходства модуль разности значений яркости, можно выписать следующий предикат, который применяется в каждой точке (x, y):

$$Q = \begin{cases} \text{TRUE} & \text{если } |f(x,y) - f_C| < T, \\ \text{FALSE} & \text{в противном случае,} \end{cases}$$

где f_C — значение яркости в центре кристаллизации, а T — заданный порог. Этот предикат основан на разности яркостей и использует одиночный порог. Мы могли бы указать более сложные схемы, в которых используется переменный порог в каждом пикселе, а также другие характеристики вместо разностей, но в данном случае выписанный предикат является достаточным для решения задачи, как показано ниже в данном примере.

Из предыдущего абзаца известно, что наименьшее значение яркости в центре кристаллизации 255, поскольку изображение было бинаризовано с порогом 254. На рис. 10.51(*д*) показано абсолютное значение разности между константой 255 и исходным изображением. Изображение на рис. 10.51(*д*) содержит все разности, необходимые для вычисления предиката в каждой координатной точке (x, y). Рис. 10.51(*е*) демонстрирует соответствующую гистограмму. Чтобы использовать предикат для установления сходства, необходим еще порог. Поскольку на гистограмме есть три главных моды, можно начать с применения к разностному изображению порогового преобразования с двойным порогом, которое описывалось в разделе 10.3.6. В данном случае были найдены два порога с величинами $T_1 = 68$ и $T_2 = 126$, которые, как видим, соответствуют впадинам на гистограмме. (В качестве краткого отступления и для сравнения мы провели сегментацию изображения с помощью двух найденных порогов. Как показывает результат на рис. 10.51(*ж*), задачу выделения дефектов не удается так решить, несмотря на то, что пороги располагаются в отчетливых впадинах гистограммы.)

На рис. 10.51(*з*) приведен результат бинаризации разностного изображения с помощью одного порога T_1. Черные точки — это те пиксели, в которых предикат принимает значение TRUE; в остальных точках значение предиката FALSE. Важным результатом здесь является то, что для точек в области сварного шва без дефектов предикат несправедлив, и поэтому они не войдут в конечный результат. Точки периферийной области будут рассматриваться в качестве кандидатов алгоритмом выращивания областей, однако они будут отброше-

ны на шаге 3, поскольку не являются 8-связанными с центрами кристаллизации. Фактически, как показывает рис. 10.51(*и*), этот шаг привел к правильной сегментации; это говорит о том, что учет связности в данном случае является фундаментальным требованием. В заключение отметим, что на шаге 4 мы использовали одно и то же значение для заполнения всех областей, выделенных рассматриваемым алгоритмом. В данном примере это предпочтительнее по соображениям наглядности. ■

а *б* *в*
г *д* *е*
ж *з* *и*

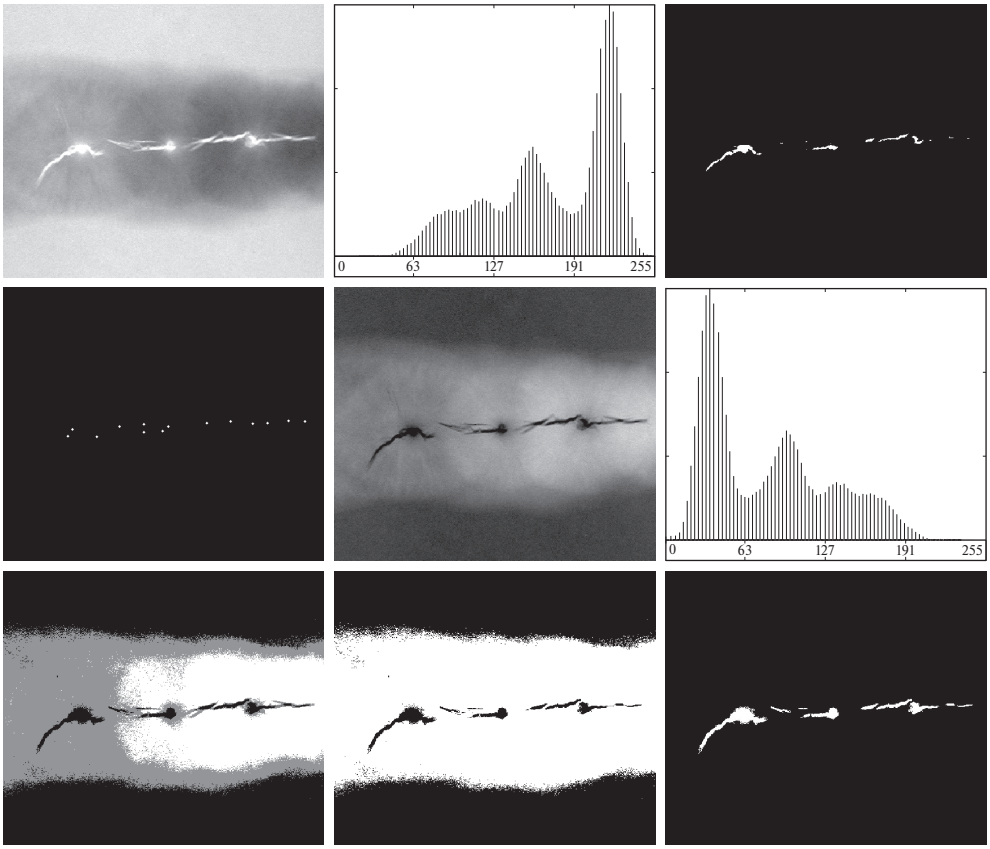


Рис. 10.51. (*а*) Рентгеновское изображение сварного шва с дефектами. (*б*) Гистограмма исходного изображения. (*в*) Изображение начальных областей кристаллизации. (*г*) Найденные центры кристаллизации (точки для наглядности увеличены). (*д*) Абсолютная величина разности константы 255 (значение в центре кристаллизации) и изображения (*а*). (*е*) Гистограмма изображения (*д*). (*ж*) Результат применения преобразования с двойным порогом к разностному изображению. (*з*) Результат порогового разделения разностного изображения по меньшему из двух порогов. (*и*) Результат сегментации, полученный методом выращивания областей. (Исходное изображение предоставлено компанией X-TEK Systems, Ltd.)

10.4.2. Разделение и слияние областей

Описанная в предыдущем разделе процедура выращивает области из множества начальных точек, играющих роль центров кристаллизации. Альтернативный подход состоит в том, чтобы провести первичное разбиение изображения на множество произвольных непересекающихся областей и в дальнейшем осуществлять слияние и/или разделение этих областей, стремясь выполнять условия, сформулированные в разделе 10.1. Ниже излагаются основы метода разделения—слияния.

Пусть вся область изображения обозначена R и выбран предикат Q. Один из подходов к сегментации R состоит в том, чтобы последовательно разбивать ее на все более и более мелкие области-квадранты (т. е. четверти) R_i, пока для каждой из них не будет выполняться условие $Q(R_i)$ = TRUE либо пока разбиение не будет остановлено в связи с достижением минимально допустимого размера области. Работа начинается со всей области изображения. Если $Q(R)$ = FALSE, изображение делится на четверти вертикальной и горизонтальной прямыми, проходящими через середину. Если для какой-то четверти предикат Q принимает значение FALSE, она аналогичным способом делится на более мелкие четверти и так далее. Такой метод разбиения удобно представлять в форме так называемого *квадродерева* (т. е. дерева, у которого вершины, не являющиеся листьями, имеют в точности четыре потомка), как показано на рис. 10.52 (изображения, соответствующие вершинам квадродерева, иногда называют *квадроизображениями* или *квадрообластями*). Отметим, что корень дерева соответствует целому изображению, а каждая другая вершина — какой-то из подобластей. В данном случае только область R_4 подверглась дальнейшему разбиению.

Если использовать только операцию разделения, то в окончательном разбиении изображения обычно присутствуют смежные области с одинаковыми свойствами. Этот недостаток можно устранить, применяя наряду с разделением также операцию слияния. Для соблюдения ограничений из раздела 10.1, касающихся сегментации, требуется, чтобы слиянию подвергались только такие смежные области, пиксели которых в совокупности удовлетворяют предикату Q. Иначе говоря, две смежные области R_j и R_k сливаются только в том случае, если $Q(R_j \cup R_k)$ = TRUE.

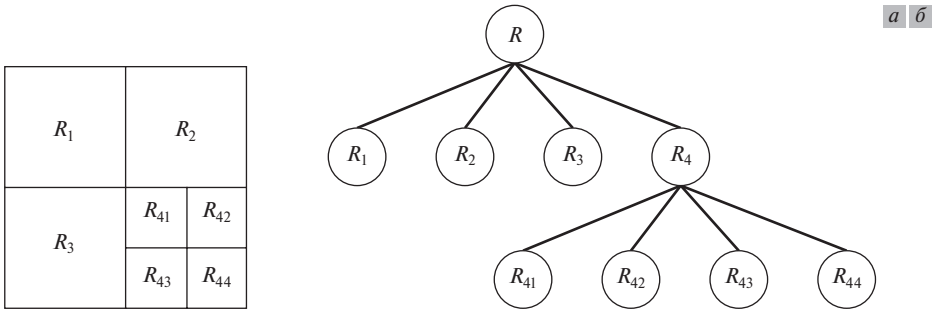


Рис. 10.52. (*а*) Изображение, разбитое на области. (*б*) Соответствующее квадродерево. Корень дерева R представляет всю область изображения

Смежности областей рассматривались в разделе 2.5.2.

Проведенное обсуждение можно кратко суммировать в виде процедуры, на каждом шаге которой выполняются следующие действия:

1. Любая область R_i, для которой $Q(R_i)$ = FALSE, разделяется на четыре непересекающиеся четверти.
2. Когда продолжать разбиение невозможно, любые две смежные области R_j и R_k, для которых $Q(R_j \cup R_k)$ = TRUE, объединяются в одну.
3. Если невозможно выполнить ни одной операции слияния, то наступает окончание процедуры.

Обычно устанавливают минимальные размеры квадрообласти, по достижении которых дальнейшее разделение не производится.

Возможны различные варианты изложенной основной схемы. Например, можно получить значительное упрощение, если на шаге 2 разрешить слияние любых двух смежных областей R_j и R_k в случае, когда каждая из них по отдельности удовлетворяет предикату Q. Это делает алгоритм намного проще (и быстрее), поскольку проверка предиката выполняется только для каждой квадрообласти в отдельности. Как показывает следующий пример, даже при таком упрощении алгоритм позволяет получить сегментацию хорошего качества.

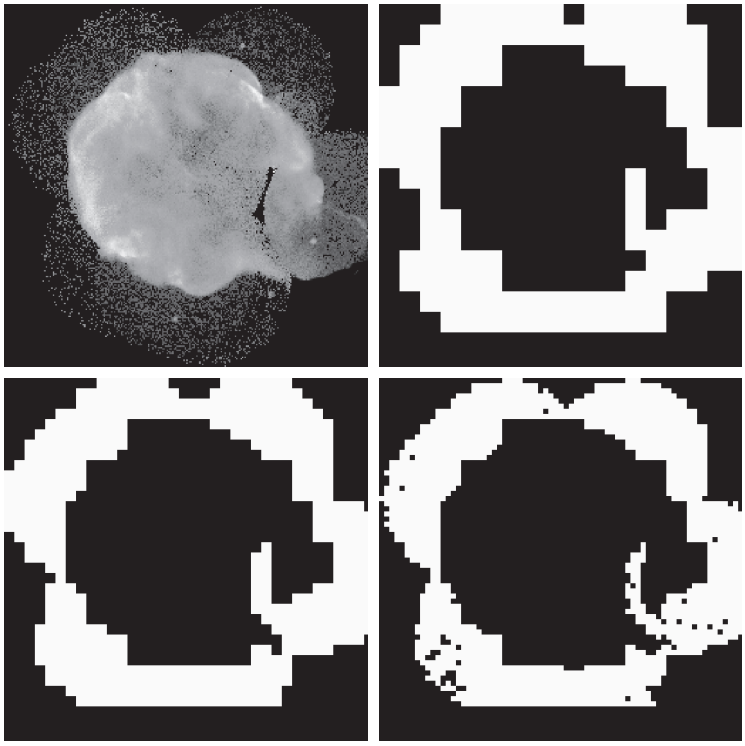
Пример 10.24. Сегментация методом разделения и слияния областей.

■ На рис. 10.53(*а*) приведено изображение Петли Лебедя в рентгеновском диапазоне, имеющее размеры 566×566 пикселей. В этом примере задача состоит в выделении «кольца» разреженной материи, окружающего более плотную центральную область. Интересующая область обладает рядом очевидных свойств, которые должны помочь ее сегментации. Прежде всего, как можно заметить, данные в этой области носят случайный характер; это указывает на то, что стандартное отклонение по такой области должно быть выше, чем для области фона, значение которого близко к 0, или для центральной области, являющейся относительно гладкой. Аналогично среднее значение (средняя яркость) интересующей области должно быть больше, чем области фона (который темнее), и меньше, чем центральной (более светлой) области. Таким образом, можно рассчитывать, что удастся сегментировать интересующую область с помощью следующего предиката:

$$Q = \begin{cases} \text{TRUE} & \text{если } \sigma > a \text{ И } 0 < m < b \\ \text{FALSE} & \text{в противном случае,} \end{cases}$$

где m и σ — соответственно среднее значение и стандартное отклонение для пикселей в квадрообласти, а a и b — константы.

Анализ нескольких участков интересующей области показал, что средняя яркость на них не превышает 125, а стандартное отклонение всегда больше 10. На рис. 10.53(*б*)—(*г*) представлены результаты, получаемые с такими значениями a и b при минимально допустимом размере квадрообласти от 32 до 8. Пиксели квадрообластей, удовлетворяющих предикату, помечены белым цветом; все прочие пиксели помечены черным. Наилучшее соответствие форме интересующей области достигается при минимальных размерах квадрообластей 16×16. Мелкие черные квадраты на рис. 10.53(*г*) — это квадрообласти размера-



а б
в г

Рис. 10.53. (*а*) Изображение сверхновой в Петле Лебедя, полученное в рентгеновском диапазоне с помощью телескопа «Хаббл» агентства NASA. (*б*)—(*г*) Результаты, полученные при минимальных размерах квадрообласти соответственно 32×32, 16×16 и 8×8 пикселей. (Исходное изображение предоставлено агентством NASA)

ми 8×8, для которых предикат не выполняется. Использование квадрообластей меньшего размера приведет к еще большему числу таких черных квадратиков. Если использовать более крупные области, чем показано на рисунке, результаты сегментации будут иметь вид более грубых блоков. Отметим, что во всех случаях сегментированные области (белого цвета) полностью разделяют внутреннюю сглаженную область и фон изображения. Таким образом, в результате сегментации исходное изображение фактически разделено на три области в соответствии с их основными характеристиками: фон, плотная и разреженная области. Если использовать любую из областей белого цвета на рис. 10.53 в качестве маски, можно относительно просто выделить эти области на исходном изображении (задача 10.40). Как и в примере 10.23, такие результаты невозможно получить методами сегментации на основе перепадов яркости или пороговых преобразований. ■

Использованные в предыдущем примере свойства, основанные на среднем значении и стандартном отклонении яркости пикселей внутри некоторой области, представляют собой попытку количественно охарактеризовать *текстуру* области (обсуждение текстуры проводится в разделе 11.3.3). Идея *сегментации*

по текстуре основана на использовании количественных текстурных признаков для построения предикатов $Q(R_j)$. Иначе говоря, сегментация по текстуре может осуществляться с помощью любого из рассмотренных в этом разделе методов, если предикаты основаны на характеристиках текстуры.

10.5. Сегментация по морфологическим водоразделам

До сих пор обсуждались способы сегментации, основанные на трех главных подходах: (а) обнаружении перепадов, (б) пороговой обработке и (в) выращивании или разделении/слиянии областей. Каждый из этих подходов обнаружил свои достоинства (например скорость в случае глобального порогового преобразования) и недостатки (в частности необходимость последующей обработки, например связывания контуров для методов, основанных на обнаружении перепадов яркости). В текущем разделе будет рассмотрен подход, основанный на идее так называемых *морфологических водоразделов*. Как станет ясно из дальнейшего обсуждения, сегментация по водоразделам заключает в себе многие концепции из трех рассмотренных подходов и часто приводит к получению более стабильных результатов сегментации, в том числе к связным границам выделяемых областей. Этот подход также предоставляет простую схему, позволяющую включать в процесс сегментации добавочные ограничения, берущиеся из базы знаний (см. рис. 1.23).

10.5.1. Исходные предпосылки

Понятие водораздела основано на представлении изображения как трехмерной поверхности, заданной двумя пространственными координатами и уровнем яркости в качестве высоты поверхности (рельефа) (см. рис. 2.18(*а*)). В такой «топографической» интерпретации рассматриваются точки[20] трех видов: (а) точки локального минимума; (б) точки, находящиеся на склоне, т. е. с которых вода скатывается в один и тот же локальный минимум; и (в) точки, находящиеся на гребне или пике, т. е. с которых вода с равной вероятностью скатывается более чем в один такой минимум. Применительно к конкретному локальному минимуму набор точек, удовлетворяющих условию (б), называется *бассейном* (или *водосбором*) этого минимума. Множества точек, удовлетворяющих условию (в), образуют линии гребней на поверхности рельефа и называются *линиями водораздела*.

Главная цель алгоритмов сегментации, основанных на введенных понятиях, состоит в нахождении линий водораздела. Основная идея метода выглядит просто, как показывает следующая аналогия. Предположим, что в каждом ло-

[20] Вообще говоря, следует различать *точки трехмерной поверхности* и *точки координатной плоскости xy*. Но поскольку между ними имеется взаимно однозначное соответствие, а также в силу того, что из контекста всегда ясно, о чем идет речь, в обоих случаях будет использоваться просто термин *точка*. Аналогично, построенная трехмерная поверхность также будет называться *изображением*. — *Прим. перев.*

кальном минимуме проколото отверстие, после чего весь рельеф заполняется водой, равномерно поступающей снизу через эти отверстия, так что уровень воды всюду одинаков. Когда поднимающаяся вода в двух соседних бассейнах близка к тому, чтобы слиться вместе, в этом месте ставится перегородка, препятствующая слиянию. В конце концов заполнение достигает фазы, когда над водой остаются видны только верхушки перегородок. Эти перегородки, соответствующие линиям водоразделов, и образуют связные границы, выделенные с помощью алгоритма сегментации по водоразделам.

Дальнейшее объяснение изложенной идеи дается с помощью рис. 10.54. На рис. 10.54(*а*) показано полутоновое изображение, а на рис. 10.54(*б*) — его представление в виде рельефа, где высота «гор» пропорциональна значениям яркости в точках исходного изображения. Для наглядности на склонах нанесены тени, которые не следует путать со значениями яркости; интерес представляет лишь объемное представление общего рельефа. Во избежание выливания воды за края изображения вообразим, что все изображение по периметру обнесено перегородкой, по высоте превышающей самую высокую гору, т. е. максимально возможный уровень яркости изображения.

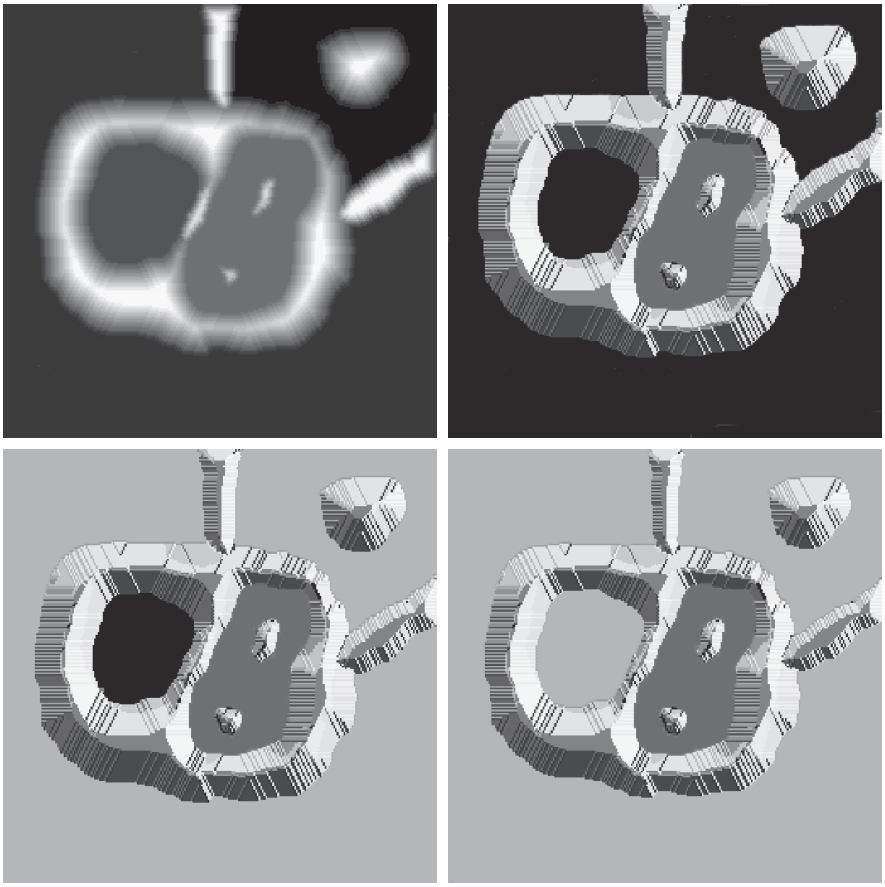


Рис. 10.54. (*а*) Исходное изображение. (*б*) Рельефное представление. (*в*)—(*г*) Две стадии заполнения

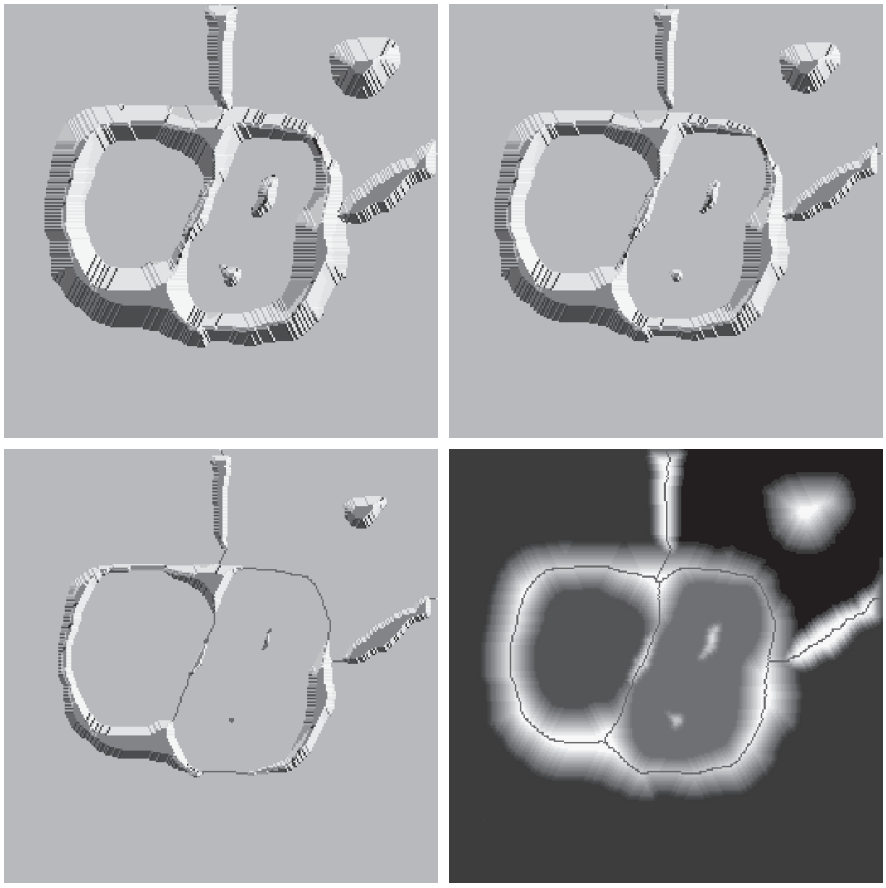


Рис. 10.54. (*Продолжение.*) (*д*) Результат дальнейшего заполнения. (*е*) Начало слияния двух бассейнов (между ними строится короткая перегородка). (*ж*) Перегородки большей длины. (*з*) Окончательные линии водоразделов (результат сегментации). (Изображения предоставлены д-ром С. Бёше, CMM/Ecole des Mines de Paris)

Как уже было сказано, полагаем, что в каждом локальном минимуме (которые показаны темными областями на рис. 10.54(*б*)) проколоты отверстия, через которые рельеф постепенно заполняется водой. На рис. 10.54(*в*) изображен первый этап такого заполнения, когда «вода», показанная серым цветом, закрыла только области, соответствующие наиболее темному фону изображения. На рис. 10.54(*г*) и (*д*) видно, что теперь вода поднялась и начала заполнять соответственно левый и правый внутренние бассейны. По мере дальнейшего подъема воды в какой-то момент эти два бассейна должны будут слиться; первые признаки этого показаны на рис. 10.54(*е*). Здесь во избежание слияния правого и левого внутренних бассейнов при повышении уровня воды строится короткая перегородка, состоящая из одиночных пикселей (построение перегородок подробно рассматривается в следующем разделе). Это явление становится более выраженным по мере того, как вода продолжает подниматься, что демонстрирует рис. 10.54(*ж*). На этом рисунке видна более длинная перегородка

между бассейнами, а также еще одна перегородка в правой верхней части правого бассейна. Последняя была построена, чтобы предотвратить слияние этого бассейна с областью, соответствующей фону. Этот процесс продолжается до тех пор, пока уровень заполнения водой не достигнет того, который соответствует максимальной яркости в исходном изображении. Заключительный набор перегородок соответствует линиям водораздела, которые и представляют собой искомый результат сегментации. Для рассматриваемого примера этот результат показан на рис. 10.54(*з*) темной линией шириной в один пиксель, наложенной на исходное изображение. Отметим то важное свойство, что линии водоразделов образуют связный путь, тем самым определяя непрерывные границы между областями.

Одним из важнейших применений сегментации по водоразделам является выделение на фоне изображения однородных по яркости объектов (в виде пятен). Зачастую протяженные области характеризуются заметными изменениями яркости, но имеют малые значения градиента. Поэтому на практике часто встречается ситуация, когда метод сегментации по водоразделам применяется не к самому изображению, а к градиенту этого изображения. В такой постановке локальные минимумы бассейнов хорошо согласуются с малыми значениями градиента, что обычно соответствует интересующим объектам.

10.5.2. Построение перегородок

Перед тем, как двигаться дальше, рассмотрим способ построения перегородок вдоль линий водоразделов, требующийся для описанного алгоритма сегментации. Построение перегородок основано на двухградационных изображениях, которые являются подмножествами двумерного целочисленного пространства Z^2 (см. раздел 2.4.2). Простейший способ построения линий раздела для множеств, образованных двоичными точками, состоит в использовании морфологической дилатации (см. раздел 9.2.2).

Начальные сведения о построении перегородок с помощью дилатации иллюстрирует рис. 10.55. На рис. 10.55(*а*) показаны участки двух бассейнов на $(n-1)$-м шаге заполнения, а на рис. 10.55(*б*) — те же участки на следующем, n-м шаге. Происходит слияние этих двух бассейнов, и, следовательно, должна быть построена перегородка, препятствующая данному событию. Для согласования с системой обозначений, которая будет принята далее, обозначим через M_1 и M_2 множества точек, соответствующие локальным минимумам двух рассматриваемых бассейнов. Через $C_{n-1}(M_1)$ и $C_{n-1}(M_2)$ обозначим множества точек, покрытых водой в этих *бассейнах* на $(n-1)$-м шаге заполнения. Эти два множества показаны серым цветом на рис. 10.55(*а*).

Обозначим $C[n-1]$ объединение двух последних множеств. На рис. 10.55(*а*) имеется две компоненты связности (о связности компонент см. раздел 2.5.2), а на рис. 10.55(*б*) — только одна, охватывающая две прежние компоненты связности (обозначены пунктирными линиями). Тот факт, что две компоненты связности превратились в *одну*, указывает, что на n-м шаге заполнения произошло слияние двух бассейнов в один. Обозначим через q образовавшуюся единую связную компоненту. Заметим, что две компоненты, имевшиеся на шаге $n-1$, можно выделить из множества q одной операцией И: $q \cap C[n-1]$. Отметим также, что

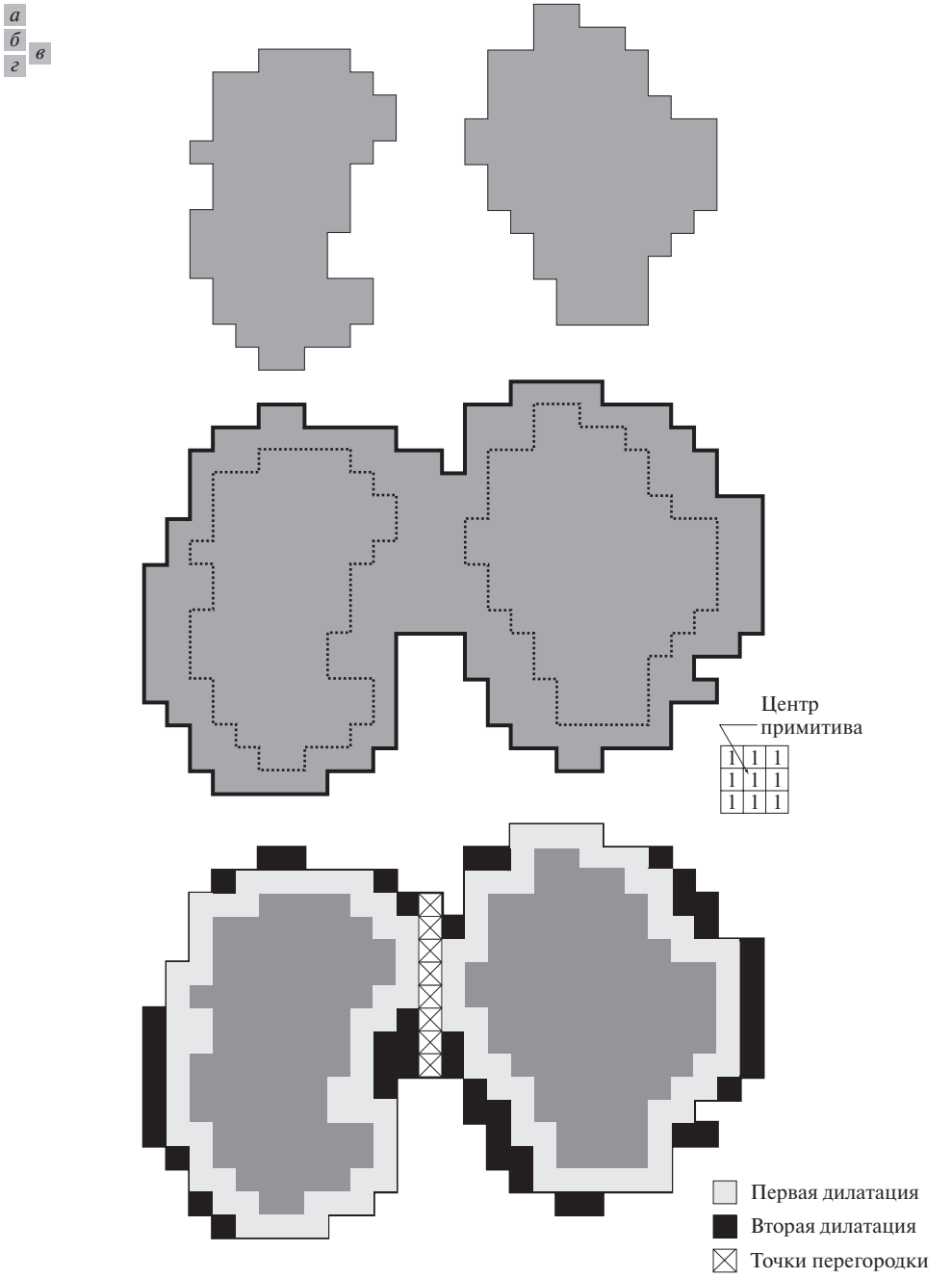


Рис. 10.55. (*а*) Два частично заполненных бассейна на $(n-1)$-м шаге заполнения. (*б*) n-й шаг заполнения, при котором два бассейна сливаются вместе. (*в*) Примитив, используемый в операции дилатации. (*г*) Результаты дилатации и построения перегородки

все точки, соответствующие отдельному бассейну, образуют одну компоненту связности.

Допустим, к каждой компоненте связности на рис. 10.55(*а*) применяется операция дилатации по примитиву, показанному на рис. 10.55(*в*), с соблюдением двух условий: (1) применение дилатации должно ограничиваться множеством q (это значит, что центр примитива может располагаться только в точках q); и (2) дилатация не должна выполняться в тех точках, где это приведет к слиянию обрабатываемых множеств, так что они станут единой связной компонентой. Из рис. 10.55(*г*) видно, что при первом проходе дилатации границы каждой из исходных компонент связности расширяются (показано светло-серым цветом). Заметим, что в ходе дилатации условие (1) соблюдалось для всех точек, а условие (2) не применялось ни разу, так что границы обеих областей раздвигались равномерно.

При втором проходе дилатации (показанном черным цветом) для некоторых точек соблюдалось условие (2), но нарушалось условие (1), что привело к разрывности множества точек, добавляемых по периметру, как это видно из рисунка. Очевидно также, что единственными точками множества q, для которых выполнено условие (1) и не выполнено условие (2), являются точки (перечеркнутые крест-накрест на рис. 10.55(*г*)), образующие связную линию толщиной в один пиксель. Эта линия и составляет искомую разделяющую перегородку на n-м шаге подъема уровня воды. Построение перегородки на этом шаге завершается тем, что всем точкам найденной линии присваивается значение яркости, превышающее максимальное в изображении. Обычно высота всех перегородок принимается равной максимально возможному уровню яркости плюс единица. Это предотвращает возможность слияния бассейнов поверх построенной перегородки в будущем по мере дальнейшего подъема воды. Важно отметить, что перегородки, построенные с применением данной процедуры и являющиеся искомыми границами сегментации, представляют собой связные компоненты. Иначе говоря, данный метод исключает проблемы, связанные с появлением разрывов в линиях сегментации.

Хотя приведенное описание процедуры основывается на простом примере, точно такой же метод применяется и в более сложных случаях; при этом используется тот же симметричный примитив размерами 3×3, показанный на рис. 10.55(*в*).

10.5.3. Алгоритм сегментации по водоразделам

Пусть $M_1, M_2, ..., M_R$ — множества точек координатной плоскости, соответствующие локальным минимумам поверхности $g(x, y)$; как указывалось в конце раздела 10.5.1, $g(x, y)$ обычно является градиентным изображением. Обозначим через $C(M_i)$ множество точек бассейна, отвечающего локальному минимуму M_i (напомним, что точки любого бассейна образуют компоненту связности). Обозначения min и max будем использовать для указания наименьшего и наибольшего значений изображения $g(x, y)$. Наконец, запись $T[n]$ означает множество точек (s, t), для которых $g(s, t) < n$, т. е.

$$T[n] = \{(s,t) \,|\, g(s,t) < n\}. \qquad (10.5\text{-}1)$$

С геометрической точки зрения $T[n]$ есть множество точек, в которых поверхность $g(x, y)$ лежит ниже плоскости $g(x, y) = n$.

При заполнении рельефа водой уровень поднимается в виде *целочисленных* дискретных приращений от $n = \min + 1$ до $n = \max + 1$. В процессе подъема воды на любом шаге n алгоритму необходимо знать число точек, лежащих ниже уровня воды. Вообразим, что все точки множества $T[n]$ (т. е. которые лежат ниже плоскости $g(x, y) = n$) отмечены черным цветом, а все остальные — белым. Тогда при произвольном (n-м) шаге подъема уровня воды рассматриваемая трехмерная поверхность в проекции на плоскость xy может быть представлена двоичным изображением, в котором черные точки соответствуют точкам исходной функции, лежащим ниже плоскости $g(x, y) = n$. Такая интерпретация весьма полезна для понимания последующего изложения.

Пусть $C_n(M_i)$ обозначает множество точек бассейна с локальным минимумом M_i, которые оказались залитыми водой на шаге n. С учетом вышесказанного $C_n(M_i)$ можно рассматривать как двоичное изображение, задаваемое соотношением

$$C_n(M_i) = C(M_i) \cap T[n]. \tag{10.5-2}$$

Другими словами, $C_n(M_i) = 1$ в тех точках (x, y), для которых одновременно выполняется $(x, y) \in C(M_i)$ и $(x, y) \in T[n]$; в остальных точках изображения $C_n(M_i) = 0$. Геометрическая интерпретация выражения в правой части (10.5-2) понятна: с помощью операции пересечения на n-м шаге подъема уровня воды мы выделяем ту часть двоичного изображения $T[n]$, которая относится к локальному минимуму M_i.

Пусть теперь $C[n]$ — объединение залитых водой частей всех бассейнов на шаге n:

$$C[n] = \bigcup_{i=1}^{R} C_n(M_i). \tag{10.5-3}$$

Тогда $C[\max+1]$ есть объединение всех имеющихся бассейнов:

$$C[\max+1] = \bigcup_{i=1}^{R} C(M_i). \tag{10.5-4}$$

Можно показать (задача 10.41), что при работе алгоритма никогда не происходит удаления элементов из множеств $C_n(M_i)$ и $T[n]$; таким образом, при увеличении n число элементов этих множеств либо возрастает, либо остается неизменным. Следовательно, $C[n-1]$ является подмножеством $C[n]$. Согласно равенствам (10.5-2) и (10.5-3), $C[n]$ также является подмножеством $T[n]$, а значит, $C[n-1]$ также есть подмножество $T[n]$. Отсюда следует важный результат: каждая компонента связности множества $C[n-1]$ содержится ровно в одной связной компоненте множества $T[n]$.

Алгоритм нахождения линий водораздела начинается с инициализации $C[\min+1] = T[\min+1]$. После этого алгоритм выполняется рекуррентно, вычисляя на n-м шаге множество $C[n]$ на основании $C[n-1]$ с помощью следующей процедуры. Пусть $Q[n]$ — множество компонент связности множества $T[n]$. Тогда для каждой связной компоненты $q \in Q[n]$ есть три возможности:

1. $q \cap C[n-1]$ — пустое множество;

2. $q \cap C[n-1]$ содержит единственную компоненту связности множества $C[n-1]$;
3. $q \cap C[n-1]$ содержит более одной компоненты связности множества $C[n-1]$.

Способ построения $C[n]$ по $C[n-1]$ зависит от того, какое из этих трех условий имеет место. Условие 1 означает, что встретился новый локальный минимум (начинается наполнение нового бассейна); в этом случае для построения множества $C[n]$ компонента q добавляется к $C[n-1]$. Условие 2 имеет место, когда q лежит внутри бассейна некоторого локального минимума; в этом случае для построения множества $C[n]$ компонента q также добавляется к $C[n-1]$. Условие 3 возникает, когда встретились точки гребня, разделяющего два или более бассейна. В этом случае дальнейший подъем воды привел бы к слиянию этих бассейнов, поэтому внутри связной компоненты q должна быть построена перегородка (или перегородки, если объединяется более двух бассейнов), не позволяющая бассейнам слиться вместе. Как объяснялось в предыдущем разделе, перегородку толщиной в один пиксель при необходимости можно построить, применяя к множеству $q \cap C[n-1]$ операцию дилатации по примитиву 3×3, заполненному единицами, и затем ограничивая результат дилатации точками множества q.

Эффективность описанного алгоритма можно повысить, используя только те значения n, которые соответствуют уровням яркости, встречающимся в изображении $g(x, y)$; эти значения, как и величины min и max, можно определить по гистограмме изображения $g(x, y)$.

Пример 10.25. Иллюстрация работы алгоритма сегментации по водоразделам.

■ Рассмотрим представленные на рис. 10.56(*а*) и (*б*) исходное изображение и его градиент. Применение описанного алгоритма к градиентному изображению дает в результате линии водораздела, показанные на рис. 10.56(*в*) белым цветом поверх изображения градиента. На рис. 10.56(*г*) те же границы сегментов наложены на исходное изображение. Как отмечалось в начале этого раздела, получаемые при таком методе сегментации границы областей обладают важным свойством: они образуют связные линии. ■

10.5.4. Использование маркеров

Непосредственное применение алгоритма сегментации по водоразделам в том виде, как описывалось в предыдущем разделе, обычно приводит к *избыточной сегментации*, вызванной шумом и другими локальными неровностями на градиентном изображении. Как видно из рис. 10.57, избыточная сегментация может быть настолько значительной, что сделает результат практически бесполезным. В данном случае это означает огромное число областей, выделенных при сегментации. Практическое решение этой проблемы состоит в том, чтобы ограничить допустимое число областей путем включения в состав процедуры шага предварительной обработки, служащего для привнесения добавочных знаний в процедуру сегментации.

а б
в г

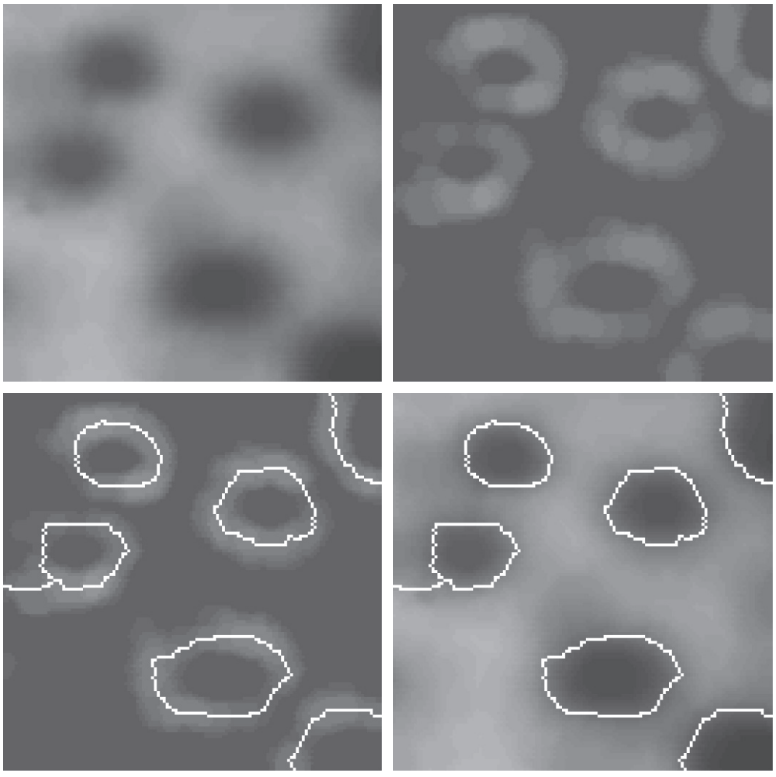


Рис. 10.56. (*а*) Изображение пятен. (*б*) Градиент исходного изображения. (*в*) Линии водоразделов, наложенные на градиентное изображение. (*г*) Линии водоразделов, наложенные на исходное изображение. (Изображения предоставлены д-ром С. Бёше, CMM/Ecole des Mines de Paris)

Подход, применяемый для управления избыточной сегментацией, основан на идее маркеров. *Маркер* представляет собой связную компоненту, принадлежащую изображению. Будем различать *внутренние* маркеры, относящиеся к интересующим объектам, и *внешние* маркеры, соответствующие фону. Процедура выбора маркера обычно состоит из двух главных шагов: (1) предобработка, и (2) выработка критериев, которым должны удовлетворять маркеры. Для иллюстрации снова обратимся к изображению на рис. 10.57(*а*). Одной из причин, которые привели к результату с избыточной сегментацией, показанному на рис. 10.57(*б*), является наличие большого числа локальных минимумов. Принимая во внимание размеры соответствующих бассейнов, многие из этих минимумов отражают несущественные детали. Как многократно указывалось ранее, эффективным способом снижения влияния, которое оказывают мелкие пространственные детали изображения, является фильтрация изображения с помощью сглаживающего фильтра. Такая схема предварительной обработки пригодна и для данного случая.

Пусть внутренний маркер определяется как область, (1) окруженная точками с большей «высотой»; (2) такая, что ее точки образуют компоненту связности; и (3) все точки которой имеют одинаковые значения яркости. Внутренние

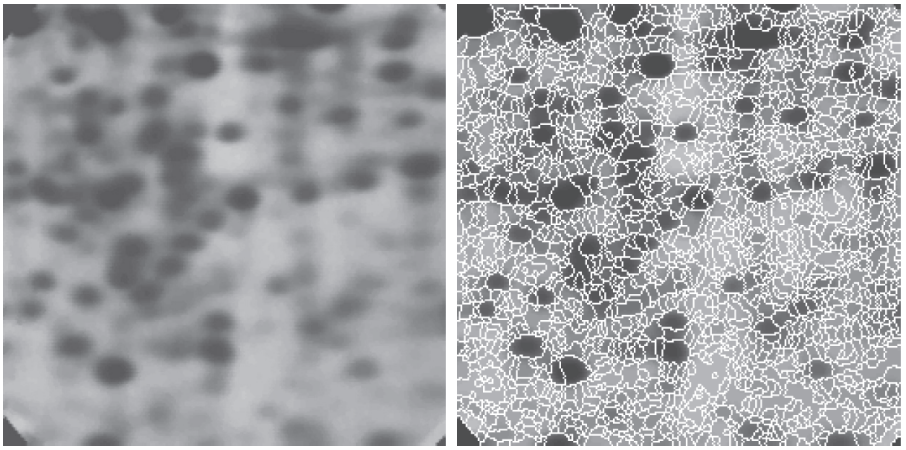


Рис. 10.57. (*а*) Исходное изображение. (*б*) Результат применения алгоритма сегментации по водоразделам к градиентному изображению, на котором хорошо видна избыточная сегментация. (Изображения предоставлены д-ром С. Бёше, CMM/Ecole des Mines de Paris)

маркеры, найденные на сглаженном изображении согласно такому определению, показаны на рис. 10.58(*а*) в виде светло-серых пятен. Затем к сглаженному изображению применяется вышеописанный алгоритм сегментации по водоразделам с тем ограничением, что в качестве локальных минимумов рассматриваются *только* внутренние маркеры. Рис. 10.58(*а*) демонстрирует полученные в результате линии водоразделов, которые по определению являются внешними маркерами. Заметим, что точки вдоль этих водоразделов проходят по самым высоким местам между соседними маркерами.

Внешние маркеры на рис. 10.58(*а*) эффективно разграничивают изображение на области, каждая из которых содержит единственный внутренний маркер и часть фоновой области. После этого задача сводится к разделению каждой такой области на две: одиночный объект и окружающий его фон. Для решения этой упрощенной задачи можно использовать многие методы сегментации, ранее рассмотренные в данной главе. В частности, можно просто применить тот же алгоритм сегментации по водоразделам к каждой отдельной области. Иначе говоря, получив градиент по всему сглаженному изображению (как на рис. 10.56(*б*), мы затем ограничиваем применение алгоритма границами только одного «водосбора», внутри которого находится маркер данной конкретной области. Полученные с применением такого подхода результаты приведены на рис. 10.58(*б*). Улучшение по сравнению с рис. 10.57(*б*) очевидно.

Способ выбора маркеров может меняться от простейших процедур, основанных на связности и значениях яркости, как было только что продемонстрировано, до более сложных описаний, в которых участвуют размеры, форма, местоположение, относительные расстояния, текстурные признаки и т. д. (дескрипторы, применяемые для описания, рассматриваются в главе 11). Суть состоит в том, что применение маркеров позволяет учесть априорные знания, имеющие отношение к задаче сегментации. Напомним читателю, что в сегментации, как и в еще более сложных задачах, повседневно решаемых человеческим зрением,

а *б*

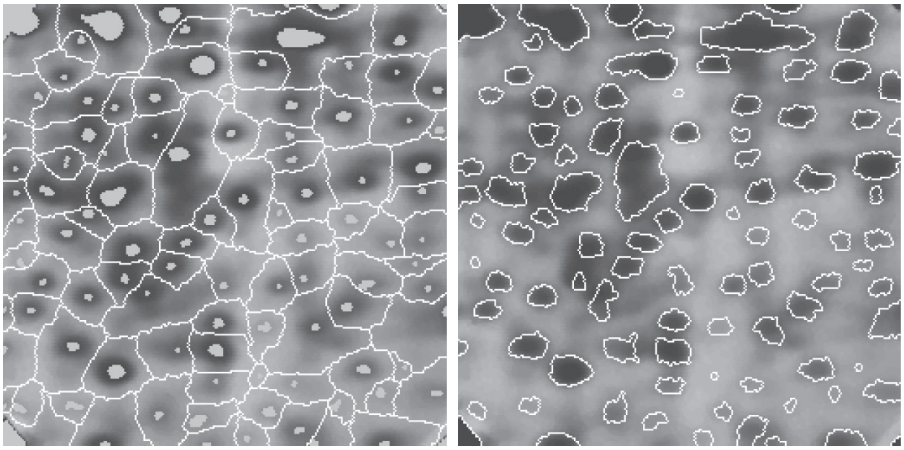


Рис. 10.58. (*а*) Сглаженное изображение с показанными внутренними (светло-серые области) и внешними (линии водоразделов) маркерами. (*б*) Результаты сегментации, на которых видно улучшение по сравнению с рис. 10.57(*б*). (Изображения предоставлены д-ром С. Бёше, CMM/Ecole des Mines de Paris)

часто полезными являются априорные знания, среди которых одно из самых известных — это конкретные знания о задаче. Тот факт, что метод сегментации по водоразделам предлагает каркас, в котором можно эффективно применять знания такого типа, является важным достоинством этого метода.

10.6. Использование движения при сегментации

Движение — это мощный ключ, которым пользуются люди и многие животные для выделения интересующих объектов или областей на фоне несущественных деталей. В прикладных задачах обработки изображений движение возникает при относительном перемещении сенсорной системы и наблюдаемой сцены, например в робототехнике, автономных навигационных системах, при анализе динамических сцен и др. В последующих разделах рассматривается использование движения при сегментации, как в пространственной, так и в частотной области.

10.6.1. Пространственные методы

Основной подход

Один из простейших подходов к обнаружению изменений в последовательности изображений, произошедших между двумя кадрами $f(x, y, t_i)$ и $f(x, y, t_j)$, полученными в моменты времени t_i и t_j соответственно, состоит в поэлементном сравнении этих двух изображений. Один из способов такого сравнения — построение разностного изображения. Предположим, что имеется опорное изображение, содержащее только неподвижные компоненты. Сравнение этого изображения с последующим изображением той же сцены, но содержащим движущийся объ-

ект, приведет к тому, что в разности двух изображений неподвижные составляющие взаимно уничтожатся, а ненулевые значения останутся только в местах, соответствующих подвижным компонентам изображений.

Разностное изображение между двумя изображениями, взятыми в моменты времени t_i и t_j, можно определить следующим образом:

$$d_{ij}(x,y)=\begin{cases}1 & \text{если } \left|f(x,y,t_i)-f(x,y,t_j)\right|>T\\ 0 & \text{в противном случае,}\end{cases} \qquad (10.6\text{-}1)$$

где T — заданный порог. Заметим, что $d_{ij}(x,y)$ принимает значение 1 в точке с пространственными координатами (x,y) только в том случае, если значения яркости в этой точке на двух изображениях отличаются *достаточно существенно*, что определяется заданным порогом T. Предполагается, что все изображения имеют одинаковые размеры, поэтому диапазон координат (x,y) в (10.6-1) совпадает с размерами исходных изображений, так что любое разностное изображение $d_{ij}(x,y)$ имеет те же размеры, что и изображения в последовательности.

При обработке динамических изображений все единичные пиксели изображения $d_{ij}(x,y)$ считаются результатом движения объекта. Такой подход применим только в том случае, если оба изображения являются пространственно совмещенными, а также если флуктуации яркости не выходят за границы, задаваемые порогом T. На практике ненулевые элементы разности $d_{ij}(x,y)$ могут появиться вследствие шума. Обычно такие элементы образуют небольшие изолированные группы точек в разностном изображении; простой способ их удаления состоит в том, что в изображении $d_{ij}(x,y)$ находятся 4- или 8-связанные области из единичных элементов и области, содержащие меньше заранее заданного числа элементов, удаляются. Хотя это и может привести к пропуску малых и/или медленно движущихся объектов, данный подход повышает шансы того, что оставшиеся ненулевые элементы разностного изображения действительно возникли в результате движения.

Накопленные разности

Рассмотрим последовательность кадров изображения $f(x, y, t_1)$, $f(x, y, t_2)$,..., $f(x, y, t_n)$ и пусть $f(x, y, t_1)$ — опорное изображение. *Накопленное разностное изображение* (НРИ) формируется путем сравнения опорного изображения с каждым следующим кадром последовательности. Для каждого пикселя в накопленном разностном изображении ведется счетчик, значение которого увеличивается всякий раз, когда в этой точке наблюдается отличие кадра последовательности от опорного изображения. Таким образом, при сравнении k-го кадра с опорным значение каждого пикселя накопленного изображения указывает число раз, когда яркость в этой точке отличалась (с точностью до порога T в равенстве (10.6-1) от значения соответствующего пикселя в опорном изображении.

Рассмотрим следующие три вида накопленных разностных изображений: *абсолютные*, *положительные* и *отрицательные* НРИ. Эти виды НРИ определяются следующим образом. Обозначим опорное изображение $R(x,y)$ и пусть k для простоты обозначает момент времени t_k, т. е. $f(x, y, k) = f(x, y, t_k)$; полагаем также, что $R(x,y) = f(x, y, 1)$. Тогда для любого $k > 1$ значения элементов (которые суть *счетчики*) НРИ упомянутых видов определяются в каждой точке (x,y) следующим образом:

$$A_k(x,y)=\begin{cases}A_{k-1}(x,y)+1, & \text{если } |R(x,y)-f(x,y,k)|>T\\ A_{k-1}(x,y) & \text{в противном случае,}\end{cases} \tag{10.6-2}$$

$$P_k(x,y)=\begin{cases}P_{k-1}(x,y)+1, & \text{если } (R(x,y)-f(x,y,k))>T\\ P_{k-1}(x,y) & \text{в противном случае}\end{cases} \tag{10.6-3}$$

и

$$N_k(x,y)=\begin{cases}N_{k-1}(x,y)+1, & \text{если } (R(x,y)-f(x,y,k))<-T\\ N_{k-1}(x,y) & \text{в противном случае,}\end{cases} \tag{10.6-4}$$

где $A_k(x,y)$, $P_k(x,y)$ и $N_k(x,y)$ обозначают соответственно абсолютное, положительное и отрицательное НРИ после обработки k изображений последовательности.

Начальные значения всех счетчиков берутся равными нулю. Заметим также, что все НРИ имеют те же размеры, что и изображения последовательности.

Пример 10.26. Формирование абсолютного, положительного и отрицательного накопленных разностных изображений.

■ На рис. 10.59 в виде яркостных изображений показаны НРИ трех указанных видов для прямоугольного светлого объекта размерами 75×50 пикселей, который движется в направлении правого нижнего угла изображения со скоростью $5\sqrt{2}$ пикселей за кадр. Размеры всех изображений составляют 256×256 пикселей. Для данного случая, когда яркость объекта больше яркости фона, заметим следующее: (1) размеры области ненулевых элементов положительного НРИ равны размерам движущегося объекта; (2) положение этой области на положительном НРИ совпадает с положением объекта в опорном кадре; (3) число ненулевых точек в положительном НРИ перестает возрастать, когда объект полностью сдвигается за пределы области, которую он занимал в опорном кадре; (4) абсолютное НРИ состоит из объединения областей положительного и отрицательного НРИ; (5) направление и скорость движения объекта можно определить по ненулевым областям в абсолютном и отрицательном НРИ. Для обратного случая, когда яркость объекта меньше яркости фона, эти утверждения справедливы при замене слов «положительное НРИ» на «отрицательное НРИ» ■

а б в

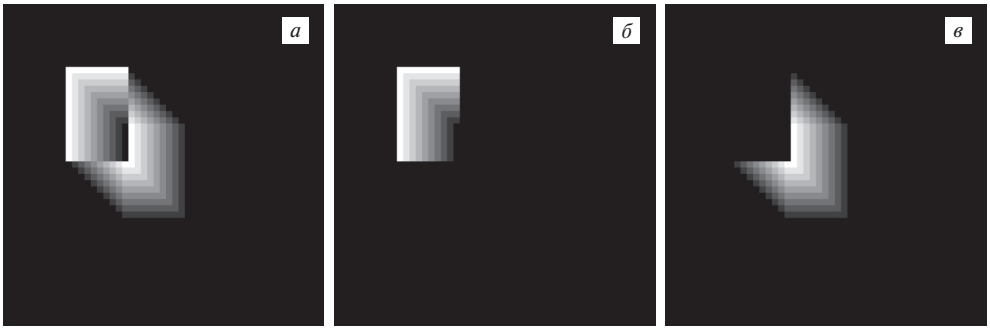


Рис. 10.59. Накопленные разностные изображения прямоугольного объекта, перемещающегося в направлении правого нижнего угла. (*а*) Абсолютное НРИ. (*б*) Положительное НРИ. (*в*) Отрицательное НРИ

Построение опорного изображения

Ключом к успеху тех методов, которые рассматривались в двух последних разделах, является наличие опорного изображения, с которым производятся дальнейшие сравнения. Для разности между двумя кадрами в последовательности динамических изображений характерно подавление всех неподвижных составляющих, в результате чего остаются лишь те элементы изображения, которые соответствуют движущимся объектам и шуму.

На практике не всегда можно получить опорное изображение, состоящее только из неподвижных элементов, и его приходится строить на основании набора изображений, содержащих один или более движущихся объектов. В частности, такая потребность возникает в ситуациях, относящихся к производственным сценам или к условиям, когда необходимо частое обновление. Опишем одну из возможных процедур построения опорного изображения. Вначале берем в качестве опорного первое изображение последовательности. Когда некоторая подвижная составляющая целиком сдвинется за пределы своего положения в этом опорном кадре, соответствующая фоновая область текущего кадра может быть скопирована в то место опорного изображения, которое занимал движущийся объект. После того, как все движущиеся объекты полностью освободят свои первоначальные места, с помощью такой процедуры будет построено опорное изображение, в котором присутствуют только неподвижные компоненты. Как указывалось в предыдущем разделе, смещение объекта устанавливается путем наблюдения за изменениями в положительном НРИ для тех объектов, яркость которых выше фона, и в отрицательном НРИ для объектов, яркость которых ниже фона.

Пример 10.27. Построение опорного изображения.

■ На рис. 10.60(*а*) и (*б*) приведены два изображения перекрестка с интенсивным движением. Первое изображение рассматривается как опорное, а второе представляет ту же сцену спустя некоторое время. Цель состоит в том, чтобы убрать из опорного изображения крупные движущиеся объекты, создав тем самым статичное изображение. Хотя в изображении присутствуют и другие, более мелкие движущиеся объекты, главная динамическая составляющая — это автомобиль, проезжающий через перекресток слева направо. Для наглядности остановимся на этом объекте. Наблюдая за изменениями в положительном и отрицательном НРИ, можно определить начальное положение движущегося объекта, как объяснялось ранее. После того, как выявлена область, первоначально занимаемая

а б в

Рис. 10.60. Построение статичного опорного изображения. (*а*)—(*б*) Два кадра из последовательности. (*в*) Движущийся вправо автомобиль убран из кадра (*а*) путем восстановления фона из соответствующей области изображения (*б*). [Jain, Jain]

этим объектом, его можно удалить из изображения. Из кадра последовательности, на котором положительное или отрицательное НРИ перестало меняться, можно скопировать область, ранее занятую объектом, в то же место опорного изображения. Тем самым фон изображения в этом месте будет восстановлен. Если проделать то же самое для всех движущихся объектов, то в результате получим опорное изображение, состоящее только из статичных компонент; с ним и будем сравнивать последующие кадры для выявления движения. Результат удаления автомобиля, движущегося вправо, демонстрируется на рис. 10.60(*в*). ■

10.6.2. Частотные методы

В этом разделе мы рассмотрим задачу нахождения движения на основе преобразования Фурье. Рассмотрим последовательность $f(x, y, t)$, $t = 0, 1,..., K - 1$, состоящую из K цифровых изображений размерами $M \times N$, полученных неподвижной камерой. При изложении метода сначала будем предполагать, что все кадры имеют равномерный фон с нулевой яркостью, а объект представлен одиночным пикселем единичной яркости, движущимся с постоянной скоростью. Предположим, что на первом кадре ($t = 0$) объект расположен в точке (x', y'). *Спроецируем* плоскость этого изображения на ось x, что эквивалентно суммированию яркостей пикселей по строкам. В результате такой операции получим одномерный массив из M значений, из которых все равны нулю, кроме x', на которое проецируется объект. Если теперь умножить все элементы массива на $\exp(i2\pi a_1 x\Delta t)$, $x = 0, 1, 2,..., M - 1$, и просуммировать результаты, получим единственный член, равный $\exp(i2\pi a_1 x'\Delta t)$, где a_1 — положительное целое число, а Δt — промежуток времени между соседними кадрами.

Предположим, что во втором кадре ($t = 1$) объект передвинулся в точку с координатами $(x'+1, y')$, т. е. на один пиксель вдоль оси x. Повторяя ту же процедуру проецирования, получим сумму, равную $\exp[i2\pi a_1(x'+1)\Delta t]$. Если объект продолжает двигаться в том же направлении и с той же скоростью 1 пиксель за кадр, то в любой целочисленный момент времени результат будет равен $\exp[i2\pi a_1(x'+t)\Delta t]$, что по формуле Эйлера можно выразить в виде

$$e^{i2\pi a_1(x'+t)\Delta t} = \cos\left[2\pi a_1(x'+t)\Delta t\right] + i\sin\left[2\pi a_1(x'+t)\Delta t\right] \qquad (10.6\text{-}5)$$

для $t = 0, 1,..., K - 1$. Другими словами, описанная процедура дает в результате комплексную синусоиду с частотой a_1. Если за время между кадрами объект сдвигается вдоль оси x на V_1 пикселей, то синусоида будет иметь частоту $V_1 a_1$. Поскольку t меняется с целым шагом от 0 до $K - 1$, то, ограничивая a_1 только целочисленными значениями, получим комплексную синусоиду, дискретное преобразование Фурье которой будет иметь два пика: один на частоте $V_1 a_1$, а другой на частоте $K - V_1 a_1$. Последний появляется вследствие симметрии ДПФ, о чем говорилось в разделе 4.6.4, и может не приниматься во внимание. Итак, в спектре Фурье имеется пик на частоте $V_1 a_1$. Таким образом, при делении этой величины на a_1 получим V_1, т. е. составляющую скорости по оси x в пикселях за кадр (частота кадров предполагается известной). Аналогичными рассуждениями можно получить значение V_2 — составляющую скорости движения объекта вдоль оси y.

Если в последовательности изображений не происходит никакого движения, то всем кадрам соответствуют одинаковые экспоненциальные члены временно́й последовательности, Фурье-преобразование которой будет состоять из одиночного пика при частоте 0 (т. е. только члена с постоянной составляющей). Поскольку все до сих пор рассмотренные операции являются линейными, то в общем случае, когда имеется один или более объектов любой формы, движущихся относительно произвольного неподвижного фона, в результате Фурье-преобразования получим пик в нуле, отвечающий статическим компонентам изображения, и пики в точках, координаты которых пропорциональны скоростям движения объектов.

Эти идеи в сжатом виде можно изложить следующим образом. Для последовательности из K цифровых изображений размерами $M \times N$ взвешенная сумма проекций на ось x в любой целочисленный момент времени равна

$$g_x(t,a_1) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y,t)e^{i2\pi a_1 x\Delta t} \quad t = 0,1,\ldots,K-1. \tag{10.6-6}$$

Аналогично сумма проекций на ось y равна

$$g_y(t,a_2) = \sum_{y=0}^{N-1}\sum_{x=0}^{M-1} f(x,y,t)e^{i2\pi a_2 y\Delta t} \quad t = 0,1,\ldots,K-1, \tag{10.6-7}$$

где, как уже отмечалось, a_1 и a_2 — положительные целые числа. Применяя одномерное преобразование Фурье к функциям (10.6-6) и (10.6-7), соответственно получаем:

$$G_x(u_1,a_1) = \sum_{t=0}^{K-1} g_x(t,a_1)e^{-i2\pi u_1 t/K} \quad u_1 = 0,1,\ldots,K-1 \tag{10.6-8}$$

и

$$G_y(u_2,a_2) = \sum_{t=0}^{K-1} g_y(t,a_2)e^{-i2\pi u_2 t/K} \quad u_2 = 0,1,\ldots,K-1. \tag{10.6-9}$$

На практике для вычисления этих преобразований используется алгоритм БПФ, как описано в разделе 4.11.

Связь между частотой и скоростью устанавливается следующими соотношениями:

$$u_1 = a_1 V_1 \tag{10.6-10}$$

и

$$u_2 = a_2 V_2. \tag{10.6-11}$$

В этих формулах скорость измеряется в единицах пикселей за время всей последовательности. Например, значение $V_1 = 10$ интерпретируется как сдвиг на 10 пикселей за время K кадров. Если кадры снимаются через равные интервалы, то реальная физическая скорость объекта зависит от частоты кадров и расстояния между пикселями. Так, при $V_1 = 10$, $K = 30$, частоте кадров, равной 2 изображениям в секунду, и шагу пикселей, соответствующему расстоянию 0,5 м, реальная физическая скорость в направлении оси x оказывается равной

$$V_1 = (10 \text{ пикселей})(0{,}5 \text{ м/пиксель})(2 \text{ кадр/с}) \,/\, (30 \text{ кадров}) = 1/3 \text{ м/с}.$$

Для нахождения знака составляющей скорости в направлении оси x вычислим выражения

$$S_{1x} = \left.\frac{d^2 \operatorname{Re}[g_x(t,a_1)]}{dt^2}\right|_{t=n} \tag{10.6-12}$$

и

$$S_{2x} = \left.\frac{d^2 \operatorname{Im}[g_x(t,a_1)]}{dt^2}\right|_{t=n}. \tag{10.6-13}$$

Поскольку g_x изменяется по синусоидальному закону, можно показать (задача 10.47), что S_{1x} и S_{2x} будут иметь одинаковые знаки в любой момент времени n, если составляющая скорости V_1 (вдоль оси x) положительна. Наоборот, разные знаки S_{1x} и S_{2x} показывают, что эта компонента отрицательна. Если какая-либо из величин S_{1x} и S_{2x} принимает нулевое значение, то рассматривается ближайший соседний момент времени $t = n \pm \Delta t$. Аналогичным способом определяется и знак составляющей по оси y (для скорости V_2).

Пример 10.28. Нахождение малых движущихся объектов с помощью частотных методов.

■ Рис. 10.61—10.64 демонстрируют эффективность изложенного выше подхода. На рис. 10.61 приведено одно изображение из последовательности 32 кадров, полученных путем добавления белого шума к опорному изображению, снятому спутником LANDSAT. На эту последовательность был наложен интересующий объект, движущийся со скоростью 0,5 пикселя на кадр вдоль оси x и 1 пиксель на кадр — вдоль оси y. Этот объект, обведенный кружком на рис. 10.62, имеет гауссово распределение яркости на малой площади (9 пикселей) и трудно различим глазом. Результаты вычисления величин (10.6-8) и (10.6-9) при $a_1 = 6$ и $a_2 = 4$ показаны на рис. 10.63 и 10.64 соответственно. Пик при $u_1 = 3$ на рис. 10.63 дает, согласно соотношению (10.6-10), значение $V_1 = 0{,}5$. Аналогично пик при $u_2 = 4$ на рис. 10.64 дает, согласно соотношению (10.6-11), значение $V_2 = 1{,}0$. ■

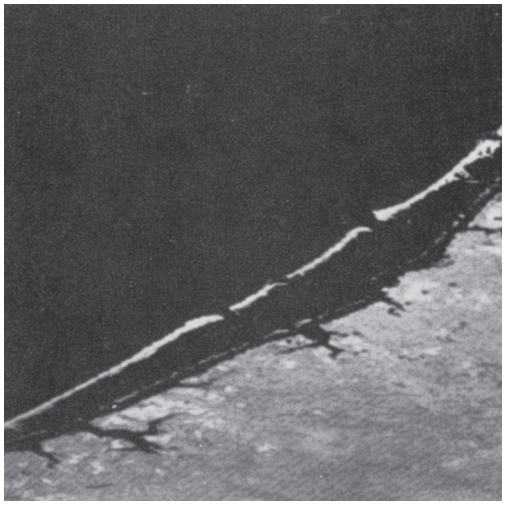


Рис. 10.61. Кадр, полученный со спутника LANDSAT. [Cowart, Snyder, Ruedger]

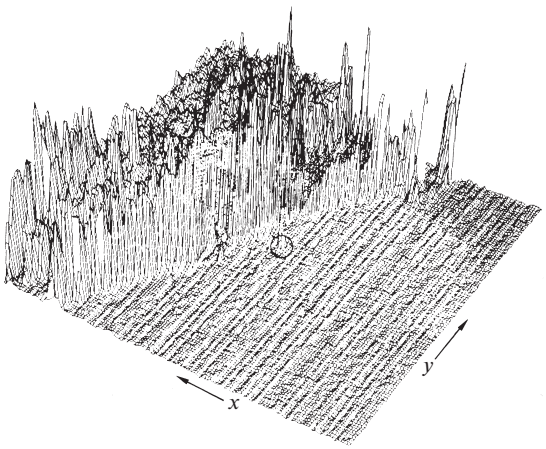


Рис. 10.62. Двумерный график яркости изображения на рис. 10.61, на котором интересующий объект обведен кружком. [Rajala, Riddle, Snyder]

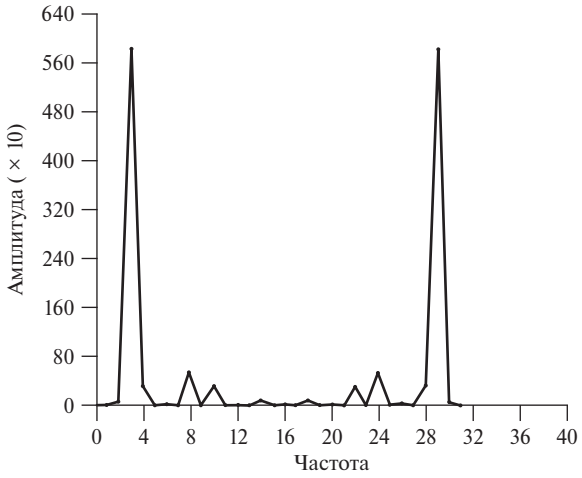


Рис. 10.63. Спектр согласно (10.6-8), на котором виден пик при $u_1 = 3$. [Rajala, Riddle, Snyder]

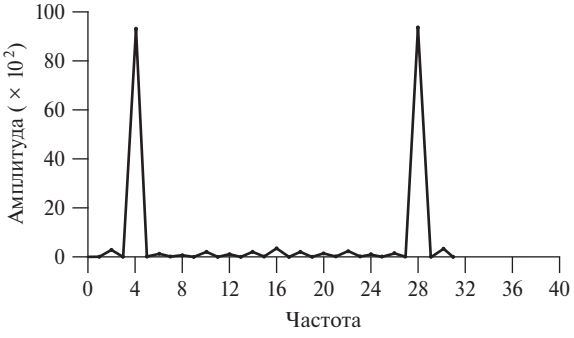


Рис. 10.64. Спектр согласно (10.6-9), на котором виден пик при $u_2 = 4$. [Rajala, Riddle, Snyder]

Правила выбора значений a_1 и a_2 можно объяснить с помощью рисунков 10.63 и 10.64. Предположим, например, что вместо $a_2 = 4$ мы использовали бы значение $a_2 = 15$. В этом случае, поскольку $V_2 = 1{,}0$, пики на рис. 10.64 находились бы в точках $u_2 = 15$ и $u_2 = 17$, что в результате привело бы к значительной ошибке, связанной с наложением спектров. Как отмечалось в разделе 4.5.4, причиной таких ошибок является недостаточная частота дискретизации (применительно к данному рассмотрению — слишком малое число кадров, поскольку диапазон для u зависит от K). Поскольку $u = aV$, одна из возможностей состоит в выборе в качестве a ближайшего целочисленного значения к величине $a = u_{max}/V_{max}$, где u_{max} — максимальная частота, определяемая величиной K, а V_{max} — максимальная ожидаемая скорость объекта.

Заключение

Сегментация изображения является важным предварительным шагом большинства систем автоматического распознавания зрительных образов и анализа сцен. Как показывают представленные в этой главе примеры, выбор того или иного метода сегментации диктуется по большей части специфическими особенностями рассматриваемой задачи. Хотя набор методов, изложенных в этой главе, далеко не исчерпывающий, они являются характерными представителями техники сегментации, обычно применяемой на практике. Приводимые ниже ссылки могут использоваться как основа для дальнейшего изучения данной темы.

Ссылки и литература для дальнейшего изучения

Благодаря центральной роли, которую играет сегментация при автоматической обработке изображений, тема сегментации представлена в большинстве книг, связанных с обработкой изображений, анализом сцен и машинным зрением. В качестве книг, которые можно рекомендовать для дополнительного изучения материала, изложенного в данной главе, укажем следующие: [Umbaugh, 2005], [Davies, 2005], [Gonzalez,Woods, Eddins, 2004], [Shapiro, Stockman, 2001], [Sonka et al., 1999] и [Petrou, Bosdogianni, 1999].

Работы, в которых рассматривается применение масок для нахождения разрывов яркости (раздел 10.2), имеют долгую историю. За эти годы было предложено большое число масок: [Roberts, 1965], [Prewitt, 1970], [Kirsh, 1971], [Robinson, 1976], [Frei, Chen, 1977] и [Canny, 1986]. В обзорной статье [Fram, Deutsch, 1975] исследуется большое число масок и дается оценка их характеристик. Вопрос о качестве масок, особенно в плане обнаружения контуров, по-прежнему вызывает значительный интерес, примером чему могут служить работы [Qian, Huang, 1996], [Wang et al., 1996], [Heath et al., 1997, 1998] и [Ando, 2000]. В большинстве прикладных задач, где используются многодиапазонные сенсоры, все более популярным становится обнаружение контуров на цветных изображениях. См., например, работы [Salinas, Abidi, Gonzalez, 1996], [Zugaj, Lattuati, 1998],

[Mirmehdi, Petrou, 2000] и [Plataniotis, Venetsanopoulos, 2000]. В настоящее время также вызывает интерес взаимосвязь между характеристиками изображения и качеством работы масок; укажем как пример работу [Ziou, 2001]. Наше изложение в этой главе свойств лапласиана в отношении пересечения нулевого уровня основано на статье [Marr, Hildredth, 1980] и книге [Marr, 1982]. См. также статью [Clark, 1989] по поводу проверки правильности контуров, найденных с помощью алгоритма пересечения нулевого уровня (отдельные исправления к этой статье были даны в работе [Piech, 1990]). Как отмечалось в разделе 10.2, использование пересечения нулевого уровня лапласиана гауссиана является важным подходом, характеристики которого в настоящее время активно изучаются ([Gunn, 1998, 1999]). Как следует из названия, детектор контуров Кэнни, рассматривавшийся в разделе 10.2.6, обязан своим появлением работе [Canny, 1986]. Как пример работы по этой тематике спустя двадцать лет укажем [Zhang, Rockett, 2006].

Преобразование Хафа [Hough, 1962] является практичным методом обнаружения кривых и глобальных связей между пикселями. За прошедшее время были предложены многочисленные обобщения базового преобразования, рассмотренного в этой главе. Например, в работе [Lo, Tsai, 1995] рассматривается подход к обнаружению толстых линий, работы [Guil et al., 1995, 1997] касаются быстрой реализации преобразования Хафа и методов обнаружения элементарных кривых. Дальнейшие обобщения для обнаружения эллиптических дуг обсуждаются в [Daul et al., 1998], а в работе [Shapiro, 1996] рассматривается применение преобразования Хафа к полутоновым изображениям.

Как отмечалось в начале раздела 10.3, методы пороговой обработки завоевали значительную популярность ввиду простоты реализации. Неудивительно, что эта тематика широко представлена в литературе. Хорошую предварительную оценку этих публикаций можно получить из обзорных статей [Sahoo et al., 1988] и [Lee et al., 1990]. Помимо методов, рассмотренных в данной главе, другие существующие подходы для компенсации влияния освещения и отражения демонстрируются в работах [Perez, Gonzalez, 1987], [Parker, 1991], [Murase, Nayar, 1994], [Bischsel, 1998], [Drew et al., 1999] и [Toro, Funt, 2007]. Дополнительный материал к разделу 10.3.2 приводится в книге [Jain et al., 1995].

Примером ранних работ по пороговым преобразованиям с оптимальным глобальным порогом (раздел 10.3.3) является классическая статья [Chow, Kaneko, 1972] (мы рассмотрим этот метод в разделе 12.2.2 в более общем контексте распознавания объектов). Хотя теоретически этот метод является оптимальным, возможности его применения для градационных преобразований ограничены, поскольку для этого требуется оценивать функции плотности вероятности. Оптимальный метод, изложенный в разделе 10.3.3, принадлежащий Оцу [Otsu, 1979], получил значительно большее признание, поскольку у него отличное качество сочетается с простотой реализации (требуется вычислить только гистограмму изображения). Простая идея использования предобработки (разделы 10.3.4 и 10.3.5) восходит к ранней статье [White, Rohrer, 1983], где для решения сложной задачи сегментации используется сочетание порогового преобразования, градиента и лапласиана. Интересно сравнить фундаментальное сходство в плане возможностей сегментации между этими двумя статьями и работами в области пороговых преобразований, проделанными почти 20 лет спу-

стя ([Cheriet et al., 1998], [Sauvola, Pietikainen, 2000], [Liang et al., 2000] и [Chan et al., 2000]). Дополнительный материал о пороговых преобразованиях с несколькими порогами (раздел 10.3.6) можно найти в работах [Yin, Chen, 1997], [Liao et al., 2001] и [Zahara et al., 2005]. Дополнительный материал о пороговых преобразованиях с переменным порогом (раздел 10.3.7) имеется в работе [Parker, 1997]. См. также [Delon et al., 2007].

Давний обзор по теме сегментации на основе областей содержится в статье [Fu, Mui, 1981]. Работы [Haddon, Boyce, 1990] и [Pavlidis, Liow, 1990] относятся к числу первых попыток объединить для целей сегментации характеристики областей и границ. Интересен также более поздний подход к выращиванию областей, предложенный в статье [Hojjatoleslami, Kittler, 1998]. Современное состояние основных концепций, связанных с сегментацией на основе областей, излагается в книгах [Shapiro, Stockman, 2001] и [Sonka et al., 1999].

Как было показано в разделе 10.5, мощную идею представляет собой сегментация по водоразделам. Ранними работами по этой теме были книга [Serra, 1988], а также работы [Beucher, 1990] и [Beucher, Meyer, 1992]. В статье [Baccar et al., 1996] рассматривается сегментация на основе морфологических водоразделов и слияния данных. Прогресс в этой области, достигнутый спустя десять лет, демонстрирует специальный выпуск журнала [*Pattern Recognition*, 2000], целиком посвященный данной теме. Как указывалось в разделе 10.5, одной из основных трудностей при использовании водоразделов является избыточная сегментация. Существующие подходы к решению этой проблемы демонстрируют статьи [Najmanand, Schmitt, 1996], [Haris et al., 1998] и [Bleau, Leon, 2000]. В статье [Bieniek, Moga, 2000] обсуждается алгоритм сегментации по водоразделам, основанный на компонентах связности.

Материал раздела 10.6.1 взят из статьи [Jain, R., 1981]. См. также книгу [Jain, Kasturi, Schunck, 1995]. Материал раздела 10.6.2 взят из статьи [Rajala, Riddle, Snyder, 1983]. См. также статьи [Shariat, Price, 1990] и [Cumani et al., 1991]. В книгах [Sonka et al., 1999], [Shapiro, Stockman, 2001], [Snyder, Qi, 2004] и [Davies, 2005] приводится дополнительный материал по оцениванию движения. См. также [Alexiadis, Sergiadis, 2007].

Задачи

★10.1 Докажите справедливость равенства (10.2-2). (*Подсказка:* воспользуйтесь разложением в ряд Тейлора, оставляя только линейные члены.)

★10.2 Двоичное изображение содержит прямые линии, ориентированные горизонтально, вертикально и под углами 45° и −45°. Приведите набор масок 5×5 для обнаружения в этих линиях разрывов шириной в один пиксель. Считайте, что линии имеют яркость 1, а фон 0.

10.3 Предложите метод обнаружения промежутков длиной от 2 до K пикселей на отрезках прямых в двоичном изображении. Считайте, что ширина линий составляет 1 пиксель. Предлагаемый метод должен основываться на анализе связности для восьмерки соседей, а не попытках построения масок для обнаружения разрывов.

10.4 Ответьте по рис. 10.7 на следующие вопросы.

★ **(а)** Некоторые из линий, соединяющих центральный элемент с контактными площадками, на рис. 10.7(*д*) превратились в одиночные линии, а другие стали двойными линиями. Объясните, почему.

(б) Предложите способ удаления тех компонент на рис. 10.7(*е*), которые являются частями прямых, идущих под углом −45°?

10.5 Рассмотрим модели яркостного перепада на рис. 10.8.

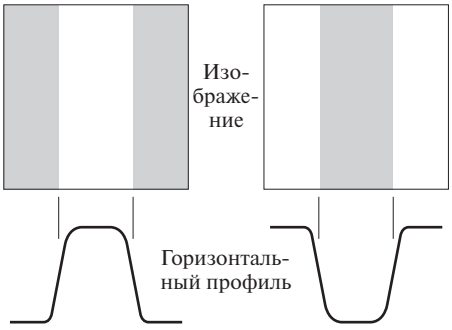
★ **(а)** Предположим, что для каждой модели вычислен модуль градиента с помощью масок оператора Превитта, показанных на рис. 10.14. Изобразите вид горизонтального профиля через середину каждого градиентного изображения.

(б) Изобразите для каждой модели горизонтальный профиль направления градиента.

(*Указание:* для ответа на вопросы не стройте изображения модуля и направления градиента. Просто изобразите профили, соответствующие *ожидаемому* виду изображений модуля и направления градиента.)

10.6 Рассмотрите горизонтальный профиль яркости через середину двоичного изображения, которое содержит идущий вертикально ступенчатый перепад в центре изображения. Нарисуйте вид профиля после сглаживания этого изображения усредняющей маской $n \times n$, коэффициенты которой равны $1/n^2$. Для простоты предполагайте, что сглаженное изображение промасштабировано по яркости так, что слева от перепада яркость равна 0, а справа 1. Считайте также, что размеры маски значительно меньше размеров изображения, так что краевые эффекты не затрагивают область перепада в середине профиля яркости.

★**10.7** Предположим, что вместо модели линейного перепада, приведенной на рис. 10.10, применяется модель, показанная на рисунке ниже. Нарисуйте вид градиента и лапласиана для каждого из профилей.



10.8 Используя рис. 10.14, ответьте на следующие вопросы.

(а) Предположим, что для получения составляющих g_x и g_y используются маски Собела. Покажите, что в этом случае вычисление градиента по формулам (10.2-10) и (10.2-20) приведет к идентичным результатам.

(б) Покажите, что то же самое справедливо и для масок Превитта.

★**10.9** Покажите, что маски Собела и Превитта дают при вычислении градиента изотропный результат только для горизонтальных и вертикальных контуров, а также для проходящих под углами ±45°.

10.10 Результаты, достигаемые одиночным проходом по изображению с использованием некоторых двумерных масок, также могут быть получены двумя проходами с использованием одномерных масок. Например, тот же результат, что и при обработке сглаживающей маской 3×3 с коэффициентами 1/9, можно получить одиночным проходом по изображению с маской [1 1 1], а затем одиночным проходом с маской

$$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

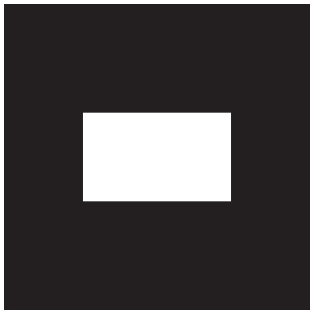
Окончательный результат получается затем масштабированием по яркости с коэффициентом 1/9. Покажите, что маски Превитта (рис. 10.14) можно реализовать однократным проходом с *дифференцирующей* маской вида [–1 0 1] (или такой же транспонированной), а затем проходом со *сглаживающей* маской вида [1 1 1] (или такой же транспонированной).

★**10.11** Для измерения градиентов контуров, ориентированных в восьми главных направлениях компаса (С, СВ, В, ЮВ, Ю, ЮЗ, З, СЗ), строятся так называемые *компасные градиентные операторы* размерами 3×3.

★ **(а)** Приведите вид каждого из этих восьми операторов, используя значения коэффициентов 0, 1, –1, 2 или –2.

(б) Укажите для каждой маски направление градиента, учитывая, что оно перпендикулярно направлению перепада.

10.12 Прямоугольник в центре изображения, приведенного на рисунке, имеет размеры $m \times n$ пикселей.



(а) Как будет выглядеть модуль градиента этого изображения при использовании приближения, даваемого соотношением (10.2-20)? Считайте, что составляющие g_x и g_y получаются с помощью операторов Превитта. Укажите на градиентном изображении все различные значения пикселей.

(б) Нарисуйте гистограмму *направлений* перепадов, вычисленных с помощью соотношения (10.2-11). Укажите точные значения высот всех пиков гистограммы.

(в) Как будет выглядеть лапласиан этого изображения при использовании приближения, даваемого соотношением (10.2-7)? Укажите на изображении лапласиана все различные значения пикселей.

10.13 Пусть в результате свертки изображения $f(x, y)$ с маской $n \times n$ (все элементы которой равны $1/n^2$) формируется сглаженное изображение $\bar{f}(x, y)$.

★ **(а)** Выведите выражение для *крутизны контура* (степени наклона перепада) на сглаженном изображении как функции от размера маски. Для простоты предполагайте, что n — нечетное число и перепады находятся с помощью частных производных

$$\partial\bar{f}/\partial x = \bar{f}(x+1,y) - \bar{f}(x,y) \quad \text{и} \quad \partial\bar{f}/\partial y = \bar{f}(x,y+1) - \bar{f}(x,y).$$

(б) Покажите, что отношение максимальной крутизны контура на сглаженном и на исходном изображениях равно $1/n$, т. е. крутизна контура обратно пропорциональна размеру сглаживающей маски.

10.14 Для уравнения (10.2-23):

★ **(а)** Покажите, что среднее значение оператора лапласиана гауссиана $\nabla^2 G(x,y)$ равно нулю.

(б) Докажите, что среднее значение любого изображения после свертки с этим оператором также равно нулю. (*Подсказка:* попробуйте решить эту задачу в частотной области, используя теорему о свертке и тот факт, что среднее значение функции пропорционально значению ее Фурье-преобразования в начале координат.)

(в) Будет ли в общем случае справедливо утверждение (б), если (1) для вычисления лапласиана от гауссова фильтра низких частот использовать показанную на рис. 10.4(*а*) маску лапласиана 3×3 и (2) выполнить свертку полученного результата с любым изображением? Ответ объясните. (*Подсказка:* сошлитесь на результат задачи 3.16.)

10.15 Обратитесь к рис. 10.22(*в*).

(а) Объясните, почему линии перепадов образуют замкнутые контуры.

★ **(б)** Всегда ли метод нахождения контуров с помощью пересечения нулевого уровня будет давать в результате замкнутые контуры? Ответ обоснуйте.

10.16 В литературе часто можно найти вывод лапласиана гауссиана, исходя из выражения

$$G(r) = e^{-r^2/2\sigma^2},$$

где $r^2 = x^2 + y^2$. Лапласиан гауссиана затем находится вычислением второй производной $\nabla^2 G(r) = \partial^2 G/\partial r^2$. Окончательно вместо r^2 подставляется $x^2 + y^2$, чтобы получить (ошибочно) результат

$$\nabla^2 G(x,y) = \left[(x^2+y^2-\sigma^2)/\sigma^4\right]\exp\left[-(x^2+y^2)/2\sigma^2\right].$$

Выведите этот результат и объясните причину отличия данного выражения от (10.2-23).

10.17 **(а)** Выведите соотношение (10.2-27).

(б) Обозначим $k = \sigma_1/\sigma_2$ отношение стандартных отклонений, которое рассматривалось в связи с функцией разности гауссианов. Выразите соотношение (10.2-27) в обозначениях k и σ_2.

10.18 Пусть G и f — массивы с размерами $n\times n$ и $M\times N$ соответственно.

★ **(а)** Покажите, что двумерную свертку гауссовой функции $G(x,y)$ вида (10.2-21) с изображением $f(x,y)$ можно вычислить через одномерную свертку вдоль строк (или столбцов) $f(x,y)$ и затем одномерную свертку

вдоль столбцов (строк) полученного результата. (Дискретная свертка рассматривается в разделе 3.4.2.)

(б) Выведите выражение для вычислительного выигрыша (выражаемого отношением числа операций умножения) от использования одномерной свертки в пункте (а) по сравнению с непосредственной реализацией двумерной свертки.

★**10.19** **(а)** Покажите, что шаги 1 и 2 алгоритма Марра—Хилдрета можно реализовать с помощью четырех операций одномерной свертки. (*Подсказки:* сошлитесь на задачу 10.18(*а*) и выразите оператор Лапласа в виде суммы двух частных производных, заданных формулами (10.2-5) и (10.2-6), после чего реализуйте вычисление каждой производной с помощью одномерной маски, как в задаче 10.10.)

(б) Выведите выражение для вычислительного выигрыша (выражаемого отношением числа операций умножения) от использования одномерной свертки в пункте (а) по сравнению с непосредственной реализацией двумерной свертки. (Размеры массивов G и f те же, что и в задаче 10.18.)

10.20 **(а)** Сформулируйте шаг 1 и построение изображения модуля градиента на шаге 2 в алгоритме Кэнни с использованием одномерных сверток вместо двумерной.

(б) Найдите вычислительный выигрыш (выраженный отношением числа операций умножения) от использования одномерных сверток вместо непосредственной реализации двумерной свертки. Считайте, что двумерный гауссов фильтр на шаге 1 представлен массивом размерами $n \times n$, а исходное изображение имеет размеры $M \times N$.

10.21 Рассмотрите три модели вертикальных перепадов и соответствующие им профили на рис. 10.8.

★ **(а)** Предположим, что модуль градиента для каждой модели перепада вычисляется с помощью масок оператора Превитта. Нарисуйте горизонтальный профиль яркости для трех градиентных изображений.

★ **(б)** Нарисуйте горизонтальный профиль яркости для трех изображений лапласиана, вычисленных с помощью маски 3×3, показанной на рис. 10.4(*а*).

★ **(в)** Сделайте то же самое для изображений, полученных в результате выполнения двух первых шагов детектора контуров Марра—Хилдрета.

(г) Сделайте то же самое для изображений, полученных в результате выполнения двух первых шагов детектора контуров Кэнни. Направление градиента можно не учитывать.

(д) Нарисуйте горизонтальный профиль яркости для трех изображений направления градиента в детекторе контуров Кэнни. (*Указание:* дайте ответ без построения изображений. Просто изобразите профили, соответствующие *ожидаемому* виду изображений направления градиента.)

10.22 Применительно к преобразованию Хафа, рассмотренному в разделе 10.2.7,

(а) Разработайте общую процедуру построения представления прямой с помощью нормали по уравнению в форме с угловым коэффициентом $y = ax + b$.

★ **(б)** Найдите представление прямой $y = -3x + 2$ с помощью нормали.

★**10.23** Используя материал раздела 10.2.7:

(а) Объясните, почему в результате преобразования Хафа точка 1 на рис. 10.33(*а*) отображается в прямую на рис. 10.33(*б*).

(б) Только ли эта точка будет давать такой результат преобразования? Ответ объясните.

(в) Объясните зеркальную симметричность точек пересечения кривых с правой и левой границами области на рис. 10.33(*б*), например, для кривой с пометкой *Q*.

10.24 Покажите, что количество операций, необходимых для реализации схемы с ячейками накопления, рассмотренной в разделе 10.2.7, линейно по n — числу не относящихся к фону точек исходного изображения.

10.25 Важной областью применения методов сегментации изображений является обработка снимков, полученных с помощью пузырьковых камер. Эти изображения регистрируются в ходе экспериментов в физике высоких энергий, когда пучками частиц с известными свойствами бомбардируются известные ядра. Типичное событие на снимке состоит из треков входящих частиц, один из которых в случае соударения с ядром разветвляется на треки вторичных частиц, испускаемых в точке соударения. Предложите метод сегментации изображения для обнаружения всех треков, состоящих не менее чем из 100 пикселей и наклоненных к горизонтали под одним из следующих углов: ±20°, ±40° и ±60°. Допустимая ошибка при оценке угла любого из этих направлений не должна превышать ±5°. Чтобы трек считался достоверным, его длина должна быть не менее 100 пикселей при наличии не более трех разрывов, каждый длиной до 10 пикселей. Предполагается, что изображения прошли предварительную обработку, в результате чего они преобразованы в двоичные, и все треки всюду имеют ширину 1 пиксель, кроме, быть может, точки соударения. Предлагаемая процедура должна быть в состоянии различать треки одинакового направления, но исходящие из разных точек. (*Подсказка*: стройте решение на основе преобразования Хафа.)

★**10.26** Переформулируйте простой алгоритм нахождения глобального порога из раздела 10.3.2 так, чтобы в нем использовалась гистограмма исходного изображения, а не само изображение.

★**10.27** Докажите, что простой алгоритм нахождения глобального порога из раздела 10.3.2 сходится за конечное число шагов. (*Подсказка:* воспользуйтесь построенной в задаче 10.26 формулировкой алгоритма на основе гистограммы.)

10.28 Объясните, почему начальное значение порога в простом алгоритме из раздела 10.3.2 должно выбираться в интервале между минимальной и максимальной яркостью изображения. (*Подсказка:* постройте пример, демонстрирующий отказ алгоритма в результате выбора начального значения порога вне указанного интервала.)

★**10.29** Зависит ли величина порога, найденного с помощью простого алгоритма из раздела 10.3.2, от выбора начального значения? Если нет — докажите это, если да — приведите пример.

10.30 В обоих случаях ниже можно предполагать, что значение порога находится внутри открытого интервала $(0, L-1)$.

★ **(а)** Докажите, что если гистограмма изображения равномерна по всевозможным значениям яркости, то простой алгоритм нахождения глобального порога из раздела 10.3.2 сходится к среднему значению яркости изображения $(L-1)/2$.

(б) Докажите, что если гистограмма изображения содержит две одинаковые моды, которые симметричны относительно своих средних значений, то простой алгоритм нахождения глобального порога сходится к точке, находящейся посередине между средними значениями этих мод.

10.31 Обратитесь к алгоритму нахождения величины порога, предложенному в разделе 10.3.2. Пусть имеется задача, в которой гистограмма является бимодальной и, более того, форма мод приближенно описывается гауссовыми кривыми вида $A_1 \exp[-(z-m_1)^2/2\sigma_1^2]$ и $A_2 \exp[-(z-m_2)^2/2\sigma_2^2]$. Предполагая, что $m_1 > m_2$ и начальное значение T выбирается между максимальной и минимальной яркостью изображения, укажите требования, которым должны удовлетворять параметры этих кривых, чтобы после сходимости алгоритма выполнялись следующие условия:

(а) Значение порога равно $(m_1 + m_2)/2$.

(б) Порог находится справа от m_1.

(в) Порог находится в интервале $(m_1 + m_2)/2 < T < m_1$.

Если какое-то из трех условий невозможно, укажите это и обоснуйте свой ответ.

★**10.32** **(а)** Объясните, как первое равенство в формуле (10.3-15) следует из соотношений (10.3-14), (10.3-10) и (10.3-11).

(б) Объясните, почему второе равенство в формуле (10.3-15) следует из первого.

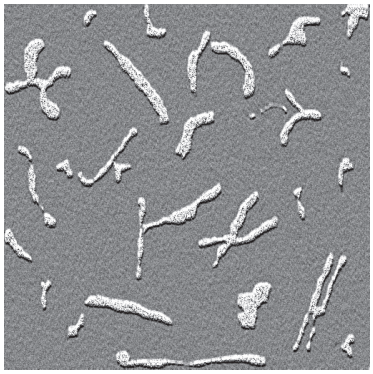
10.33 Покажите, что для k в интервале $0 \le k < L-1$ всегда существует максимум в формуле (10.3-18).

10.34 Обратитесь к формуле (10.3-20) и предложите аргументы в обоснование того, что $0 \le \eta(k) \le 1$ для k в интервале $0 \le k < L-1$, причем минимум достигается только для изображений с одинаковой яркостью во всех точках, а максимум — только для двухградационных изображений со значениями яркости 0 и $L-1$.

★**10.35** **(а)** Пусть изображение $f(x, y)$ имеет диапазон яркостей [0, 1] и с помощью порога T на нем успешно сегментированы объекты и фон. Покажите, что порог величиной $T' = 1 - T$ будет успешно сегментировать негатив $f(x, y)$ на те же области. Термин *негатив* используется здесь в смысле определения в разделе 3.2.1.

(б) Функция преобразования яркости в пункте (а), которая превращает изображение в его негатив, является линейной функцией с отрицательным угловым коэффициентом. Сформулируйте условия, которым должна удовлетворять произвольная функция градационного преобразования, чтобы сохранялась сегментация исходного изображения, полученная при пороге T. Каким должно быть значение порога после такого градационного преобразования?

★**10.36** На приводимом рисунке пиксели объектов и фона имеют средние значения яркости соответственно 180 и 70 по шкале [0, 255]. Изображение искажено гауссовым шумом с нулевым средним и стандартным отклонением 10 градаций яркости. Предложите для такого изображения метод пороговой сегментации, выделяющий объекты из фона с вероятностью 90 % или выше. (Помните, что 99,7 % площади под гауссовой кривой лежит в интервале $\pm 3\sigma$ относительно среднего значения, где σ — стандартное отклонение.)



10.37 Обратитесь к показанному на рис. 10.37(*б*) изображению наклонного перепада и алгоритму скользящего среднего, изложенному в разделе 10.3.7. Считайте, что изображение имеет размеры 400×700 пикселей и диапазон значений от 0 до 1, причем нули содержатся только в первом столбце.

★ **(а)** Каковы будут результаты сегментации этого изображения алгоритмом скользящего среднего при использовании $b = 0$ и произвольного значения n? Опишите, как будет выглядеть изображение.

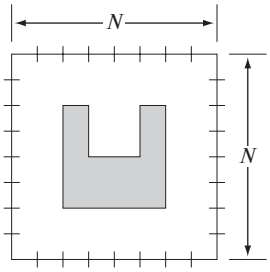
(б) Теперь измените направление наклона перепада на противоположное, так что слева находится значение 1, а справа 0, и повторите пункт (а).

(в) То же, что и в пункте (а), но при параметрах $b = 1$ и $n = 4$.

(г) То же, что и в пункте (а), но при параметрах $b = 1$ и $n = 100$.

10.38 Предложите для изображения из задачи 10.36 метод сегментации на основе выращивания областей.

★**10.39** Выполните сегментацию приведенного ниже изображения с помощью процедуры разделения и слияния областей, описанной в разделе 10.4.2. Используйте предикат следующего вида: $Q(R_i)$ = TRUE, если все пиксели области R_i имеют одинаковую яркость. Изобразите квадродерево, соответствующее Вашей сегментации.



10.40 В изображении Петли Лебедя из примера 10.24 рассмотрите область с единичными значениями, полученную в результате сегментации разреженных участков. Предложите метод использования этого изображения в качестве маски для выделения трех основных составляющих исходного изображения: (1) фона, (2) плотной внутренней области и (3) разреженной внешней области.

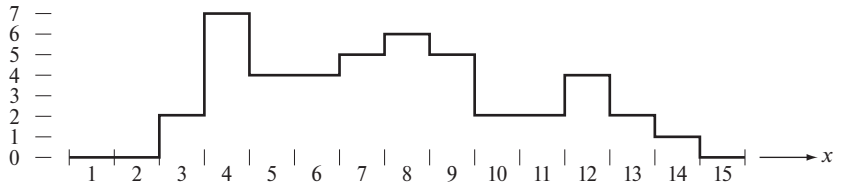
10.41 Обратитесь к материалу раздела 10.5.3.

★ **(а)** Покажите, что во время работы алгоритма сегментации по водоразделам элементы множеств $C_n(M_i)$ и $T[n]$ никогда не удаляются.

(б) Покажите, что с увеличением n число элементов в множествах $C_n(M_i)$ и $T[n]$ возрастает или остается неизменным.

10.42 В разделе 10.5 на примере было показано, что границы областей, выделенных с помощью алгоритма сегментации по водоразделам, образуют замкнутые контуры (см., например, рис. 10.56 и 10.58). Выдвиньте аргументы в подтверждение или опровержение того, что применение этого алгоритма всегда приводит к получению замкнутых границ.

★**10.43** Дайте пошаговую реализацию процедуры построения перегородок для одномерного профиля яркости, показанного на рисунке ниже. Для каждого шага процедуры нарисуйте поперечное сечение с указанием уровня «воды» и построенных перегородок.



10.44 Как выглядело бы отрицательное НРИ на рис. 10.59(*в*), если бы в соотношении (10.6-4) сравнение производилось с T, а не с $-T$?

10.45 Предполагая, что яркость движущегося объекта выше яркости фона, укажите, являются ли следующие высказывания истинными или ложными. Ответ обоснуйте.

★ **(а)** Область ненулевых элементов в абсолютном НРИ увеличивается в размерах, пока объект продолжает двигаться.

(б) Область ненулевых элементов в положительном НРИ всегда занимает одну и ту же площадь, независимо от движения, которое претерпевает объект.

(в) Область ненулевых элементов в отрицательном НРИ увеличивается в размерах, пока объект продолжает двигаться.

10.46 Предположим, что в примере 10.28 отсутствует движение вдоль оси x. Теперь объект движется лишь вдоль оси y со скоростью 1 пиксель за кадр на протяжении 28 кадров, после чего мгновенно меняет направление на противоположное и движется обратно с той же скоростью еще на протяжении 28 кадров. Как при таких условиях выглядели бы рис. 10.63 и 10.64?

★**10.47** Выдвиньте аргументы в подтверждение того, что если знаки S_{1x} и S_{1y} в формулах (10.6-12) и (10.6-13) одинаковые, то составляющая скорости V_1 положительная.

10.48 Автоматизированное фармацевтическое предприятие применяет систему обработки изображений для измерения формы таблеток в целях контроля качества. Этап сегментации в этой системе основан на методе Оцу. Скорость линии контроля настолько высокая, что для получения неподвижного изображения требуется применять импульсное освещение с очень коротким временем вспышки. Новые импульсные лампы создают равномерное освещение, однако по мере старения равномерность ухудшается и начинает зависеть от срока службы и пространственных координат в соответствии с функцией

$$i(x,y) = A(t) - t^2 e^{-\left[(x-M)^2+(y-N)^2\right]},$$

где (M, N) — центр поля зрения, а время t измеряется целым числом месяцев эксплуатации. Лампы являются экспериментальными, и зависимость $A(t)$ не вполне ясна изготовителю. Все, что известно, — это что на протяжении срока службы $A(t)$ в приведенном выше соотношении всегда больше вычитаемого члена, поскольку освещенность заведомо неотрицательна. Сделано наблюдение, что алгоритм Оцу хорошо работает при новых лампах, создающих приблизительно одинаковое освещение по всему полю изображения, однако со временем качество сегментации ухудшается. Будучи экспериментальными, лампы исключительно дорогие, и поэтому Вас подрядили в качестве консультанта для решения проблемы продления срока службы ламп с помощью вычислительных методов. Вам дана свобода выбора в части установки любых специальных маркеров или иных визуальных индикаторов по краям поля зрения камер. Предложите достаточно подробное решение, чтобы главный инженер предприятия был в состоянии понять предлагаемый Вами метод. (*Подсказка:* проанализируйте модель изображения, рассмотренную в разделе 2.3.4, и попробуйте воспользоваться небольшим объектом с известным коэффициентом отражения).

10.49 С помощью высокоскоростной съемки необходимо оценить скорость полета пули. Выбран способ, в котором используется телевизионная камера и лампа-вспышка, освещающая сцену в течение K секунд. Длина пули 3 см, диаметр 1 см, а диапазон возможных скоростей составляет 700 ± 200 м/с. Оптика камеры создает изображение размерами 256×256, на котором пуля занимает 10 % размера по горизонтали.

★ **(а)** Определите максимальное значение K, при котором расфокусировка из-за движения не превышает 1 пикселя.

(б) Определите минимальную частоту съемки (число кадров в секунду), необходимую, чтобы гарантировалось получение как минимум двух полных изображений пули во время пролета ее в поле зрения камеры.

(в) Предложите процедуру сегментации для автоматического выделения изображения пули в последовательности кадров.

(г) Разработайте способ автоматического определения скорости полета пули.

ГЛАВА 11

ПРЕДСТАВЛЕНИЕ И ОПИСАНИЕ

Сократ: Разве нам не приходилось уже много раз соглашаться, что хорошо установленные имена подобны тем вещам, которым они присвоены, и что имена — это изображения вещей?

Платон «Кратил» (пер. Т. В. Васильевой)

Введение

После того, как выполнена сегментация изображения на области, например, при помощи методов, описанных в главе 10, полученные совокупности пикселей обычно описываются и представляются в форме, удобной для последующей компьютерной обработки. По существу, при выборе способа представления областей возникает следующая альтернатива: область можно представить (1) ее внешними характеристиками (т. е. границей) или (2) внутренними характеристиками (т. е. совокупностью элементов изображения, составляющих эту область). Однако выбор способа представления является только частью задачи преобразования данных в форму, удобную для компьютерной обработки. Следующая задача состоит в том, чтобы *описать* область, исходя из выбранного способа представления. Например, область может быть *представлена* своей границей, а граница *описана* с помощью таких характеристик, как длина границы, направления прямых, соединяющих угловые точки, и число вогнутостей границы.

Внешнее представление обычно выбирается в тех случаях, когда основное внимание обращено на характеристики формы области. Внутреннее представление выбирается, если интерес представляют свойства самой области, например цвет и текстура. Иногда приходится использовать оба способа представления одновременно. В любом случае выбранные для описания признаки (дескрипторы) должны быть как можно менее чувствительными к изменению размеров области и ее перемещению по полю изображения (сдвиг, поворот). За редким исключением рассматриваемые в этой главе дескрипторы обладают всеми или некоторыми из этих свойств.

11.1. Представление

Методы сегментации, обсуждавшиеся в главе 10, дают на выходе необработанные данные в форме множества пикселей, расположенных вдоль границы или

внутри области. Обычная практика состоит в применении методов, преобразующих данные сегментации в компактное представление, способствующее более эффективному вычислению дескрипторов. В этом разделе мы рассмотрим различные подходы к представлению данных об областях.

11.1.1. Прослеживание границы

Для некоторых из рассматриваемых в этой главе алгоритмов необходимо, чтобы точки границы области были упорядочены по часовой стрелке (или наоборот). Поэтому мы начнем с изложения алгоритма прослеживания границы, который дает на выходе *упорядоченную* последовательность точек. Предполагается, что (1) мы работаем с двухградационными изображениями, в которых точки объекта и фона имеют значения 1 и 0 соответственно; и (2) изображения дополнены фоном из нулевых элементов, так что объекты не примыкают к краям изображения. Для удобства ограничимся случаем единственной области; расширение описываемого метода для случая нескольких несвязных областей осуществляется путем независимой обработки каждой области по очереди.

> Перед тем, как двигаться дальше, рекомендуется освежить в памяти материал разделов 2.5.2 и 9.5.3.

Пусть задана двоичная область (или ее граница) R. Алгоритм прослеживания границы R состоит из следующих шагов.

1. В качестве начальной точки b_0 выбирается *самая левая верхняя*[1] точка изображения, имеющая значение 1. Обозначим c_0 левого соседа b_0, как показано на рис. 11.1(*б*); ясно, что c_0 всегда является точкой фона. Рассмотрим восьмерку соседей b_0, начиная с c_0 и двигаясь по часовой стрелке. Пусть b_1 — *первая* встретившаяся точка со значением 1, а c_1 — точка фона, непосредственно ей предшествующая в указанном порядке обхода. Запоминаем положения точек b_0 и b_1 для использования на шаге 5.
2. Пусть $b = b_1$ и $c = c_1$ (см. рис. 11.1(*в*)).
3. Начиная с точки c и двигаясь по часовой стрелке, обозначим восьмерку соседей точки b через n_1, n_2,..., n_8. Находим первую точку n_k, имеющую значение 1.
4. Кладем $b = n_k$ и $c = n_{k-1}$.
5. Повторяем шаги 3 и 4 до тех пор, пока не получим, что $b = b_0$ *и* следующая найденная точка границы — b_1. Последовательность найденных точек b в момент остановки алгоритма является множеством упорядоченных точек границы.

Заметим, что на шаге 4 точка c всегда является точкой фона, потому что n_k — первая единичная точка, найденная в порядке обхода по часовой стрелке. Изложенный алгоритм иногда называют *алгоритмом Мура прослеживания грани-*

---

[1] Как говорится ниже в данной главе, самая левая верхняя точка границы обладает тем важным свойством, что аппроксимирующий границу многоугольник является выпуклым в этой точке. Кроме того, левый и верхний соседние пиксели этой точки гарантированно принадлежат фону изображения. Эти свойства делают ее хорошей «стандартной» начальной точкой для алгоритма прослеживания границы.

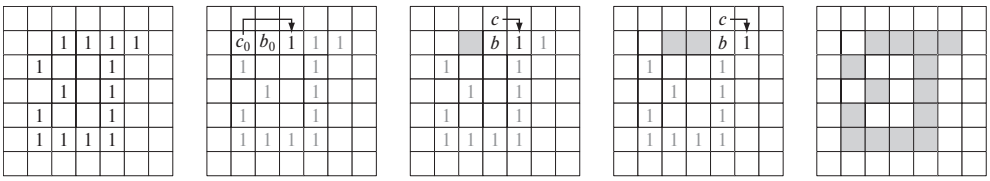


Рис. 11.1. Иллюстрация нескольких первых шагов алгоритма прослеживания границы. Точка, которая будет обрабатываться следующей, выделена жирным шрифтом, а уже найденные точки отмечены серыми квадратами

цы [Moore, 1968]. Правило остановки на шаге 5 алгоритма в литературе часто формулируют неправильно как «остановиться после того, как снова встретилась точка b_0». Как показано чуть ниже, это может привести к неверным результатам.

На рис. 11.1 показаны несколько первых шагов описанного алгоритма прослеживания границы. Нетрудно проверить, что, продолжая данную процедуру, мы получим приведенную на рис. 11.1(*е*) правильную границу, точки которой образуют последовательность, упорядоченную по часовой стрелке.

Чтобы проверить необходимость того правила остановки, которое было сформулировано на шаге 5 алгоритма, рассмотрим границу, показанную на рис. 11.2. Верхний участок этой границы мог возникнуть, например, в результате неполного усечения отростков (этот вопрос рассматривается в разделе 9.5.8). Если начать с самой левой верхней точки, будут проделаны показанные на рисунке шаги. Как видно на рис. 11.2(*в*), алгоритм вернулся в начальную точку. Если остановка произойдет в этот момент, то очевидно, что оставшаяся часть границы не будет найдена. Использование правила остановки, указанного выше в шаге 5, позволяет алгоритму продолжить работу и, как легко показать, обработать всю границу, показанную на этом рисунке.

Алгоритм прослеживания границы так же хорошо работает и в случае, когда в качестве R задано все множество точек области, а не только ее граница, как на приведенных рисунках. То есть описанная процедура находит *внешнюю границу* области на двоичном изображении. Если в задаче требуется находить границы дырок внутри области (которые называют *внутренними границами*), то простой метод состоит в выделении дырок, как описано в разделе 9.5.9, и затем работе с ними как с областями из единичных пикселей на фоне с нулевым значением, используя вышеописанный алгоритм прослеживания границ. В результате будут найдены внутренние границы исходной области.

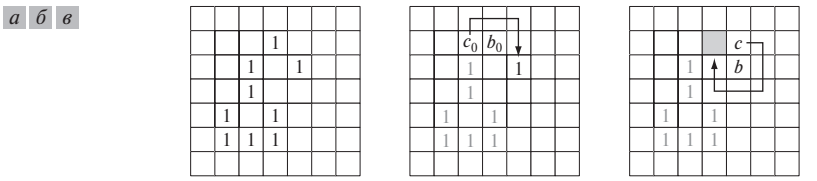


Рис. 11.2. Иллюстрация неправильной работы алгоритма, когда в качестве правила остановки используется только условие возврата в начальную точку b_0

Изложенный алгоритм несложно переформулировать для прослеживания границы в направлении против часовой стрелки, и, несомненно, читателю придется встретиться с алгоритмами, которые сформулированы в предположении, что точки границы упорядочены именно таким образом. В последующих разделах оба направления упорядочения будут использоваться попеременно (но согласованно), чтобы читатель мог освоиться с обоими подходами.

11.1.2. Цепные коды

С помощью цепных кодов граница представляется в виде последовательности соединенных отрезков, для которых указаны длина и направление. Как правило, такое представление основывается на отрезках с 4- или 8-связностью. *Направление* каждого отрезка кодируется числом в соответствии со схемой нумерации, изображенной на рис. 11.3. Границу, закодированную с помощью последовательности так занумерованных направлений, называют *цепным кодом Фримена*.

Как правило, регистрация и обработка цифровых изображений осуществляется на квадратной решетке, имеющей одинаковые шаги дискретизации в направлениях осей x и y, поэтому цепной код можно построить путем прослеживания границы, скажем, по часовой стрелке, присваивая отрезкам, соединяющим каждую пару пикселей, номера соответствующих направлений. В общем случае такой метод является неприемлемым по двум главным причинам: (1) получаемая цепочка кодов оказывается слишком длинной, и (2) любые малые возмущения вдоль границы области, вызванные наличием шума или несовершенством алгоритма сегментации, приводят к изменениям в кодовой последовательности, которые не связаны с общей формой границы.

Чтобы обойти указанные проблемы, часто применяется подход, использующий вторичную дискретизацию границы по сетке с увеличенным шагом, как показано на рис. 11.4(*а*). После этого в процессе обхода границы строятся отрезки, соединяющие узлы укрупненной сетки, выбираемые по признаку их близости к первоначальной границе, что иллюстрируется на рис. 11.4(*б*). Полученная таким способом граница с пониженным разрешением может затем представляться с помощью 4- или 8-связного цепного кода. На рис. 11.4(*в*) показаны точки огрубленной границы с представлением в виде 8-связного цепного кода, который несложно преобразовать в 4-связный (задача 2.12). В качестве начальной точки на рис. 11.4(*в*) произвольно выбран левый верхний угол границы,

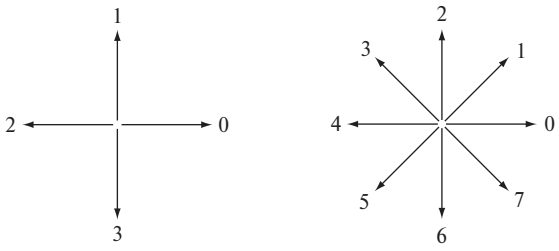


Рис. 11.3. Нумерация направлений для (*а*) 4-связного цепного кода и (*б*) 8-связного цепного кода

а б в

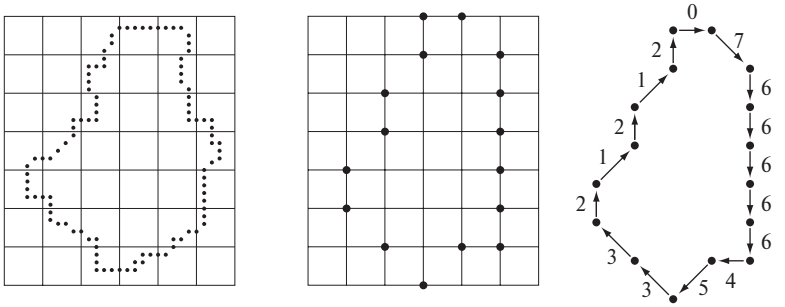


Рис. 11.4. (*а*) Дискретная граница с наложенной укрупненной сеткой дискретизации. (*б*) Результат новой дискретизации. (*в*) Представление границы 8-связным цепным кодом

что приводит к цепному коду 0766...12. Как и следовало ожидать, точность представления границы полученным кодом определяется шагом дискретизации новой сетки.

Цепной код границы области зависит от начальной точки, но с помощью простой процедуры его можно сделать инвариантным к ее выбору. Для этого цепной код просто рассматривается как циклическая последовательность номеров направлений отрезков, и начальная точка переопределяется таким образом, чтобы при начале отсчета с нее получалась линейная запись, соответствующая целому числу наименьшей величины. Цепной код также можно сделать инвариантным относительно поворота (на углы дискретных направлений согласно рис. 11.3), если вместо самого кода рассматривать его *разности первого порядка*, которые формируются как разности значений направлений пар соседних элементов кодовой последовательности. Каждая разность вычисляется циклически (против часовой стрелки и в соответствии с рис. 11.3), так что, например, для цепного кода с 4 направлениями 10103322 разность первого порядка задается последовательностью 3133030. Если и сам код трактовать как циклическую последовательность, инвариантную к выбору начальной точки, то первый элемент разностной последовательности должен вычисляться как циклическая разность последнего и первого элемента исходного цепного кода, и тогда разности первого порядка отвечает код 33133030. Нормировка по размерам области может быть достигнута путем изменения шага сетки, применяемой для новой дискретизации.

Упомянутые способы нормировки будут точными только в том случае, если границы сами по себе инвариантны относительно поворота (опять-таки на углы дискретных направлений согласно рис. 11.3) и изменения масштаба, что редко достигается на практике. Например, один и тот же объект, дискретизуемый в двух различных направлениях, вообще говоря, будет иметь границы различной формы, причем степень отличия будет тем больше, чем выше разрешение изображения. Этот эффект можно уменьшить, выбирая длину элементов цепи большей, чем единичный шаг между пикселями оцифрованного изображения, а также ориентируя сетку вторичной дискретизации вдоль главных осей описываемого объекта, как это рассматривается в разделе 11.2.2, или вдоль его собственных осей, как рассматривается в разделе 11.4.

Пример 11.1. Цепной код Фримена и некоторые его разновидности.

■ На рис. 11.5(*а*) приведено 8-битовое полутоновое изображение (размерами 570×570 пикселей) кольцевой детали, окруженной мелкими отражающими фрагментами. Цель данного примера — для внешней границы самого крупного объекта на рис. 11.5(*а*) получить цепной код Фримена в виде целого числа наименьшей величины, а также его разность первого порядка. Поскольку интересующий объект находится среди мелких фрагментов, непосредственное выделение его границы привело бы к появлению искаженной кривой, плохо описывающей общую форму объекта. При выделении границ объекта с шумом обычной процедурой является сглаживание изображения. На рис. 11.5(*б*) показано изображение после сглаживания усредняющей маской размерами 9×9, а на рис. 11.5(*в*) — результат его разделения с глобальным порогом, найденным с помощью метода Оцу. Обратите внимание, что число областей уменьшилось до двух (одна из которых представляет собой точку), что значительно упрощает задачу.

На рис. 11.5(*г*) показана внешняя граница большей из областей на рис. 11.5(*в*). Получение цепного кода непосредственно этой границы привело бы к длинной кодовой последовательности, имеющей небольшие вариации значений и не являющейся репрезентативной с точки зрения описания общей формы границы.

а *б* *в*
г *д* *е*

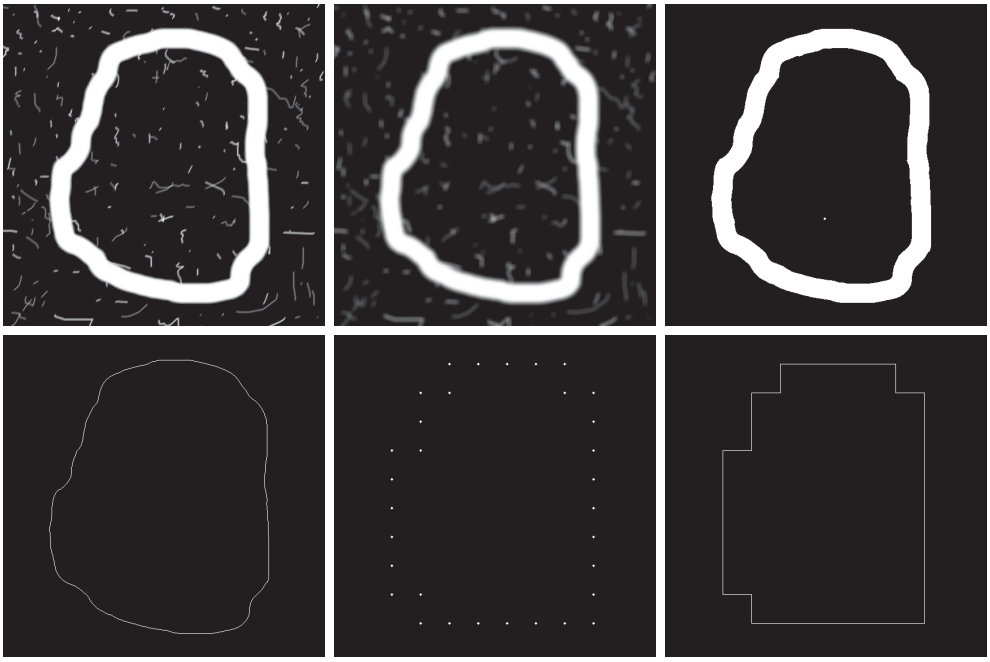


Рис. 11.5. (*а*) Зашумленное изображение. (*б*) Результат сглаживания усредняющей маской 9×9. (*в*) Результат порогового разделения сглаженного изображения методом Оцу. (*г*) Наиболее длинная внешняя граница на изображении (*в*). (*д*) Граница после вторичной дискретизации с бóльшим шагом (точки для наглядности увеличены). (*е*) Соединенные точки изображения (*д*)

Как уже отмечалось выше, перед построением цепного кода границу обычно подвергают вторичной дискретизации с увеличенным шагом, чтобы снизить изменчивость кода. На рис. 11.5(*д*) показан результат вторичной дискретизации с шагом 50 пикселей (приблизительно 10% от размера изображения), а на рис.11.5(*е*) — соединение полученных точек отрезками. Эта аппроксимация, хотя и упрощенная, сохраняет важнейшие признаки границы исходного объекта.

8-связный цепной код Фримена для этой упрощенной границы будет

0 0 0 0 6 0 6 6 6 6 6 6 6 6 4 4 4 4 4 4 2 4 2 2 2 2 2 0 2 2 0 2.

Начальная точка границы имеет координаты (2, 5) на разреженной сетке; это самая левая верхняя точка на рис. 11.5(е). Целое число с наименьшей при циклических перестановках величиной в данном случае совпадает с самим исходным кодом:

0 0 0 0 6 0 6 6 6 6 6 6 6 6 4 4 4 4 4 4 2 4 2 2 2 2 2 0 2 2 0 2.

Разность первого порядка этого цепного кода составит

0 0 0 6 2 6 0 0 0 0 0 0 0 6 0 0 0 0 0 6 2 6 0 0 0 0 6 2 0 6 2 6.

Использование любого из вышеуказанных кодов представления границы приводит к существенному уменьшению объема хранимых данных. Кроме того, работа с числовым кодом предоставляет унифицированный способ анализа формы границы, как мы увидим в разделе 11.2. Наконец, следует иметь в виду, что огрубленную границу можно восстановить по любому из приведенных кодов. ■

11.1.3. Аппроксимация ломаной линией минимальной длины

Дискретная граница может быть сколь угодно точно приближена ломаной линией. Аппроксимация становится точной, когда число отрезков ломаной равно числу точек границы (что справедливо в случае замкнутой границы) и каждую пару соседних точек соединяет свой отрезок. На практике цель аппроксимации ломаной состоит в том, чтобы с помощью как можно меньшего числа отрезков приблизить «существо» формы границы. В общем случае эта задача нетривиальна и ее решение часто выливается в трудоемкие переборные схемы. Тем не менее некоторые методы аппроксимации, которые характеризуются умеренной вычислительной сложностью, хорошо подходят для цифровой обработки изображений. Среди таких методов один из самых мощных — это представление границы *ломаной линией минимальной длины* (ЛМД), к рассмотрению которого мы сейчас перейдем.

Основной принцип

Интуитивно привлекательным подходом для построения алгоритма нахождения ЛМД представляется следующий. Разобьем изображение на квадратные элементы и выберем множество соединенных в цепочку элементов, заключающих в себе границу области (рис. 11.6(*а*)), как это показано на рис. 11.6(*б*). Такой подход позволяет рассматривать границу области как резиновую ленту, находящуюся между двумя стенками, соответствующими внутренней и внешней гра-

ницам указанной цепочки элементов. При стягивании ленты она примет форму, указанную на рис. 11.3(*в*), образуя многоугольник с минимальным периметром, отвечающий геометрической форме данной цепочки элементов. Заметим, что на этом рисунке все вершины ЛМД совпадают с углами либо внутренней, либо наружной границы сформированной цепочки элементов.

Точность аппроксимации ломаной линией определяется размерами элементов, на которые разбита область, содержащая границу исходного объекта. В пределе, когда каждый элемент разбиения соответствует одному пикселю границы, величина отклонения фактической границы от ее ЛМД-приближения внутри любого элемента не превышает $\sqrt{2}d$, где d — минимальное расстояние между пикселями, т. е. шаг дискретизации при получении исходного цифрового изображения. Это отклонение можно считать вдвое меньшим, если принимать за опорную точку элемента центр соответствующего ему пикселя. Задача состоит в использовании элементов разбиения как можно большего размера, чтобы полученная ЛМД содержала минимальное число вершин и обеспечивала при этом достаточную для конкретного приложения точность аппроксимации. В данном разделе мы сформулируем процедуру нахождения вершин такой ломаной линии.

Согласно изложенному подходу, основанному на клеточном разбиении, форма объекта, заключенного внутри исходной границы, приводится к области, которая ограничена сформированной цепочкой элементов, аналогично показанной темным цветом на рис. 11.6(*б*). На рис. 11.7(*а*) исходный объект отмечен темно-серым цветом. Как видим, его граница состоит из 4-связных отрезков прямой. Будем обходить границу объекта в направлении против часовой стрелки. Каждый встретившийся поворот будет либо выпуклой, либо вогнутой вершиной в зависимости от величины *внутреннего* угла 4-связной границы в этой точке. На рис. 11.7(*б*) выпуклые и вогнутые вершины отмечены соответственно белыми и черными точками. Заметим, что все эти вершины принадлежат вну-

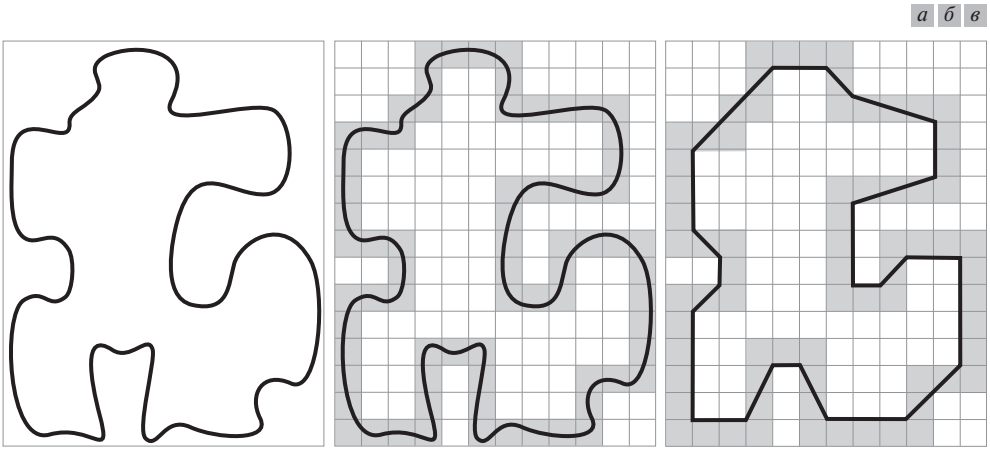


Рис. 11.6. (*а*) Граница объекта (черная кривая). (*б*) Граница объекта, заключенная внутри цепочки элементов (выделены темным цветом). (*в*) Ломаная минимальной длины, полученная в результате стягивания границы. Вершины ломаной совпадают с углами внутренней или внешней границ выделенной области

а б в

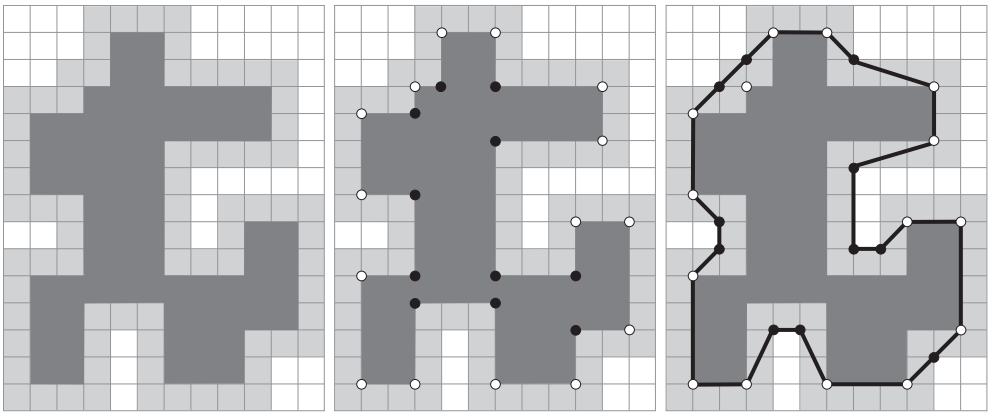


Рис. 11.7. (*а*) Темно-серый объект, граница которого заключена в сформированную цепочку элементов (см. рис. 11.6). (*б*) Выпуклые (белые) и вогнутые (черные) вершины, найденные при обходе границы темно-серого объекта против часовой стрелки. (*в*) Вогнутые вершины (черные точки), перенесенные в диагонально противоположные точки на внешней границе сформированной области, выделенной светло-серым цветом; выпуклые вершины на прежних местах. Черной линией для наглядности показана ЛМД

тренней границе сформированной области, отмеченной светло-серым цветом на рис. 11.7(*б*), и что для каждой вогнутой вершины границы объекта (черной точки) имеется «зеркальная» точка внешней границы указанной области, расположенная в диагонально противоположном углу соответствующего элемента разбиения. Зеркальные точки всех вогнутых вершин показаны на рис. 11.7(*в*), куда для наглядности аналогично рис. 11.6(*в*) наложена ЛМД. Как видим, вершины ЛМД совпадают либо с выпуклыми вершинами внутренней границы сформированной светло-серой области (белые точки), либо для вогнутых вершин с их зеркальными точками на ее внешней границе (черные точки). Нетрудно догадаться, что в качестве вершин ЛМД могут выступать только выпуклые вершины внутренней границы и вогнутые вершины внешней границы окружающей области. Таким образом, наш алгоритм должен сосредоточиться только на таких вершинах.

> Выпуклой вершиной является средняя из трех точек, если они образуют угол в интервале $0° < \theta < 180°$; аналогично угол при вогнутой вершине находится в интервале $180° < \theta < 360°$. Угол 180° означает вырожденную вершину (находящуюся на прямой, соединяющей две соседние точки), которая не может быть вершиной ЛМД, а углы 0° или 360° означают поворот пути в обратном направлении, что невозможно в данном рассмотрении.

Алгоритм построения ломаной линии минимальной длины

Описанное в предыдущем параграфе сформированное множество элементов разбиения, содержащее дискретную границу (объекта), называется *клеточным*

комплексом. В силу выбранного способа построения мы рассматриваем только границы без самопересечений и только *связные* клеточные комплексы. Исходя из такого предположения и присваивая *выпуклым* вершинам белый цвет (*W*), а *зеркальным к вогнутым* вершинам — черный (*B*), можно сформулировать следующие утверждения.

1. ЛМД, ограниченная связным клеточным комплексом, является несамопересекающейся.
2. Каждая *выпуклая* вершина ЛМД является белой, но не каждая белая вершина границы является вершиной ЛМД.
3. Каждая *вогнутая* вершина ЛМД является черной, но не каждая черная вершина границы является вершиной ЛМД.
4. Все черные вершины расположены на ЛМД или снаружи от нее, а все белые вершины расположены на ЛМД или внутри нее.
5. Левой верхней вершиной в последовательности вершин, заключенных внутри клеточного комплекса, всегда является белая вершина ЛМД.

Эти утверждения можно строго доказать ([Sklansky et al., 1972], [Sloboda et al., 1998] и [Klette, Rosenfeld, 2004]), но для наших целей их справедливость очевидна (см. рис. 11.7), поэтому мы здесь не будем останавливаться на доказательствах. В отличие от углов в вершинах темно-серой области на рис. 11.7, углы, образованные в вершинах ЛМД, не обязательно кратны 90°.

В последующем рассмотрении нам понадобится вычислять ориентацию троек точек. Рассмотрим такую тройку точек (a, b, c) с координатами $a = (x_1, y_1)$, $b = (x_2, y_2)$ и $c = (x_3, y_3)$. Если расположить эти точки в строках матрицы

$$\mathbf{A} = \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix}, \tag{11.1-1}$$

то, как следует из элементарного матричного анализа,

$$\det(\mathbf{A}) = \begin{cases} > 0 & \text{если последовательность } (a,b,c) \text{ обходится против часовой стрелки,} \\ = 0 & \text{если точки } (a,b,c) \text{ расположены на одной прямой,} \\ < 0 & \text{если последовательность } (a,b,c) \text{ обходится по часовой стрелке,} \end{cases} \tag{11.1-2}$$

где det(**A**) — определитель матрицы **A**. В этой формуле направление по или против часовой стрелки понимается в терминах правосторонней системы координат (см. примечание в разделе 2.4.2). Например, при использовании системы координат изображения (рис. 2.18), в которой начало координат расположено в левом верхнем углу, ось x направлена вниз, а ось y — вправо, последовательность точек $a = (3, 4)$, $b = (2, 3)$ и $c = (3, 2)$ обходится против часовой стрелки, и если подставить эти координаты в формулу (11.1-2), получим положительное значение определителя. Для удобства обозначений при описании алгоритма, определим здесь функцию sgn(·), совпадающую со значением определителя:

> В предположении координатной системы согласно рис. 2.18(*б*) при обходе границы многоугольника против часовой стрелки все точки, находящиеся справа от направления обхода, расположены снаружи этого многоугольника, а находящиеся слева — внутри многоугольника.

$$\operatorname{sgn}(a,b,c) \equiv \det(\mathbf{A}). \qquad (11.1\text{-}3)$$

Таким образом, для последовательности точек, которая обходится против часовой стрелки, будет $\operatorname{sgn}(a, b, c) > 0$, по часовой стрелке — $\operatorname{sgn}(a, b, c) < 0$, а для точек на одной прямой $\operatorname{sgn}(a, b, c) = 0$. С геометрической точки зрения условие $\operatorname{sgn}(a, b, c) > 0$ означает, что точка c лежит с положительной стороны от прямой[2], проходящей через точки a и b (прямая (a, b)); если же $\operatorname{sgn}(a, b, c) < 0$, точка c находится с отрицательной стороны от этой прямой. Формулы (11.1-2) и (11.1-3) дают тот же самый результат для последовательностей (c, a, b) или (b, c, a), поскольку направление обхода такое же, как для последовательности (a, b, c), однако геометрическая интерпретация условия становится другой. Например, $\operatorname{sgn}(c, a, b) > 0$ означает, что точка b лежит с положительной стороны от прямой через точки c и a (прямой (c, a)).

Подготовка данных для алгоритма ЛМД начинается с формирования массива, строки которого содержат координаты всех вершин и дополнительную пометку цвета вершины (т. е. является она черной или белой). Важно, чтобы: (1) вогнутые вершины представлялись своими зеркальными отражениями, как на рис. 11.7(*в*); (2) вершины были упорядочены[3] и (3) первой стояла левая верхняя вершина, которая, как мы знаем из свойства 5, является белой вершиной ЛМД. Обозначим эту вершину V_0 и предположим, что вершины далее упорядочены в направлении против часовой стрелки. В алгоритме построения ЛМД используются две скользящие точки — белый указатель (W_c) и черный указатель[4] (B_c), первый из которых скользит по выпуклым (белым) вершинам, а второй — по зеркальным парам вогнутых вершин (черным). Для описываемой процедуры необходимы только эти два указателя, текущая рассматриваемая вершина и предыдущая (последняя найденная) вершина ЛМД.

В начале алгоритма значения указателей устанавливаются равными $W_c = B_c = V_0$ (напомним, что V_0 является вершиной ЛМД). На любом из последующих шагов алгоритма V_L обозначает последнюю найденную вершину ЛМД, а V_k — текущую рассматриваемую вершину. Для этих вершин и упомянутых указателей может выполняться одно из трех условий:

1. $\operatorname{sgn}(V_L, W_c, V_k) > 0$, т. е. V_k находится с положительной стороны от прямой (V_L, W_c).
2. $\operatorname{sgn}(V_L, W_c, V_k) \le 0$ и $\operatorname{sgn}(V_L, B_c, V_k) \ge 0$, т. е. V_k находится с отрицательной стороны от прямой (V_L, W_c) или на самой этой прямой, а V_k находится с положительной стороны от прямой (V_L, B_c) или на самой этой прямой; каждое из данных двух условий выполняется также в случаях, когда соответственно, W_c или B_c совпадают с V_L.
3. $\operatorname{sgn}(V_L, B_c, V_k) < 0$, т. е. V_k находится с отрицательной стороны от прямой (V_L, B_c).

---

[2] В данном разделе предполагается, что прямая, проведенная через пару точек a и b, имеет направление, определяемое очередностью выбора точек. Так, прямая (b, a) имеет противоположное направление по отношению к прямой (a, b). Это позволяет говорить о «положительной» и «отрицательной» сторонах прямой. — *Прим. перев.*

[3] Упорядочивание вершин границы области осуществляется с помощью алгоритма прослеживания границы, например описанного в разделе 11.1.1.

[4] В оригинале *white crawler* и *black crawler*. — *Прим. перев.*

Если выполнено условие (а), следующей вершиной ЛМД становится W_c. Указатели устанавливаются в одно значение $V_L = B_c = W_c$, и работа продолжается со следующей за V_L вершиной.

Если выполнено условие (б), V_k становится *кандидатом* на роль вершины ЛМД. В этом случае, если V_k — выпуклая вершина (т. е. белая), устанавливается $W_c = V_k$; в противоположном случае (V_k — черная) устанавливается $B_c = V_k$. После этого осуществляется переход к следующей по порядку вершине.

Если выполнено условие (в), следующей вершиной ЛМД становится B_c. Указатели устанавливаются в одно значение $V_L = W_c = B_c$, и работа продолжается со следующей за V_L вершиной.

Алгоритм останавливается, когда снова достигнута начальная вершина, тем самым обработаны все вершины ломаной. Найденные алгоритмом вершины V_L являются вершинами ЛМД. Доказано, что этот алгоритм находит все вершины кратчайшей ломаной, заключенной внутри односвязного клеточного комплекса ([Sloboda et al., 1998] и [Klette, Rosenfeld, 2004]).

Пример 11.2. Иллюстрация алгоритма нахождения ЛМД.

■ Выполненный вручную пример поможет уяснить изложенные идеи. Рассмотрим вершины на рис. 11.7(*в*). В принятой системе координат изображения левая верхняя точка сетки имеет координаты (0, 0). В предположении, что сетка проведена с единичным шагом, несколько начальных строк массива вершин (в направлении против часовой стрелки) будут иметь следующий вид:

$$\begin{array}{lll} V_0 & (1, 4) & W \\ V_1 & (2, 3) & B \\ V_2 & (3, 3) & W \\ V_3 & (3, 2) & B \\ V_4 & (4, 1) & W \\ V_5 & (7, 1) & W \\ V_6 & (8, 2) & B \\ V_7 & (9, 2) & B. \end{array}$$

Первой в списке всегда стоит вершина ЛМД, поэтому вначале $W_c = B_c = V_L = V_0 = (1, 4)$. Следующей является вершина $V_1 = (2, 3)$. Вычисление функции sgn дает $\text{sgn}(V_L, W_c, V_1) = 0$ и $\text{sgn}(V_L, B_c, V_1) = 0$, так что выполнено условие (б). Устанавливаем $B_c = V_1 = (2, 3)$, т. к. вершина V_1 — вогнутая (черная). Значение указателя W_c не меняется и по-прежнему равно (1, 4), как и значение V_L, поскольку новая вершина ЛМД не была найдена.

Далее рассматриваем вершину $V_2 = (3, 3)$, для которой значения функции sgn следующие: $\text{sgn}(V_L, W_c, V_2) = 0$ и $\text{sgn}(V_L, B_c, V_2) = 1$, так что снова выполнено условие (б) алгоритма. Поскольку V_2 — выпуклая (белая), устанавливаем $W_c = V_2 = (3, 3)$. К этому моменту значения указателей равны $W_c = (3, 3)$ и $B_c = (2, 3)$, а значение V_L не изменилось.

Следующей является вершина $V_3 = (3, 2)$. Для нее $\text{sgn}(V_L, W_c, V_3) = -2$ и $\text{sgn}(V_L, B_c, V_3) = 0$, так что снова выполнено условие (б). Поскольку это черная вершина, модифицируем значение черного указателя $B_c = (3, 2)$. Указатель W_c не меняется, как и V_L.

Следующей является вершина $V_4 = (4, 1)$, и имеем $\text{sgn}(V_L, W_c, V_4) = -3$ и $\text{sgn}(V_L, B_c, V_4) = 0$, так что снова выполнено условие (б). Поскольку это белая вершина, модифицируем значение белого указателя $W_c = (4, 1)$. Указатель B_c по-прежнему указывает на (3, 2), а V_L — на (1, 4).

Следующей идет вершина $V_5 = (7, 1)$, и для нее $\text{sgn}(V_L, W_c, V_5) = 9$, поэтому выполнено условие (а) алгоритма, и мы устанавливаем $V_L = W_c = (4, 1)$. Поскольку найдена новая вершина ЛМД, заново инициализируем указатели $W_c = B_c = V_L$. Работа должна продолжаться с новой вершиной, которая идет после вновь найденной V_L, т. е. мы опять попадаем в вершину V_5.

Для $V_5 = (7, 1)$ с новыми значениями V_L, W_c и B_c имеем $\text{sgn}(V_L, W_c, V_5) = 0$ и $\text{sgn}(V_L, B_c, V_5) = 0$, так что опять выполняется условие (б). Следовательно, устанавливаем $W_c = V_5 = (7, 1)$, поскольку V_5 является белой вершиной.

Следующей идет вершина $V_6 = (8, 2)$ и $\text{sgn}(V_L, W_c, V_6) = 3$, так что выполнено условие (а). Устанавливаем $V_L = W_c = (7, 1)$ и заново инициализируем указатели $W_c = B_c = V_L$. Из-за этого следующей рассматриваемой вершиной снова будет $V_6 = (8, 2)$. Продолжая действия описанным выше образом с оставшимися вершинами, получим ЛМД, приведенную на рис. 11.7(*в*). Как уже отмечалось, черные вершины (2, 3), (3, 2) и (13, 10) в правом нижнем углу рисунка хотя и являются точками ЛМД, но расположены на одной прямой с соседними вершинами и поэтому *не являются* вершинами ломаной минимальной длины. Соответственно, алгоритм их и не обнаруживает. ■

Пример 11.3. Применение алгоритма нахождения ЛМД.

■ На рис. 11.8(*а*) представлено двухградационное изображение кленового листа (размерами 566×566 пикселей), а на рис. 11.8(*б*) — его 8-связная граница. Последовательность рис. 11.8(*в*)—(*и*) демонстрирует аппроксимацию этой границы с помощью ломаной минимальной длины, полученной при использовании клеточных комплексов с квадратными элементами размерами 2, 3, 4, 6, 8, 16 и 32 пикселя соответственно (на каждом из рисунков найденные вершины ЛМД для наглядности соединены отрезками, чтобы образовалась замкнутая граница). Главными характерными деталями листа являются черенок и три выступающих доли. Как показывает рис. 11.8(*е*), черенок теряется, начиная с размеров элемента больше, чем 4×4. Основные три выступа листа сохраняются достаточно хорошо, даже при размерах элемента 16×16 (см. рис. 11.8(*з*)). Однако, как показывает рис. 11.8(*и*), при увеличении шага сетки до 32 эти характерные признаки почти утрачиваются.

Исходная граница на рис. 11.8(*б*) состоит из 1900 точек. Число вершин ЛМД на рис. 11.8(*в*)—(*и*) составляет соответственно 206, 160, 127, 92, 66, 32 и 13. Рис. 11.8(*д*), в котором ломаная имеет 127 вершин, сохраняет все основные признаки первоначальной границы и при этом обеспечивает сокращение объема данных свыше 90%, тем самым демонстрируя значительные преимущества представления границы с помощью ЛМД. Другим важным достоинством является то, что ЛМД сглаживает границу. Как мы видели в предыдущем разделе, это является обычным требованием при представлении границы цепным кодом. ■

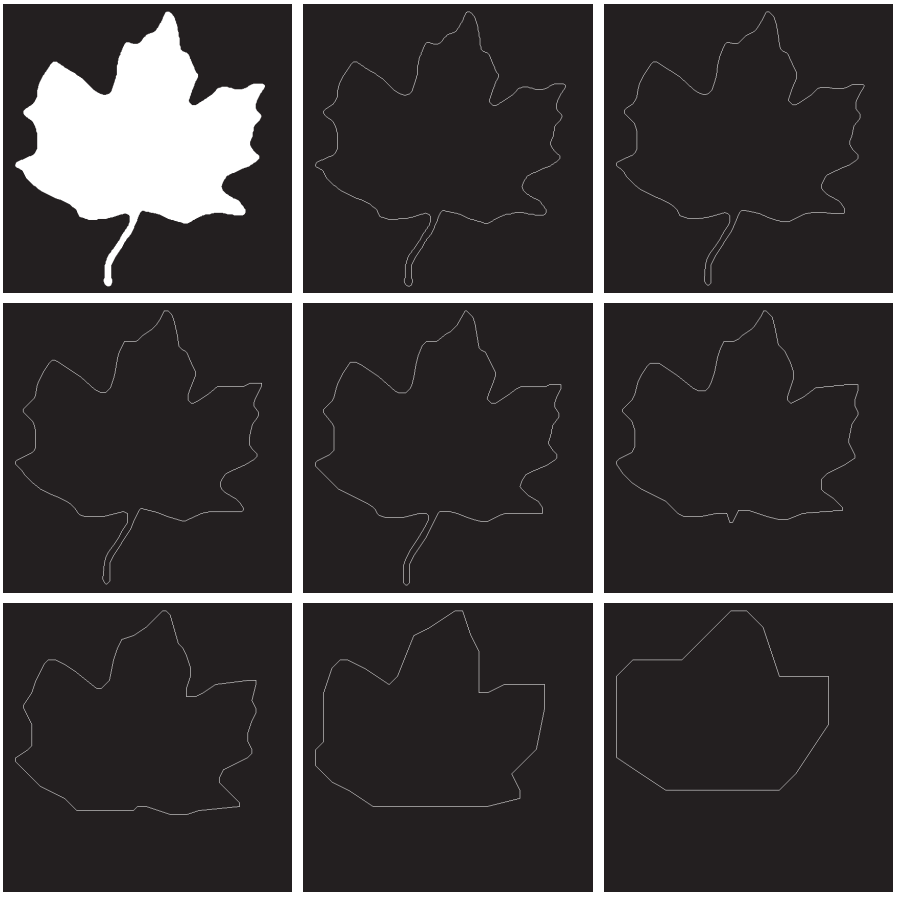


Рис. 11.8. (*а*) Двоичное изображение размерами 566×566. (*б*) 8-связная граница области. (*в*)—(*и*) ЛМД, полученные при использовании квадратных элементов размерами 2, 3, 4, 6, 8, 16 и 32 пикселей соответственно (вершины для наглядности соединены отрезками). Число точек границы на (*б*) равно 1900; число вершин на (*в*)—(*и*) составляет соответственно 206, 160, 127, 92, 66, 32 и 13

11.1.4. Другие методы аппроксимации ломаной линией

Иногда для аппроксимации ломаной линией могут применяться другие методы, более простые, чем изложенный в предыдущем параграфе алгоритм ЛМД. Два таких метода рассматриваются в данном разделе.

Методы слияния

Методы *слияния* основаны на применении к задаче кусочно-линейной аппроксимации критерия средней ошибки (или критерия другого вида). Согласно одному из подходов происходит объединение точек вдоль границы в одну прямую линию до тех пор, пока среднеквадратическое отклонение объединяемых точек от формируемой прямой не превысит некоторый заранее заданный порог. После этого параметры прямой запоминаются и начинается новое объединение

точек, сопровождающееся построением новой прямой и новым накоплением ошибки отклонения. В результате повторения такой процедуры в конце будет построена ломаная, состоящая из соседних отрезков аппроксимирующих прямых. Одним из главных недостатков описанного метода является то, что вершины полученной ломаной линии не всегда совпадают с изгибами изначальной границы (например с ее изломами), так как новое звено ломаной не начинается до тех пор, пока величина отклонения предыдущего отрезка ломаной не превысит заданный порог. Например, если при прослеживании длинного прямолинейного отрезка границы встретится угол, то к этому отрезку будет добавлено еще некоторое (зависящее от установленного порога) количество точек за вершиной угла, прежде чем обнаружится превышение порога. Однако этот недостаток можно уменьшить, применяя наряду со слиянием обсуждаемые ниже методы разбиения.

Методы разбиения

Один из возможных подходов к *разбиению* отрезков границы состоит в том, что отрезок последовательно разбивается на две части (т.е. заменяется двумя новыми отрезками) до тех пор, пока не начнет выполняться некоторый заданный критерий. Например, может быть поставлено такое требование, чтобы максимум кратчайших расстояний от отрезка прямой, соединяющей две точки границы, до промежуточных точек границы не превышал установленного порога. Если это условие нарушается, то максимально удаленная от отрезка точка границы становится новой вершиной аппроксимирующей ломаной, а первоначальный отрезок ломаной заменяется двумя новыми. Этот метод обладает тем достоинством, что позволяет обнаруживать наиболее заметные точки изгиба или излома на границе объекта. В случае замкнутой границы лучшими начальными точками для такого алгоритма обычно являются две точки границы, находящиеся на максимальном расстоянии друг от друга. В качестве примера на рис. 11.9(*а*) изображена граница объекта, а рис. 11.9(*б*) демонстрирует этап дробления начального отрезка аппроксимации этой границы, соединяющего ее наиболее удаленные друг от друга точки. Точка *c* является наиболее удаленной (в смысле кратчайшего расстояния) от отрезка *ab* точкой верхнего участка

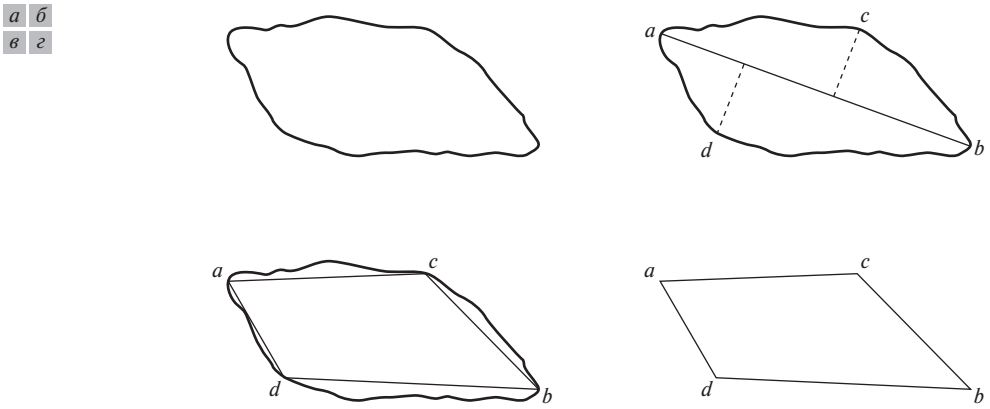


Рис. 11.9. (*а*) Исходная граница. (*б*) Разбиение границы на участки с помощью угловых точек. (*в*) Добавление вершин. (*г*) Полученная ломаная

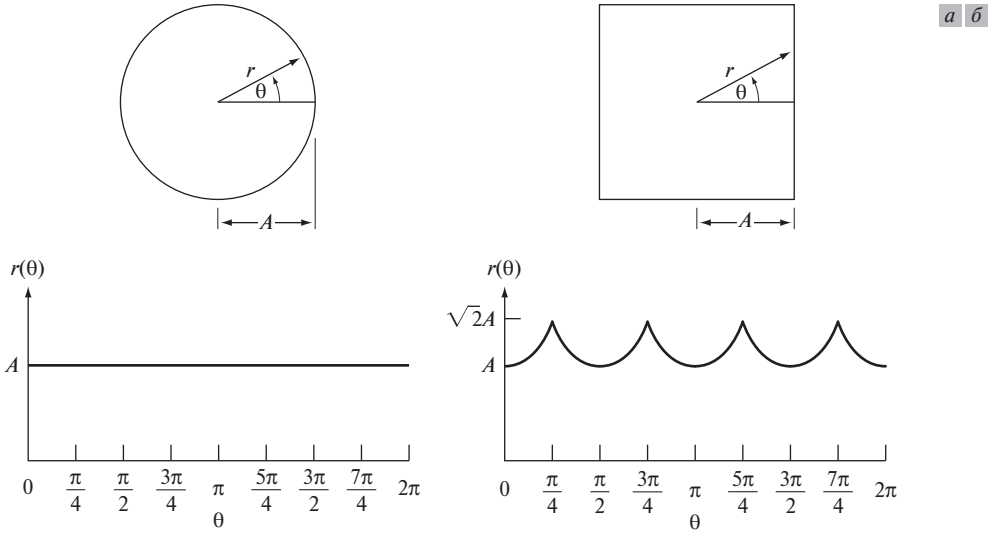


Рис. 11.10. Сигнатуры «угол—расстояние». В случае (*а*) $r(\theta) = \text{const}$. В случае (*б*) сигнатура состоит из повторяющихся с периодом $\pi/2$ участков зависимостей $r(\theta) = A/\cos\theta$ для $0 \le \theta \le \pi/4$ и $r(\theta) = A/\sin\theta$ для $\pi/4 < \theta \le \pi/2$

границы, а точка d — наиболее удаленной точкой нижнего участка границы. На рис. 11.9(*в*) показан результат применения описанной выше процедуры разбиения с порогом, равным одной четверти длины отрезка ab. Поскольку ни для одного из полученных отрезков этот порог (максимум кратчайших расстояний от точек границы до соответствующих отрезков) не превышен, процедура завершается построением замкнутой ломаной линии, показанной на рис. 11.9(*г*).

11.1.5. Сигнатуры

Сигнатура есть описание границы объекта с помощью одномерной функции, которое может строиться различными способами. Один из простейших состоит в построении зависимости расстояния от центроида (т. е. от некоторой средней точки объекта, например его центра тяжести) до границы объекта в виде функции угла, как иллюстрирует рис. 11.10. Независимо от способа построения сигнатуры, основная идея состоит в том, чтобы свести представление границы к одномерной функции, которую предположительно описать легче, чем исходную двумерную границу.

Сигнатуры, построенные описанным выше способом, инвариантны по отношению к параллельному переносу, однако зависят от поворота и изменения масштаба. Инвариантность к повороту можно получить, найдя способ выбора одной и той же начальной точки для построения сигнатуры, независимо от ориентации фигуры. Один из способов сделать это — выбирать в качестве начальной точку, максимально удаленную от центроида (в предположении, что такая точка оказывается единственной для интересующих фигур). Другой способ может заключаться в выборе максимально удаленной от центроида точки на собственной оси фигуры (см. раздел 11.4). Такой метод требует большего объема вычисле-

ний, но является и более устойчивым, поскольку направление собственной оси фигуры определяется с учетом всех точек ее контура. Еще один способ основан на получении цепного кода границы и последующем применении метода, описанного в разделе 11.1.2, полагая, что кодирование является достаточно грубым, так что поворот не нарушает его цикличности.

Если предполагать, что изменение масштаба производится одинаково по обеим осям и дискретизация по углу θ является равномерной, то масштабирование фигуры приводит к изменению амплитуды соответствующей сигнатуры. Этот результат можно пронормировать путем масштабирования функций сигнатуры таким образом, чтобы они всегда охватывали один и тот же диапазон значений, например [0, 1]. Достоинством данного метода является простота, однако его потенциальный серьезный недостаток кроется в том, что масштабирование всей функции зависит всего от двух значений — минимального и максимального. Если в изображении присутствует шум, эта зависимость может стать источником значительных расхождений между объектами. Более устойчивый (но также требующий бóльших вычислений) метод состоит в делении каждого углового отсчета на дисперсию функции, задающей сигнатуру, в предположении что эта дисперсия не нулевая (как на рис. 11.10(*а*)) и не столь мала, чтобы это вызывало вычислительные трудности. Учет дисперсии приводит к переменному коэффициенту масштабирования, который обратно пропорционален изменениям размеров и действует аналогично автоматической регулировке усиления. Какой бы способ ни применялся, следует иметь в виду, что основная цель состоит в устранении зависимости от масштаба при сохранении формы кривой в целом.

Зависимость расстояния от угла — не единственный способ построения сигнатуры. Например, другой способ состоит в обходе точек границы с одновременным построением зависимости угла между касательной к границе в текущей точке и фиксированной опорной прямой от положения точки на границе. Сигнатура, получаемая таким способом, имеет совершенно другой вид по сравнению с кривой $r(\theta)$ на рис. 11.10, однако также несет информацию об основных характеристиках формы объекта. Например, горизонтальные участки такой функции соответствуют прямолинейным отрезкам границы области, поскольку вдоль них угол наклона касательной постоянен. Вариантом данного метода является использование в качестве сигнатуры так называемой *функции плотности вероятности наклона*. Эта зависимость представляет собой обычную гистограмму значений угла наклона касательной. Поскольку гистограмма является мерой концентрации значений, данная функция дает большой отклик на участках границы с постоянными углами наклона касательной (т. е. прямолинейных или близких к таковым) и характеризуется глубокими спадами на участках быстрого изменения угла наклона (т. е. в углах и местах резких изгибов).

Пример 11.4. Сигнатуры двух объектов простой формы.

■ На рис. 11.11(*а*) и (*б*) показаны два двоичных объекта, а на рис. 11.11(*в*) и (*г*) — их границы. Соответствующие сигнатуры $r(\theta)$ на рис. 11.11(*д*) и (*е*) получены для углов θ в интервале от 0° до 360° с шагом 1°. Числа выраженных пиков на сигнатуре достаточно для различения формы двух объектов. ■

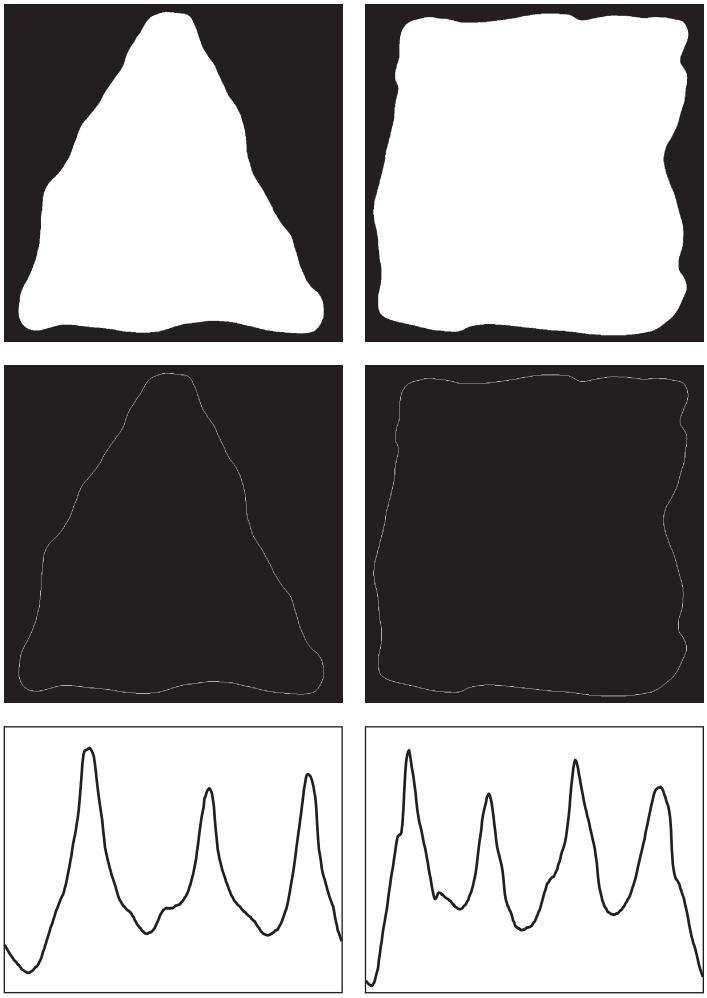


Рис. 11.11. Две двоичные области, их внешние границы и соответствующие им сигнатуры $r(\theta)$. Горизонтальные оси на (*д*) и (*е*) соответствуют углам от 0° до 360° с шагом 1°

11.1.6. Сегменты границы

Часто оказывается полезным разбиение границы на сегменты. При такой декомпозиции уменьшается сложность границы и тем самым упрощается процесс ее описания. Такой подход особенно привлекателен, когда граница содержит одну или несколько хорошо выраженных вогнутостей, несущих информацию о форме объекта. В этом случае мощным инструментом для устойчивой декомпозиции границы является использование выпуклой оболочки области, находящейся внутри границы.

Согласно определению из раздела 9.5.4, *выпуклая оболочка H* произвольного множества *S* есть наименьшее выпуклое множество, содержащее *S*. Разность множеств $H \setminus S$ называется *дефицитом выпуклости D* множества *S*. Чтобы проил-

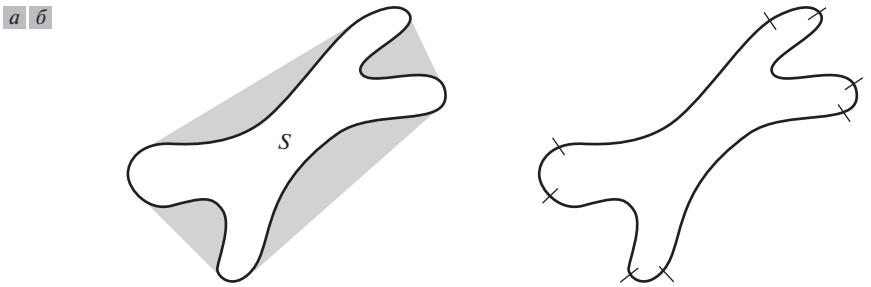


Рис. 11.12. (*а*) Область *S* и ее дефицит выпуклости (отмечен темным цветом). (*б*) Разбиение границы области

люстрировать, как эти понятия могут использоваться для разбиения границы на содержательные сегменты, рассмотрим рис. 11.12(*а*), на котором изображен объект (множество *S*) со своим дефицитом выпуклости (области, закрашенные темным цветом). Разбиение границы области осуществляется путем обхода контура *S* и пометки точек входа в область дефицита выпуклости и выхода из нее. На рис. 11.12(*б*) показано местоположение этих точек в рассматриваемом случае. Отметим, что результат такого разбиения в принципе не зависит от изменения размеров и ориентации области.

На практике границы областей на дискретных изображениях часто являются неровными из-за шума, погрешностей дискретизации и отклонений при сегментации. В результате область дефицита выпуклости содержит незначащие мелкие составляющие, случайно разбросанные вдоль границы. Вместо того чтобы пытаться отсеять эти неровности в ходе последующей обработки, общепринятый способ состоит в сглаживании границы перед ее разбиением. Для этого существует целый ряд способов. При одном из них во время обхода границы координаты каждого пикселя заменяются средними значениями координат *k* его соседей по границе. Этот способ справляется с небольшими неровностями, однако требует большого объема вычислений и трудно поддается контролю. При больших значениях *k* происходит излишнее сглаживание, а малые значения *k* могут оказаться недостаточными на некоторых участках границы. Более надежный способ состоит в том, что перед нахождением дефицита выпуклости выполняется кусочно-линейная аппроксимация границы. Как правило, границы интересующих объектов на дискретных изображениях аппроксимируются замкнутыми ломаными без самопересечений (см. раздел 11.1.3), т. е. области являются простыми многоугольниками. Грэм и Яо [Graham, Yao, 1983] предложили алгоритм построения выпуклой оболочки таких многоугольников.

Понятия выпуклой оболочки и дефицита выпуклости оказываются столь же полезными и для описания всей области, а не только ее границы. Например, в основу описания области может быть положена ее площадь, площадь дефицита выпуклости, число компонент дефицита выпуклости, их относительное расположение и т. д. Напомним, что морфологический алгоритм для построения выпуклой оболочки был описан в разделе 9.5.4. В списке литературы для дальнейшего изучения в конце данной главы содержатся ссылки на другие способы построения.

11.1.7. Остовы областей

Важным для практики является подход, в котором представление формы плоской области строится путем сведения ее к графу. Такое сокращенное представление можно получить, выделяя *остов* этой области с помощью алгоритма утончения (этот процесс иначе называют *скелетонизацией*). Процедуры утончения занимают центральное место в широком классе прикладных задач обработки изображений, от автоматического контроля печатных плат до подсчета волокон асбеста в воздушных фильтрах. В разделе 9.5.7 мы уже рассматривали построение остова с использованием морфологического подхода. Однако, как отмечалось, в описанной там процедуре не было предусмотрено никаких условий сохранения связности остова. Ниже излагается алгоритм, позволяющий исправить этот недостаток.

Остов области может быть построен с помощью преобразования к главным осям (ПГО), предложенного Блюмом [Blum, 1967]. ПГО области R с границей B выполняется следующим образом. Для каждой точки p области R находим ближайшую к ней точку B. Если таких точек больше одной, то говорится, что точка p лежит на *срединной оси* области R (т. е. остове). Понятие «ближайшей» точки (и соответствующее ему ПГО) зависит от определения расстояния (см. раздел 2.5.3). На рис. 11.13 приводится несколько примеров, в которых используется евклидово расстояние. Те же результаты были бы получены при использовании максимальных кругов из раздела 9.5.7.

ПГО области имеет наглядное объяснение, образно называемое «пожар в степи». Рассмотрим область изображения как степь, равномерно покрытую сухой травой, и предположим, что вся ее граница одновременно загорается. Фронт пожара распространяется внутрь области, всюду с постоянной скоростью. Результатом ПГО области будет множество точек, куда фронт огня доходит одновременно более чем с одного направления.

Хотя ПГО области приводит к получению интуитивно приемлемого остова, непосредственная реализация такого алгоритма требует большого объема вычислений, поскольку потенциально связана с вычислением расстояний от каждой внутренней точки области до всех точек границы. Предложено большое число алгоритмов, позволяющих повысить эффективность вычислений при построении результата ПГО области. Как правило, в них используются алгоритмы утончения, в которых постепенно убираются точки гра-

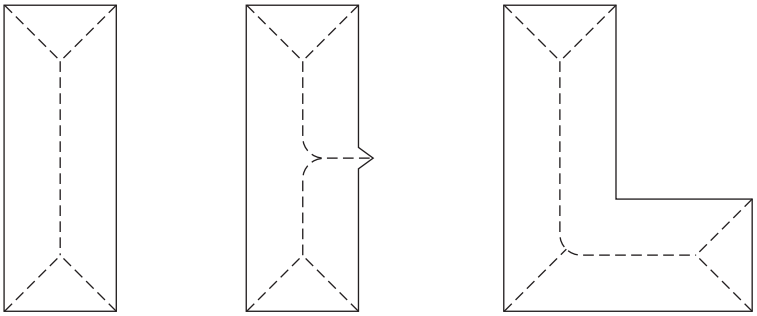


Рис. 11.13. Срединные оси трех областей простой формы (показаны пунктиром)

ницы области при тех условиях, что (1) они не являются концевыми точками, (2) после удаления область остается связной и (3) это не приводит к излишней эрозии области.

В этом разделе мы изложим алгоритм утончения двоичных областей, предполагая, что точки области имеют значения 1, а точки фона — 0. Метод состоит в последовательном применении двух основных шагов, которые применяются для граничных точек данной области. В соответствии с определением, данным в разделе 2.5.2, *граничной точкой* является любой единичный пиксель, среди соседей которого есть хотя бы один элемент с нулевым значением. С использованием приведенных на рис. 11.14 обозначений для восьмерки соседей на первом шаге алгоритма контурная точка p_1 помечается для удаления, если выполнены следующие условия:

(а) $2 \le N(p_1) \le 6$,
(б) $T(p_1) = 1$,
(в) $p_2 \cdot p_4 \cdot p_6 = 0$,
(г) $p_4 \cdot p_6 \cdot p_8 = 0$, (11.1-4)

где $N(p_1)$ — число ненулевых соседей элемента p_1, т. е.

$$N(p_1) = p_2 + p_3 + \ldots + p_8 + p_9, \quad (11.1\text{-}5)$$

где каждый из p_i есть либо 0, либо 1, а $T(p_1)$ — число переходов 0—1 в упорядоченной последовательности $p_2, p_3, \ldots, p_8, p_9, p_2$. Например, на рис. 11.15 $N(p_1) = 4$ и $T(p_1) = 3$.

На втором шаге условия (а) и (б) остаются теми же, а условия (в) и (г) заменяются на

(д) $p_2 \cdot p_4 \cdot p_8 = 0$,
(е) $p_2 \cdot p_6 \cdot p_8 = 0$. (11.1-6)

Вначале к каждой точке границы рассматриваемой двоичной области применяется шаг 1. Если нарушается хотя бы одно из условий (а)—(г), то значение соответствующего элемента не меняется. При выполнении всех условий элемент помечается для удаления, но само удаление не производится, пока не будут обработаны все точки границы. Такая задержка предотвращает изменение структуры данных в процессе выполнения алгоритма. После того, как шаг 1 выполнен для всех точек границы, отмеченные элементы удаляются (значение изменяется на 0). После этого точно так же выполняется шаг 2 алгоритма.

Таким образом, одна итерация алгоритма утончения складывается из (1) применения шага 1 ко всем точкам границы с пометкой кандидатов на уда-

p_9	p_2	p_3
p_8	p_1	p_4
p_7	p_6	p_5

Рис. 11.14. Обозначения элементов окрестности в алгоритме утончения

0	0	1
1	p_1	0
1	0	1

Рис. 11.15. Иллюстрация условий (*а*) и (*б*) в (11.1-4). В данном случае $N(p_1) = 4$ и $T(p_1) = 3$

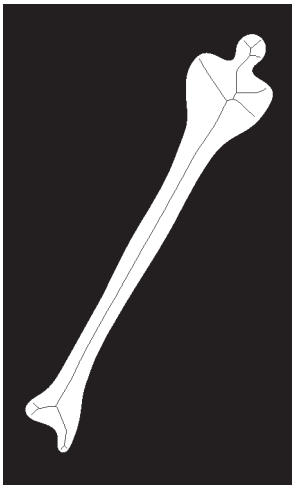


Рис. 11.16. Бедренная кость человека с наложенным остовом области

ление; (2) удаления отмеченных точек; (3) применения шага 2 с пометкой кандидатов на удаление среди оставшихся точек границы и (4) удаления отмеченных точек. Описанная процедура повторяется до тех пор, пока не прекратится процесс удаления точек. При этом алгоритм останавливается, приводя в результате к остову исходной области.

Условие (а) нарушается, если среди восьмерки соседей контурной точки p_1 имеется только один единичный элемент или только один нулевой. Первое означает, что p_1 является концевой точкой остова и, разумеется, не подлежит удалению. Удаление точки p_1, имеющей семь единичных соседей, привело бы к эрозии внутрь региона. Условие (б) не соблюдается для элементов, лежащих на линии толщиной в 1 пиксель, что препятствует разделению отрезков остова в ходе операции утончения. Минимальный набор фоновых элементов в окрестности, при котором одновременно выполняются условия (в) и (г), следующий: ($p_4 = 0$ *или* $p_6 = 0$) или ($p_2 = 0$ *и* $p_8 = 0$). То есть с учетом расположения элементов в окрестности согласно рис. 11.14 одновременно все четыре условия (а)—(г) соблюдаются для граничных точек, находящихся на нижней или правой границах либо в левых верхних углах границы. В любом из этих случаев точка p_1 не является частью остова и подлежит удалению. Аналогично условия (д) и (е) одновременно выполняются для следующего минимального набора элементов фона в окрестности: ($p_2 = 0$ *или* $p_8 = 0$) или ($p_4 = 0$ *и* $p_6 = 0$). Это соответствует точкам верхней или левой границ, а также правым нижним углам границы. Заметим, что для точек в правых верхних углах границы $p_2 = 0$ и $p_4 = 0$, т. е. условия (в) и (г) выполняются, так же как и условия (д) и (е). То же самое справедливо для левых нижних углов границы, где $p_6 = 0$ и $p_8 = 0$.

Пример 11.5. Остов области.

■ На рис. 11.16 приведено сегментированное изображение бедренной кости человека, на которое наложен вид остова данной области. По большей части этот остов совпадает с интуитивным представлением. Обратим внимание на верхнюю головку кости, выглядящую как силуэт туловища человека. На правом «плече» имеется двойная ветвь, которая, на первый взгляд, должна была быть одинарной, как и слева. Заметим, однако, что правое «плечо» несколько длиннее левого, что и явилось причиной построения алгоритмом еще одной ветви[5]. Непредсказуемое поведение такого рода часто встречается у алгоритмов скелетонизации. ■

[5] Причина появления второй ветви, по-видимому, все же не в размере «плеча», а в наличии на правом контуре небольшого выступа (в продолжение аналогии с силуэтом — в месте «локтя»). Этот эффект хорошо иллюстрируется на модельной области на рис. 11.13(*б*), где в середине прямоугольника из-за небольшого выступа на контуре возникла дополнительная ветвь. — *Прим. перев.*

11.2. Дескрипторы границ

В этом разделе будут рассмотрены некоторые подходы, применяемые для описания границы области, а в разделе 11.3 мы обратимся к дескрипторам всей области. Ряд подходов, рассмотренных в разделах 11.4 и 11.5, в равной мере применимы и к границам, и к областям.

11.2.1. Некоторые простые дескрипторы

Одним из простейших дескрипторов границы является ее *длина*. Общее число пикселей границы является грубым приближением ее длины. Для кривой, представленной цепным кодом с единичными шагами дискретизации по обоим направлениям, сумма числа вертикальных, горизонтальных и умноженных на $\sqrt{2}$ диагональных составляющих дает точное значение длины границы.

Диаметр границы B определяется соотношением

$$\text{Diam}(B) = \max_{i,j}\left[D(p_i, p_j)\right], \tag{11.2-1}$$

где D — мера расстояния (см. раздел 2.5.3), а p_i и p_j суть точки границы. Полезными дескрипторами границы являются значение ее диаметра и направление отрезка, соединяющего две экстремальные точки, которые определяют диаметр (этот отрезок называется *большой осью* границы). *Малая ось* границы определяется как отрезок, перпендикулярный большой оси и имеющий такую (минимальную) длину, что проведенный через концы обеих осей прямоугольник со сторонами, параллельными этим осям, полностью содержит в себе всю границу[6]. Упомянутый прямоугольник называется *базовым прямоугольником*, а отношение длины большой оси к длине малой — *эксцентриситетом* границы, величина которого также является полезным дескриптором.

Кривизна определяется как скорость изменения угла наклона. В общем случае трудно надежно измерить кривизну в некоторой точке дискретной границы, потому что обычно на таких границах имеются локальные «зазубрины». Тем не менее в качестве дескриптора кривизны границы часто оказывается полезным использовать разность углов наклона соседних сегментов границы (которые приближены отрезками ломаной) в точке пересечения этих отрезков. Например, вершины границ наподобие показанных на рис. 11.6(*в*) хорошо согласуются со способом описания с помощью значений кривизны. Говорят, что вершинная точка p лежит в *выпуклом* сегменте, если при обходе границы по часовой стрелке изменение угла наклона в точке p отрицательно; в противном случае точка называется лежащей в *вогнутом* сегменте. Описание с помощью кривизны в точке может быть подвергнуто дальнейшему уточнению посредством ранжирования изменений угла наклона. Например, точка p может считаться частью почти прямолинейного отрезка границы, если изменение угла наклона не превышает 10°, или же *угловой* точкой, если это изменение более 90°. Такого рода дескрипторы необходимо использовать с осторожностью, поскольку их интерпретация зависит от длины отдельных сегментов по отношению к общей длине границы.

---

[6] Не следует путать данное определение большой и малой осей с собственными осями, определение которых дается в разделе 11.4.

11.2.2. Нумерация фигур

Как объяснялось в разделе 11.1.2, вид разности первого порядка для границы, представленной цепным кодом, зависит от выбора начальной точки. *Номер фигуры* такой границы строится на основе 4-связного кода по рис. 11.3(*а*) и определяется как минимальное числовое представление разности первого порядка кода границы. *Порядком n* номера фигуры по определению называется число цифр в его записи. Более того, для замкнутой границы n — четное число, и его значение ограничивает число возможных различных фигур. На рис. 11.17 приведены все возможные фигуры 4-связного кода 4-го, 6-го и 8-го порядков вместе с их представлениями в форме цепного кода, разности первого порядка и номера фигуры. Заметим, что разность первого порядка вычисляется, считая цепной код циклической последовательностью, как говорилось в разделе 11.1.2. Хотя разность первого порядка цепного кода инвариантна относительно поворота, в общем случае код границы зависит от направления сетки дискретизации. Один из способов стандартизовать ориентацию сетки состоит в совмещении ее направлений со сторонами базового прямоугольника, рассмотренного в предыдущем разделе.

На практике исходя из желаемого порядка фигуры находят прямоугольник порядка n, эксцентриситет которого (определение дано в предыдущем разделе) ближе всего соответствует базовому прямоугольнику фигуры, и затем используют этот прямоугольник для построения сетки дискретизации. Например, при $n = 12$ все существующие прямоугольники порядка 12 (т. е. с периметром, равным 12) суть 2×4, 3×3 и 1×5. Если ближайшим по значению

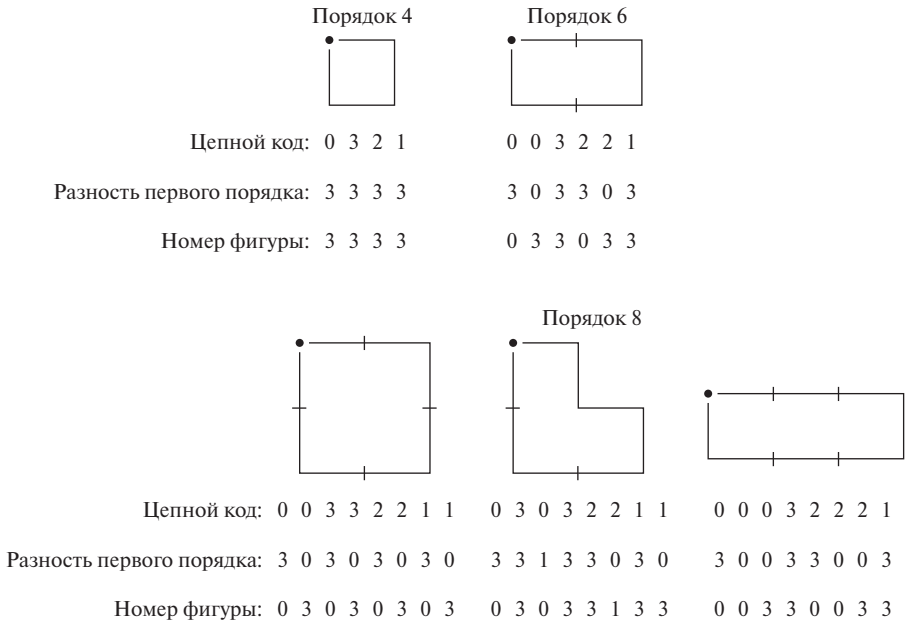


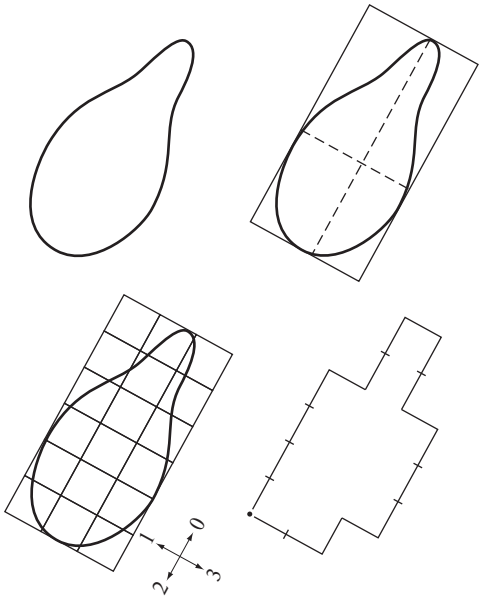
Рис. 11.17. Все возможные фигуры порядков 4, 6 и 8. Направления закодированы согласно рис. 11.3(*а*), а точкой отмечено положение начальной точки

эксцентриситета к базовому прямоугольнику заданной границы является прямоугольник 2×4, то строится выровненная по этому базовому прямоугольнику сетка 2×4 и затем применяется описанная в разделе 11.1.2 процедура построения цепного кода. Номер фигуры получается из разности первого порядка этого кода. Хотя порядок получаемого номера фигуры обычно равен n в силу выбора сетки дискретизации, при обработке границ, на которых имеются вогнутости с размерами, сравнимыми с шагом дискретизации, иногда получаются номера фигур большего, чем n порядка. В таких случаях уменьшают порядок прямоугольника n, повторяя процедуру, пока не будет получен номер фигуры, равный n.

Пример 11.6. Вычисление номеров фигур.

■ Пусть для границы, приведенной на рис. 11.18(*а*), выбрано значение $n = 18$. Чтобы получить для нее номер фигуры, необходимо проделать вышеописанную процедуру. Первый шаг состоит в нахождении базового прямоугольника, что иллюстрирует рис. 11.18(*б*). Ближайшим к нему по эксцентриситету среди прямоугольников порядка 18 является прямоугольник 3×6, поэтому строится сетка разбиения базового прямоугольника, показанная на рис. 11.18(*в*), и четыре направления цепного кода ориентируются вдоль линий сетки. Заключительный шаг состоит в построении цепного кода и использовании его разности первого порядка для вычисления номера фигуры, как показано на рис. 11.18(*г*). ■

а *б*
в *г*



Цепной код: 0 0 0 0 3 0 0 3 2 2 3 2 2 2 1 2 1 1

Разность первого порядка: 3 0 0 0 3 1 0 3 3 0 1 3 0 0 3 1 3 0

Номер фигуры: 0 0 0 3 1 0 3 3 0 1 3 0 0 3 1 3 0 3

Рис. 11.18. Шаги построения номера фигуры

11.2.3. Фурье-дескрипторы

На рис. 11.19 приведена K-точечная дискретная граница на плоскости xy. Начиная с ее произвольной точки (x_0, y_0), будем обходить границу, скажем, против часовой стрелки и обозначим координаты встречающихся точек границы (x_0, y_0), (x_1, y_1), (x_2, y_2),..., (x_{K-1}, y_{K-1}). Эти координаты можно записать в форме $x(k) = x_k$ и $y(k) = y_k$. С использованием таких обозначений границу можно представить в виде последовательности координатных пар $s(k) = [x(k), y(k)]$, где $k = 0, 1, 2,..., K–1$. Далее каждую пару координат можно рассматривать как комплексное число, так что

$$s(k) = x(k) + i\,y(k) \tag{11.2-2}$$

для $k = 0, 1, 2,..., K–1$. Таким образом, x и y рассматриваются как действительная и мнимая оси для последовательности комплексных чисел. Несмотря на изменившийся способ интерпретации этой последовательности, сущность границы осталась прежней. Конечно, такое представление имеет одно крупное преимущество: оно позволяет свести двумерную задачу к одномерной.

Согласно формуле (4.4-6), дискретное преобразование Фурье (ДПФ) конечной последовательности $s(k)$ имеет вид

$$a(u) = \sum_{k=0}^{K-1} s(k) e^{-i2\pi uk/K} \tag{11.2-3}$$

для $u = 0, 1, 2,..., K–1$. Комплексные коэффициенты $a(u)$ называются *Фурье-дескрипторами* границы. Обратное преобразование Фурье, примененное к этим коэффициентам, позволяет восстановить границу $s(k)$. То есть, согласно формуле (4.4-7),

$$s(k) = \frac{1}{K} \sum_{u=0}^{K-1} a(u) e^{i2\pi uk/K} \tag{11.2-4}$$

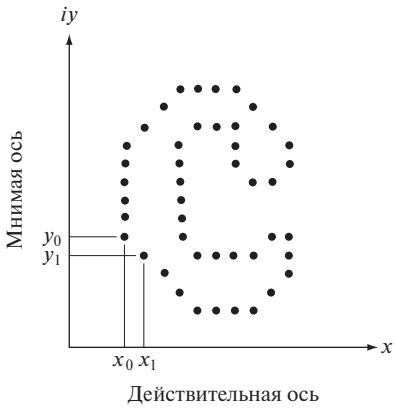


Рис. 11.19. Дискретная граница и ее представление в виде комплексной последовательности. Отмеченные точки (x_0, y_0) и (x_1, y_1) являются первыми двумя точками последовательности (начальная точка выбрана произвольно)

для $k = 0, 1, 2,..., K–1$. Предположим, однако, что вместо всех коэффициентов Фурье используются только первые P из них. Это равносильно тому, что в уравнении (11.2-4) принимается $a(u) = 0$ при $u > P–1$. Результатом восстановления окажется следующее *приближение* $s(k)$:

$$\hat{s}(k) = \frac{1}{K}\sum_{u=0}^{P-1} a(u)e^{i2\pi uk/K} \qquad (11.2\text{-}5)$$

для $k = 0, 1, 2,..., K–1$. Хотя при вычислении каждой компоненты используется лишь P членов, k по-прежнему пробегает весь диапазон от 0 до $K–1$, т. е. в приближенной границе будет *то же самое* число точек, хотя для восстановления их координат используется меньшее число членов. Вспомним из рассмотрения Фурье-преобразования в главе 4, что высокочастотные составляющие описывают мелкие детали, тогда как низкочастотные компоненты определяют общую форму границы. Следовательно, чем меньше P, тем больше деталей границы теряется, что демонстрирует следующий пример.

Пример 11.7. Использование Фурье-дескрипторов.

■ На рис. 11.20(*а*) изображена граница хромосомы человека, состоящая из $K = 2868$ точек. С помощью формулы (11.2-3) для этой границы были получены 2868 соответствующих Фурье-дескрипторов. Цель данного примера — изучить, как влияет на восстановление границы уменьшение числа используемых

а *б* *в* *г*
д *е* *ж* *з*

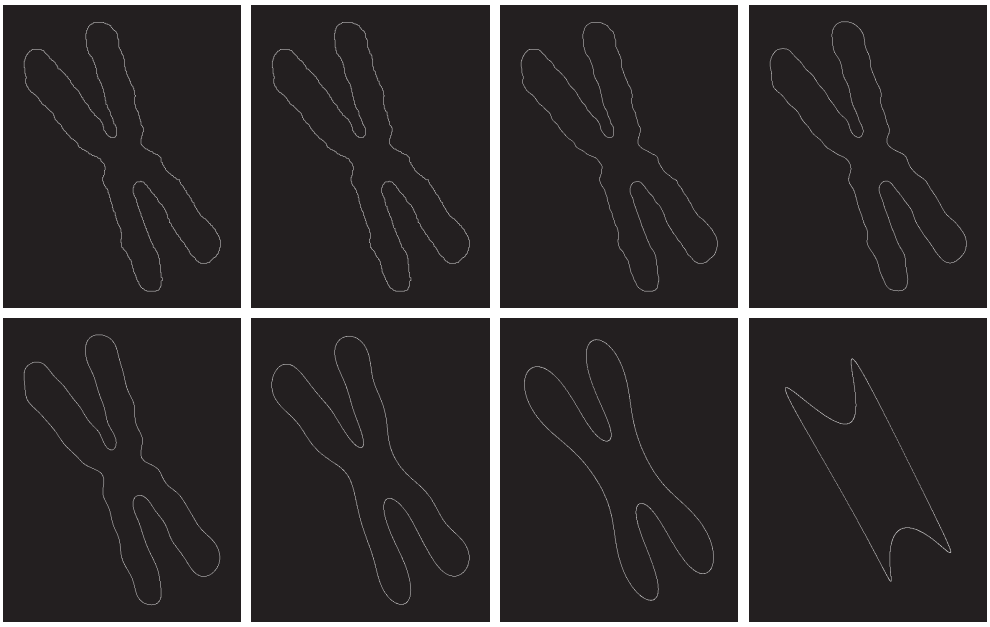


Рис. 11.20. (*а*) Граница человеческой хромосомы (содержит 2868 точек). (*б*)—(*з*) Результаты восстановления границы с использованием соответственно 1434, 286, 144, 72, 36, 18 и 8 Фурье-дескрипторов (приблизительно 50 %, 10 %, 5 %, 2,5 %, 1,25 %, 0,63 % и 0,28 % от общего числа 2868)

Фурье-дескрипторов. На рис. 11.20(*б*) представлена граница, восстановленная по половине из 2868 дескрипторов. Интересно отметить, что на ней нет видимых отличий от оригинала. На рис. 11.20(*в*)—(*з*) приведены границы, восстановленные с использованием соответственно 10 %, 5 %, 2,5 %, 1,25 %, 0,63 % и 0,28 % от всех 2868 Фурье-дескрипторов, что после округления до ближайшего четного целого составило 286, 144, 72, 36, 18 и 8 дескрипторов. Важно здесь отметить, что всего 18 дескрипторов, т. е. лишь 0,6 % от всех изначальных 2868, оказалось достаточно для передачи характерных признаков формы исходной границы — четырех длинных выступов и двух глубоких впадин. Результат на рис. 11.20(*з*), полученный с помощью 8 дескрипторов, следует признать непригодным, поскольку указанные признаки теряются. Дальнейшее уменьшение числа дескрипторов до 4 и 2 приведет к результату восстановления в форме эллипса и окружности соответственно (задача 11.13). ■

Из предыдущего примера видно, что небольшого числа Фурье-дескрипторов достаточно для описания границы по существу. Такое свойство является ценным, поскольку эти коэффициенты несут информацию о форме, и, как мы увидим в главе 12, могут служить основой для различения границ по форме.

Выше неоднократно утверждалось, что дескрипторы должны быть как можно менее чувствительными к параллельному переносу, повороту и изменению масштаба объектов. В тех случаях, когда результат зависит от порядка обработки точек границы, ставится дополнительное требование, чтобы дескрипторы не зависели от выбора начальной точки. Фурье-дескрипторы сами по себе не инвариантны к указанным геометрическим изменениям, однако измененные дескрипторы могут быть получены несложными преобразованиями. Например, применительно к повороту вспомним из основ математического анализа, что поворот точки комплексной плоскости на угол θ относительно начала координат равносилен умножению соответствующего числа на $e^{i\theta}$. Выполнение этой операции для каждой точки $s(k)$ приводит к повороту всей последовательности на угол θ относительно начала координат. Повернутая последовательность $s(k)$ $e^{i\theta}$ характеризуется Фурье-дескрипторами следующего вида:

$$a_r(u) = \sum_{k=0}^{K-1} s(k) e^{i\theta} e^{-i2\pi uk/K} = a(u) e^{i\theta} \qquad (11.2\text{-}6)$$

для $u = 0, 1, 2, \ldots, K-1$. Следовательно, поворот объекта приводит просто к умножению всех коэффициентов на одинаковую мультипликативную *константу* $e^{i\theta}$.

В табл. 11.1 приведены выражения для Фурье-дескрипторов последовательности точек границы $s(k)$ после ее поворота, параллельного переноса, изменения масштаба и смены начальной точки. Символ Δ_{xy} обозначает число $\Delta_{xy} = \Delta_x + i\Delta_y$, поэтому запись $s_t(k) = s(k) + \Delta_{xy}$ соответствует последовательности, переопределенной путем параллельного переноса:

$$s_t(k) = [x(k) + \Delta_x] + i[y(k) + \Delta_y]. \qquad (11.2\text{-}7)$$

Другими словами, параллельный перенос состоит в прибавлении постоянного смещения к координатам всех точек границы. Заметим, что параллельный перенос не оказывает влияния на все дескрипторы, кроме первого ($u = 0$), у ко-

Таблица 11.1. Основные свойства дескрипторов Фурье

Преобразование	Граница	Фурье-дескрипторы
Тождественное	$s(k)$	$a(u)$
Поворот	$s_r(k) = s(k)e^{i\theta}$	$a_r(u) = a(u)e^{i\theta}$
Параллельный перенос	$s_t(k) = s(k) + \Delta_{xy}$	$a_t(u) = a(u) + \Delta_{xy}\delta(u)$
Изменение масштаба	$s_s(k) = \alpha s(k)$	$a_s(u) = \alpha a(u)$
Смена начальной точки	$s_p(k) = s(k - k_0)$	$a_p(u) = a(u)e^{-i2\pi k_0 u/K}$

торого значение дельта-функции $\delta(u)$ будет ненулевым[7]. Наконец, выражение $s_p(k) = s(k - k_0)$ означает переопределение последовательности в соответствии с уравнением

$$s_p = x(k-k_0) + i\,y(k-k_0), \tag{11.2-8}$$

что попросту соответствует смене начальной точки последовательности с $k = 0$ на $k = k_0$. Последняя строка таблицы показывает, что изменение начальной точки влияет на все дескрипторы по-разному, хотя и известным способом, поскольку $a(u)$ умножается на член, зависящий от u.

11.2.4. Статистические характеристики

Форму участков границы (или графических представлений сигнатур) можно количественно описывать с помощью статистических характеристик, таких как среднее, дисперсия и моменты более высокого порядка. Чтобы увидеть, как это достигается, обратимся к рис. 11.21(*а*), где показан участок границы, и рис. 11.21(*б*), на котором этот участок представлен в виде одномерной функции $g(r)$ свободной переменной r. Эта функция получается путем соединения двух крайних точек границы отрезком и последующим поворотом его до горизонтального положения. Координаты точек границы поворачиваются на тот же угол.

Краткий обзор теории вероятностей имеется на интернет-сайте книги.

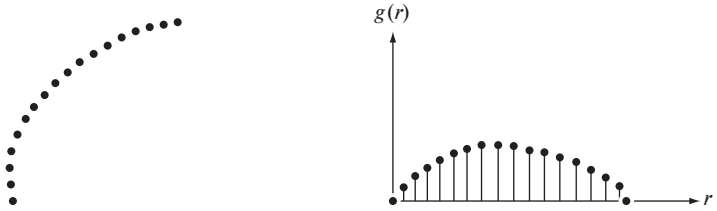
Будем рассматривать амплитуду функции g как дискретную случайную величину v. Полагая, что A — число дискретных интервалов, на которые разбит диапазон амплитуд v, построим ее гистограмму $p(v_i)$, $i = 0, 1, 2,..., A-1$. Рассматривая $p(v_i)$ как оценку вероятности появления значения v_i, согласно определению (3.3-17) можно записать следующее выражение для центрального момента порядка n случайной величины v:

$$\mu_n(v) = \sum_{i=0}^{A-1} (v_i - m)^n\, p(v_i), \tag{11.2-9}$$

где

$$m = \sum_{i=0}^{A-1} v_i\, p(v_i). \tag{11.2-10}$$

[7] Вспомним из главы 4, что преобразование Фурье от константы есть дельта-функция в начале координат, принимающая во всех остальных точках комплексной плоскости нулевые значения.



а б

Рис. 11.21. (*а*) Участок границы. (*б*) Его представление одномерной функцией

Здесь величина m представляет собой среднее, а μ_2 — дисперсию случайной величины v. В общем случае требуется лишь несколько первых моментов, чтобы различать явно отличающиеся формы сигнатур.

Альтернативный подход состоит в рассмотрении самой функции $g(r)$ как гистограммы. Нормируя ее до единичной площади, можно трактовать $g(r_i)$ как вероятность[8] появления значения r_i. В этом случае r рассматривается как случайная величина с центральными моментами

$$\mu_n(r) = \sum_{i=0}^{K-1} (r_i - m)^n g(r_i), \tag{11.2-11}$$

где

$$m = \sum_{i=0}^{K-1} r_i g(r_i). \tag{11.2-12}$$

В этой записи K — число точек границы, а $\mu_n(r)$ непосредственно связано с формой функции $g(r)$. Например, второй момент $\mu_2(r)$ характеризует разброс значений функции относительно среднего значения r, а третий момент $\mu_3(r)$ является характеристикой симметричности кривой относительно среднего значения.

По существу, нам удалось свести задачу описания двумерной границы к описанию одномерных функций. Хотя описание с помощью моментов — далеко не самый распространенный метод, это не единственные дескрипторы, применяемые для такой цели. Например, другой метод основан на вычислении спектра посредством одномерного дискретного преобразования Фурье и использовании затем первых q составляющих спектра для описания функции $g(r)$. Преимущество моментов перед другими методами заключается в простоте реализации, а также в том, что они позволяют «физически» интерпретировать форму границы. Из рис. 11.21 ясно, что данный метод инвариантен к повороту объекта, а нормировку по размерам можно при желании получить путем масштабирования диапазона значений g и r.

11.3. Дескрипторы областей

В этом разделе мы рассмотрим различные подходы, применяемые для описания областей изображения. Напомним, что общепринятая практика состоит

[8] Такая интерпретация возможна лишь при условии, что функция $g(r)$ неотрицательна, т. е. выбранный участок границы оказывается по одну сторону отрезка, соединяющего крайние точки этого участка. Кроме того, использование получаемых значений моментов для сравнения форм кривых потребует нормировки не только площади под кривой, но также и диапазона изменения значений r. — *Прим. перев.*

в использовании комбинированного описания, включающего дескрипторы как границ, так и областей.

11.3.1. Некоторые простые дескрипторы

Площадь области определяется как число содержащихся в ней пикселей. *Периметр* области есть длина ее границы. Хотя площадь и периметр иногда и применяются в качестве дескрипторов, это относится преимущественно к тем случаям, когда размеры интересующих областей не меняются. Чаще эти дескрипторы используются при вычислении меры *компактности* области, которая определяется как отношение квадрата периметра к площади. Немного отличающийся (скалярным множителем) дескриптор компактности — это *коэффициент округлости*, который определяется как отношение площади области к площади круга (максимально компактная фигура) с таким же периметром. Поскольку площадь круга с периметром P равна $P^2/4\pi$, коэффициент округлости R_c задается формулой

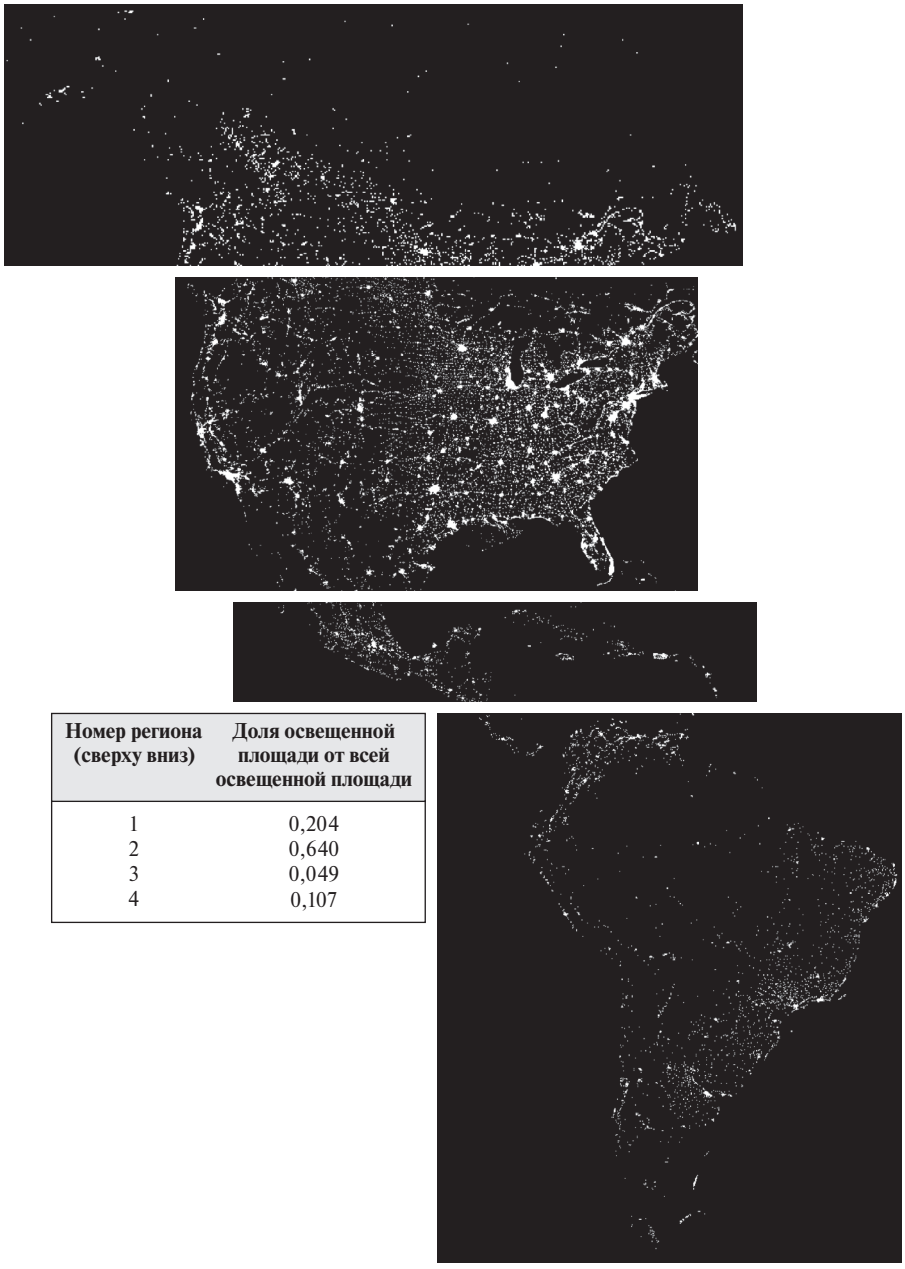
$$R_c = \frac{4\pi A}{P^2}, \tag{11.3-1}$$

где A — площадь рассматриваемой области, а P — ее периметр. Такая характеристика принимает значение 1 для круглой области и $\pi/4$ — для квадратной. Компактность является безразмерной величиной (и поэтому инвариантна к однородным изменениям масштаба). С точностью до погрешностей, возникающих при повороте дискретных областей, компактность также инвариантна к ориентации объекта.

К числу других простых дескрипторов, применяемых для описания областей, относятся различные яркостные характеристики, такие как минимальное, максимальное и среднее значения, медиана яркостей элементов области, а также число пикселей со значениями яркости больше и меньше среднего значения.

Пример 11.8. Извлечение информации из изображения с помощью вычисления площадей.

■ Даже такой простой дескриптор области, как нормированная площадь, может оказаться весьма полезным для извлечения информации из изображений. Рассмотрим в качестве примера рис. 11.22, где приведено инфракрасное спутниковое изображение американского континента. Как уже говорилось в разделе 1.3.4, подобные изображения позволяют вести глобальный учет населенных пунктов. Применяемые для регистрации таких изображений сенсоры чувствительны к излучению в видимом и ближнем инфракрасном диапазонах и позволяют фиксировать свет и огонь, в том числе кратковременные вспышки. В таблице на рисунке приведены отношения площади белого (освещенные области) по регионам к общей освещенной площади во всех четырех регионах. Даже такие простые измерения позволяют, например, получить относительные оценки энергопотребления по регионам. Эти данные можно уточнить путем нормирования по отношению к площади суши в регионе, численности населения и т. д. ■



Номер региона (сверху вниз)	Доля освещенной площади от всей освещенной площади
1	0,204
2	0,640
3	0,049
4	0,107

Рис. 11.22. Инфракрасные спутниковые изображения американского континента ночью. (Предоставлены службой NOOA)

11.3.2. Топологические дескрипторы

Для глобального описания областей на плоскости изображения часто оказываются полезными их топологические свойства. В общих чертах *топология* изучает свойства фигур, на которые не влияют любые их непрерывные деформации, то

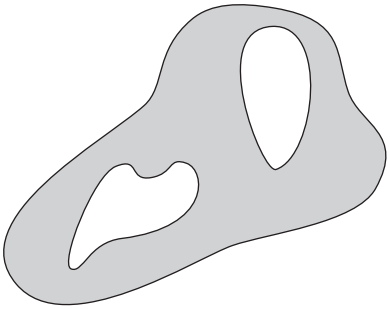


Рис. 11.23. Область с двумя дырками

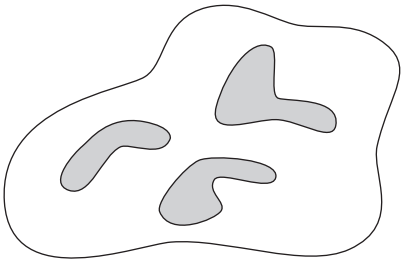


Рис. 11.24. Область, состоящая из трех компонент связности

есть такие, при которых не происходит разрывов и склеек (как будто плоскость изображения ведет себя аналогично *листу резины*). Например, на рис. 11.23 показана область с двумя дырками внутри. Если в качестве топологического дескриптора использовать число дырок внутри области, то это свойство, очевидно, будет инвариантным относительно растяжения или поворота. Однако, вообще говоря, число дырок будет меняться, если область складывается или разрывается. Заметим, что хотя при растяжении и меняются расстояния между точками области, топологические свойства не зависят ни от понятия расстояния, ни от каких-либо других свойств, неявно основанных на измерении расстояний.

Другое полезное для описания области топологическое свойство — это число ее связных компонент. Определение *компоненты связности* области было дано в разделе 2.5.2. На рис. 11.24 изображена область, состоящая из трех компонент связности (см. раздел 9.5.3, где рассматривался алгоритм выделения связных компонент).

Число дырок H и число связных компонент C некоторой фигуры используются в определении ее *числа Эйлера* E:

$$E = C - H. \tag{11.3-2}$$

Число Эйлера также является топологическим свойством. Например, показанные на рис. 11.25 области характеризуются числами Эйлера 0 и –1 соответственно, поскольку область «A» состоит из одной связной компоненты и имеет одну дырку, а область «B» также состоит из одной связной компоненты, однако имеет две дырки.

а *б*

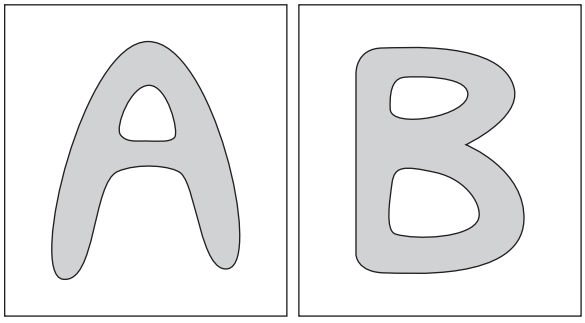


Рис. 11.25. Области, для которых числа Эйлера равны 0 и –1 соответственно

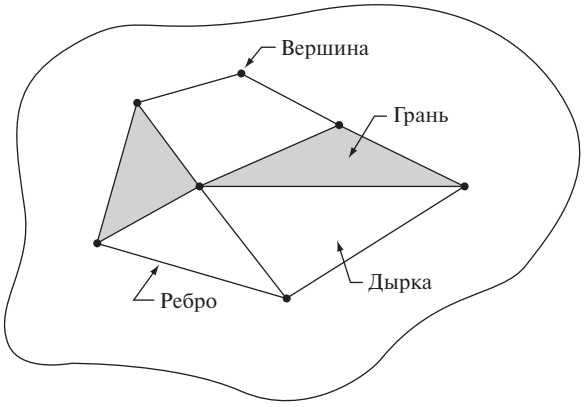


Рис. 11.26. Область, содержащая многоугольную сеть

Области, образованные отрезками прямых (такие области принято называть *многоугольными сетями*), допускают особо простую интерпретацию в терминах числа Эйлера. На рис. 11.26 показан пример многоугольной сети. Внутренние области таких многоугольных сетей часто бывает важно классифицировать как грани или дырки. Обозначая V число вершин сети, Q — число ее ребер, а F — число граней, выпишем следующее соотношение, называемое *формулой Эйлера*:

$$V - Q + F = C - H .$$

С учетом определения (11.3-2) обе части этого равенства равны числу Эйлера:

$$V - Q + F = C - H = E . \qquad (11.3\text{-}3)$$

У сети, показанной на рис. 11.26, имеется 7 вершин, 11 ребер, 2 грани, 1 связная компонента и 3 дырки; следовательно, число Эйлера равно –2:

$$7 - 11 + 2 = 1 - 3 = -2.$$

Топологические дескрипторы часто бывают полезными дополнительными признаками, характеризующими области представленной на изображении сцены.

Пример 11.9. Использование связных компонент для выделения максимально крупных признаков в сегментированном изображении.

■ На рис. 11.27(*а*) приведено полученное со спутника «Ландсат» агентства NASA 8-битовое изображение района г. Вашингтон (округ Колумбия), имеющее размеры 512×512 элементов. Это конкретное изображение было получено в ближнем инфракрасном диапазоне (подробнее см. рис. 1.10). Предположим, что мы хотим выделить область реки на основе только этого изображения (без использования других компонент многозонального изображения, что упростило бы задачу). Поскольку речной поверхности на изображении соответствует относительно темная однородная область, для ее выделения естественно попробовать применить пороговое преобразование. На рис. 11.27(*б*) показан результат применения такого преобразования с максимальным значением порога, при котором область реки еще остается связной. Этот порог был выбран вручную, чтобы на-

глядно показать, что в данном примере с помощью порогового преобразования невозможно выделить на изображении только реку, не захватив одновременно и части других областей. Цель этого примера — показать, как можно использовать компоненты связности на заключительной стадии сегментации.

В изображении на рис. 11.27(*б*) имеется 1591 связная компонента (при выделении на основе отношения 8-связности), и число Эйлера для него равно 1552, откуда можно заключить, что имеется 39 дырок. Рис. 11.27(*в*) демонстрирует компоненту связности с наибольшим числом элементов (8479). Это и есть желаемый результат, который, как мы уже знаем, невозможно получить просто сегментацией пороговым разделением. Обратите внимание на чистоту полученных данных. Если мы хотим провести некоторые измерения, например длин всех рукавов и притоков реки, можно для этого провести скелетонизацию найденной компоненты связности (рис. 11.27(*г*)); при этом длина каждой ветви остова области будет довольно точным приближением длины соответствующего участка реки. ■

а б
в г

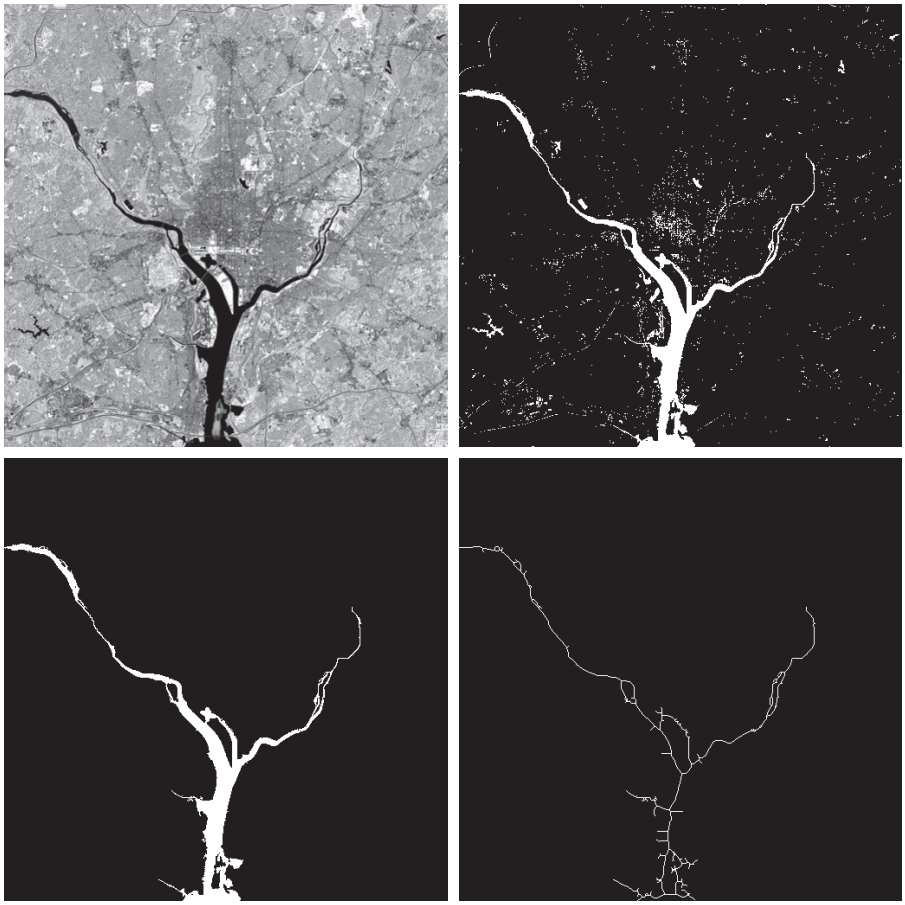


Рис. 11.27. (*а*) Инфракрасное изображение окрестностей г. Вашингтон, округ Колумбия. (*б*) Изображение после порогового преобразования. (*в*) Наиболее крупная компонента связности (*б*). (*г*) Остов области (*в*)

11.3.3. Текстурные дескрипторы

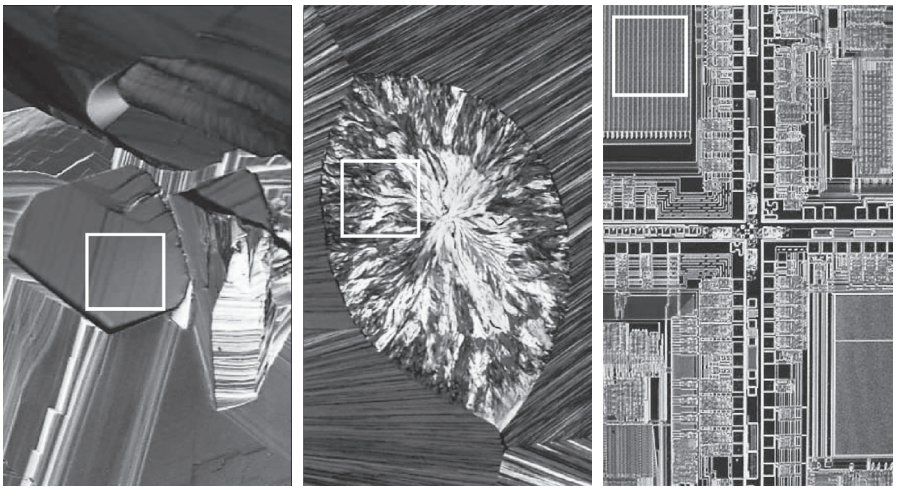
Одним из важных подходов к описанию областей является количественное представление их *текстурных признаков*. Несмотря на отсутствие формального определения текстуры, интуитивно ясно, что этот дескриптор является мерой таких свойств области, как гладкость, шероховатость и регулярность (на рис. 11.28 даны несколько примеров текстурных изображений). В цифровой обработке изображений для описания текстуры области применяются три основных подхода: статистический, структурный и спектральный. Статистические методы позволяют охарактеризовать текстуру области как гладкую, грубую, зернистую и т. д. Структурные методы занимаются изучением взаимного положения простейших составляющих изображения, как например при описании текстуры из параллельных линий, проходящих с постоянным шагом. Спектральные методы основаны на свойствах Фурье-спектра и используются прежде всего для обнаружения глобальной периодичности в изображении по имеющим большую энергию узким выбросам на спектре.

Статистический подход

Один из простейших подходов, применяемых для описания текстуры, состоит в использовании статистических характеристик, определяемых по гистограмме яркости всего изображения или его области. Пусть z — случайная величина, соответствующая яркости элементов изображения, а $p(z_i)$, $i = 0, 1, 2, ..., L-1$ — ее гистограмма, где L означает число различных уровней яркости. Согласно формуле (3.3-18), центральный момент порядка n случайной величины z равен

$$\mu_n(z) = \sum_{i=0}^{L-1} (z_i - m)^n p(z_i), \qquad (11.3\text{-}4)$$

где m — среднее значение z (средняя яркость изображения):



а б в

Рис. 11.28. Белые квадраты отмечают (слева направо) области с гладкой, грубой и периодичной текстурами. Приведены изображения сверхпроводника, человеческого холестерина и микропроцессора, полученные с помощью оптического микроскопа. (Изображения предоставил д-р Майкл У. Дэвидсон, университет шт. Флорида)

$$m = \sum_{i=0}^{L-1} z_i \, p(z_i). \tag{11.3-5}$$

Из (11.3-4) видно, что $\mu_0 = 1$ и $\mu_1 = 0$. Для описания текстуры особенно важен второй момент, т. е. *дисперсия* $\sigma^2(z) = \mu_2(z)$. Она является мерой яркостного контраста, что можно использовать для построения дескрипторов относительной гладкости. Например, величина

$$R = 1 - \frac{1}{1 + \sigma^2(z)} \tag{11.3-6}$$

равна 0 для областей постоянной яркости (где дисперсия нулевая) и приближается к 1 для больших значений $\sigma^2(z)$. Поскольку для полутоновых изображений со значениями элементов, скажем, от 0 до 255 значения дисперсии оказываются большими, для использования в уравнении (11.3-6) целесообразно нормировать дисперсию до интервала изменения [0, 1], для чего необходимо поделить $\sigma^2(z)$ на $(L - 1)^2$. Значение стандартного отклонения $\sigma(z)$ также часто используется в качестве характеристики текстуры, поскольку оно, как правило, является более наглядным.

Третий момент

$$\mu_3(z) = \sum_{i=0}^{L-1} (z_i - m)^3 \, p(z_i) \tag{11.3-7}$$

является характеристикой асимметрии гистограммы, а четвертый момент характеризует так называемый эксцесс, т. е. остроту распределения. Пятый и шестой моменты не так легко соотнести с формой гистограммы, но они, тем не менее, обеспечивают дальнейшее количественное разграничение текстурных составляющих. Среди прочих полезных характеристик текстуры, основанных на гистограмме, отметим «однородность», задаваемую выражением

$$U = \sum_{i=0}^{L-1} p^2(z_i), \tag{11.3-8}$$

и *среднюю энтропию*, которая, как известно из основ теории информации, определяется формулой

$$e(z) = -\sum_{i=0}^{L-1} p(z_i) \log_2 p(z_i). \tag{11.3-9}$$

Поскольку все значения p находятся в интервале [0, 1] и их сумма равна 1, то максимум величины U достигается для изображения, все элементы которого имеют одинаковую яркость (максимально однородное), и уменьшается по мере роста яркостных различий. Энтропия характеризует изменчивость яркости изображения; она, наоборот, равна 0 для области постоянной яркости и максимальна в случае равновероятных значений.

Пример 11.10. Характеристики текстуры, основанные на гистограмме.

■ В табл. 11.2 приведены значения описанных выше характеристик для текстур трех видов, которые были отмечены на рис. 11.28. Значение среднего характеризует просто средний уровень яркости каждой из областей и на самом деле дает лишь грубое представление об интенсивности, а не о текстуре. Стандартное отклонение значительно более информативно; цифры ясно показывают, что первая текстура характеризуется значительно меньшей изменчивостью, чем две другие (т. е. является более гладкой). Грубая текстура резко выделяется по этой характеристике. Как и следовало ожидать, то же самое справедливо для величи-

Таблица 11.2. Характеристики текстуры для областей изображений из рис. 11.28

Текстура	Среднее	Стандартное отклонение	R (нормировано)	Третий момент	Однородность	Энтропия
Гладкая	82,64	11,79	0,002	−0,105	0,026	5,434
Грубая	143,56	74,63	0,079	−0,151	0,005	7,783
Периодичная	99,72	33,73	0,017	0,750	0,013	6,674

ны R, потому что она характеризует по существу то же, что и стандартное отклонение. Третий момент в общем случае оказывается полезным для определения степени симметрии гистограммы: наблюдается ли на ней перекос влево (при отрицательном значении момента) или вправо (при положительном значении). Это дает грубое представление о «скошенности» распределения яркости в сторону светлых или темных значений от среднего. Применительно к описанию текстуры информация, содержащаяся в третьем моменте, оказывается полезной только при большой разнице значений этой характеристики. Глядя на характеристику равномерности, можно вновь заключить, что первая область является более гладкой (более равномерной, чем остальные), а наиболее случайной (наименее равномерной) оказывается область с грубой текстурой, что неудивительно. Наконец, значения энтропии располагаются в обратном порядке и приводят нас к тем же выводам, что и характеристика равномерности. Область первого изображения характеризуется наименьшей вариабельностью яркости, а область с грубой текстурой является самой изменчивой. Периодичная текстура занимает промежуточное положение по обеим этим характеристикам. ■

Текстурные характеристики, которые вычисляются только на основании гистограммы, не несут никакой информации о взаимном расположении элементов изображения. Это важно для описания текстуры, и один из способов учесть подобную информацию при анализе текстуры состоит в том, чтобы рассматривать не только распределение яркостей, но и относительное расположение пикселей в изображении.

Пусть Q — оператор, определяющий положение двух пикселей относительно друг друга, и рассматривается изображение f с возможными L градациями яркости. Пусть **G** — матрица порядка $L \times L$, составленная из элементов g_{ij}, каждый из которых показывает, сколько раз пара пикселей с яркостями z_i и z_j встретилась в изображении f со взаимным расположением, заданным оператором Q, где $1 \le i, j \le L$. Сформированная так матрица называется *матрицей сочетаемости яркостей*. Если смысл ясен из контекста, **G** называют просто *матрицей сочетаемости*[9].

> Обратите внимание, что здесь используется диапазон значений яркости [1, L] вместо привычного [0, L−1]. Это сделано для того, чтобы значения яркости соответствовали обычной индексации элементов матрицы (т. е. значение яркости 1 соответствовало левому верхнему элементу G).

[9] В оригинале — *co-occurrence matrix. — Прим. перев.*

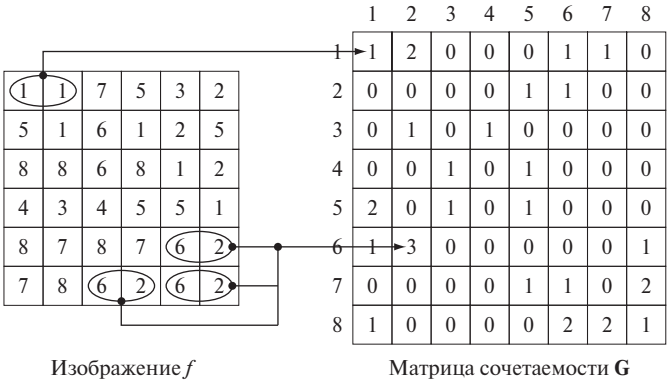


Рис. 11.29. Формирование матрицы сочетаемости яркостей

Рис. 11.29 иллюстрирует процесс построения матрицы сочетаемости при $L = 8$ и операторе взаимного расположения Q, который определяется как «на один пиксель вправо» (т. е. описывает сочетаемость пикселя с его правым соседом). Слева на рисунке представлено небольшое исходное изображение, а справа — матрица **G**. Видно, что элемент (1, 1) матрицы **G** равен 1, потому что в f только однажды встречается единичный пиксель, справа от которого также находится пиксель со значением 1. Аналогично элемент (6, 2) матрицы **G** равен 3, поскольку в f три раза встречается пиксель со значением 6, у которого сосед справа имеет значение 2. Прочие элементы **G** вычисляются аналогично. Если бы оператор Q определялся иначе, скажем, «на один пиксель вправо и вверх», то элемент (1, 1) матрицы **G** был бы равен 0, поскольку в изображении f ни разу не встречается пара единичных пикселей со взаимным расположением согласно так заданному оператору Q. С другой стороны, элементы (1, 3), (1, 5) и (1, 7) матрицы сочетаемости стали бы равны 1, потому что пиксели с единичным значением яркости по одному разу встречаются в изображении f в нужном сочетании с диагональными соседними пикселями с яркостью 3, 5, 7. В качестве упражнения читателю рекомендуется составить полную матрицу **G** для такого определения оператора Q.

Порядок матрицы **G** определяется числом градаций яркости L в исходном изображении. Для 8-битового изображения (256 градаций яркости) размеры **G** составят 256×256. Работа с одной такой матрицей не представляет трудностей, однако, как показывает пример 11.11, иногда используются последовательности матриц сочетаемости. Для уменьшения объема вычислений часто проводят вторичное квантование на меньшее число градаций яркости, чтобы размеры матрицы **G** оставались в разумных пределах. Например, в случае с 256 градациями яркости можно первым 32 из них приписать значение 1, следующим 32 — значение 2 и т. д.; при этом матрица сочетаемости будет иметь размеры 8×8.

Пусть n — число пар пикселей, которые удовлетворяют условиям оператора Q (это сумма всех элементов матрицы **G**; в вышеприведенном примере $n = 30$). Тогда величина $p_{ij} = g_{ij}/n$ будет оценкой вероятности того события, что пара пикселей со взаимным расположением согласно правилу Q имеет значения (z_i, z_j). Эти вероятности лежат в интервале [0, 1] и в сумме составляют 1:

$$\sum_{i=1}^{K}\sum_{j=1}^{K} p_{ij} = 1,$$

где K — число строк и столбцов матрицы **G**.

Поскольку **G** зависит от Q, присутствие яркостной текстуры определенного вида можно обнаружить, выбирая подходящий оператор взаимного положения и затем анализируя элементы матрицы **G**, построенной с его помощью. В табл. 11.3 приводится набор полезных дескрипторов, позволяющих охарактеризовать содержимое **G**. При вычислении дескриптора «корреляция» (во второй строке таблицы) используются величины, определяемые следующим образом:

$$m_r = \sum_{i=1}^{K} i \sum_{j=1}^{K} p_{ij}, \quad m_c = \sum_{j=1}^{K} j \sum_{i=1}^{K} p_{ij}$$

и

$$\sigma_r^2 = \sum_{i=1}^{K} (i - m_r)^2 \sum_{j=1}^{K} p_{ij}, \quad \sigma_c^2 = \sum_{j=1}^{K} (j - m_c)^2 \sum_{i=1}^{K} p_{ij}.$$

Таблица 11.3. Дескрипторы, характеризующие матрицу сочетаемости **G** (для заданного оператора Q), имеющую размеры $K \times K$. Член p_{ij} равен элементу (i, j) матрицы **G**, деленному на сумму всех ее элементов

Дескриптор	Объяснение	Формула
Максимум вероятности	Характеризует максимальный отклик матрицы **G** в интервале значений [0, 1].	$\max_{i,j}(p_{ij})$
Корреляция	Характеризует, насколько сильно пиксель коррелирован со своим «соседом» по всему изображению. Интервал возможных значений — от 1 до –1, что соответствует полной положительной и полной отрицательной корреляции. Не определена, если любое из стандартных отклонений равно 0.	$\sum_{i=1}^{K}\sum_{j=1}^{K} \frac{(i - m_r)(j - m_c)p_{ij}}{\sigma_r \sigma_c}$, $\sigma_r \neq 0, \sigma_c \neq 0$
Контраст	Мера яркостного контраста между пикселем и его «соседом» по всему изображению. Интервал возможных значений — от 0 (когда все элементы **G** равны константе) до $(K-1)^2$.	$\sum_{i=1}^{K}\sum_{j=1}^{K} (i - j)^2 p_{ij}$
Равномерность (иначе энергия)	Характеризует равномерность значений элементов. Меняется в интервале [0, 1], где значение 1 достигается, когда все элементы равны константе.	$\sum_{i=1}^{K}\sum_{j=1}^{K} p_{ij}^2$
Однородность	Характеризует пространственную близость распределения элементов **G** к диагонали. Диапазон возможных значений — [0, 1], где значение 1 достигается в случае, когда **G** — диагональная матрица.	$\sum_{i=1}^{K}\sum_{j=1}^{K} \frac{p_{ij}}{1 + \lvert i - j \rvert}$
Энтропия	Является мерой хаотичности значений элементов **G**. Энтропия равна 0, когда все p_{ij} нулевые, и достигает максимума $2\log_2 K$, когда все p_{ij} равны между собой (касательно энтропии см. соотношение (11.3-9)).	$-\sum_{i=1}^{K}\sum_{j=1}^{K} p_{ij} \log_2 p_{ij}$

Вводя обозначения

$$P(i) = \sum_{j=1}^{K} p_{ij}, \quad P(j) = \sum_{i=1}^{K} p_{ij},$$

можно переписать вышеприведенные соотношения в виде

$$m_r = \sum_{i=1}^{K} iP(i), \quad m_c = \sum_{j=1}^{K} jP(j)$$

и

$$\sigma_r^2 = \sum_{i=1}^{K} (i - m_r)^2 P(i), \quad \sigma_c^2 = \sum_{j=1}^{K} (j - m_c)^2 P(j).$$

С учетом формул (11.3-4), (11.3-5) и их объяснения видно, что m_r имеет форму среднего по строкам нормированной матрицы **G**, а m_c — среднего по столбцам[10]. Аналогично σ_r и σ_c имеют форму стандартного отклонения (квадратного корня из дисперсии), вычисленного по строкам и столбцам соответственно. Каждая из этих величин является скалярной.

При изучении табл. 11.3 следует помнить, что в ней «соседство» рассматривается с позиций заданного оператора Q (т. е. такие соседи не обязательно являются смежными элементами), а также что p_{ij} — не более чем нормированное число событий появления в изображении f пар пикселей со значениями яркости z_i и z_j, имеющих взаимное расположение в соответствии с оператором Q. Таким образом, все, что мы тут делаем, — попытка обнаружить закономерность (текстуру) по значениям этих счетчиков.

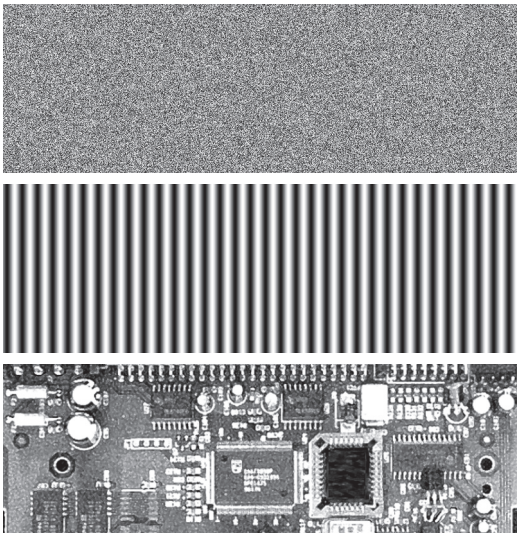
Пример 11.11. Использование дескрипторов для получения характеристик матриц сочетаемости.

■ На рис. 11.30(*а*)—(*в*) приведены изображения со случайной, горизонтально периодической (синусоидальной) и смешанной текстурами соответственно. Данный пример преследует две цели: (1) показать значения дескрипторов табл. 11.3 для матриц сочетаемости $\mathbf{G}_1$, $\mathbf{G}_2$ и $\mathbf{G}_3$ данных изображений и (2) продемонстрировать, как последовательности матриц сочетаемости могут применяться для обнаружения текстур на изображении.

На рис. 11.31 матрицы сочетаемости $\mathbf{G}_1$, $\mathbf{G}_2$ и $\mathbf{G}_3$ представлены в форме полутоновых изображений. Эти матрицы были сформированы для числа градаций $L = 256$ и оператора позиционирования вида «на один пиксель вправо». Значение с координатами (i, j) в каждом таком изображении есть число раз, когда пара пикселей с яркостями z_i, z_j встречается в соответственном исходном изображении и имеет взаимное расположение, определяемое оператором Q. Неудивительно поэтому, что изображение на рис. 11.31(*а*) является случайным, учитывая сущность того изображения, по которому оно получено.

Рис. 11.31(*б*) более интересен. Первая очевидная особенность состоит в симметрии изображения относительно главной диагонали. Благодаря симметричности синусоиды число встреч пары (z_i, z_j) такое же, как для пары (z_j, z_i), поэтому

[10] Легко видеть, что полученные значения P_i по строкам и P_j по столбцам матрицы сочетаемости (с точностью до краевых эффектов, вызванных пространственной протяженностью оператора Q) есть не что иное, как гистограмма изображения (нормированная). — *Прим. перев.*



а
б
в

Рис. 11.30. Изображения, содержащие (*а*) случайную, (*б*) периодическую и (*в*) смешанную текстуру. Размеры каждого изображения 263×800 пикселей

матрица сочетаемости оказывается симметрической. Ненулевые элементы $\mathbf{G}_2$ располагаются точечно. Для понимания этого рисунка полезно вспомнить, что в результате дискретизации и квантования синусоида превращается в ступенчатую функцию, у которой высота и ширина ступенек зависят от частоты (числа отсчетов на один период функции) и числа уровней квантования[11].

Матрица сочетаемости $\mathbf{G}_3$ (рис. 11.31(*в*)) имеет более сложную структуру. Большие значения счетчиков также группируются вблизи главной диагонали, но расположены плотнее, чем в матрице $\mathbf{G}_2$. Это говорит о том, что яркость изображения варьируется в широких пределах, но больших скачков значений между соседними пикселями немного. Из рассмотрения рис. 11.30(*в*) видно, что на нем имеются большие области с мало меняющейся яркостью. Большие скачки яркости происходят на границах объектов, но значения соответствующих счетчиков оказываются малыми по сравнению с теми, которые создаются умеренными перепадами на большой площади, и поэтому при одновременном показе на изображении эти малые значения подавляются большими, как говорилось в главе 3.

Вышеприведенные наблюдения носят качественный характер; количественный анализ содержания матриц сочетаемости требует использования дескрипторов, подобных приведенным в табл. 11.3. Значения этих дескрипторов, вычисленные для трех матриц сочетаемости, представлены в табл. 11.4. Заметим, что перед вычислением дескрипторов матрицы сочетаемости должны быть нормированы путем деления каждого элемента на сумму всех элементов, как

[11] Синусоидальное изображение на рис. 11.30(*б*), по-видимому, было синтезировано в компьютере дублированием строки одномерного синуса, что привело к *точному совпадению* значений сигнала во всех строках, а значит, и повторению (наращиванию) значений, суммируемых в одной и той же ячейке матрицы сочетаемости. — *Прим. перев.*

а б в

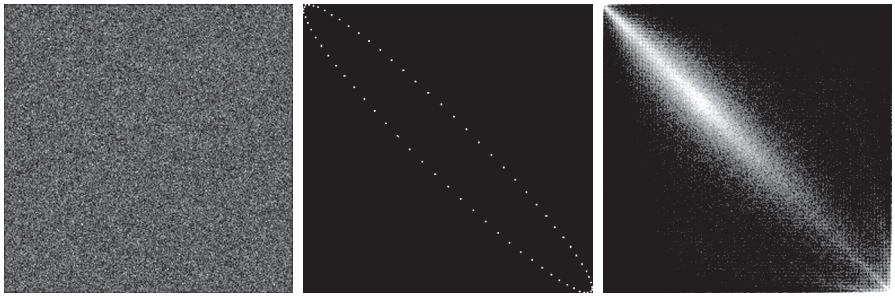


Рис. 11.31. Матрицы сочетаемости $\mathbf{G}_1$, $\mathbf{G}_2$ и $\mathbf{G}_3$ размерами 256×256, соответствующие изображениям (*а*), (*б*), (*в*) на рис. 11.30

говорилось выше. Указанные в таблице значения согласуются с тем, что можно было ожидать по виду исходных изображений на рис. 11.30 и соответствующих матриц сочетаемости на рис. 11.31. Рассмотрим, например, столбец «Максимум вероятности». Самое большое значение достигается для третьей матрицы; это говорит о том, что ей соответствуют самые большие значения счетчиков, т. е. в изображении встречается большее число пар пикселей с повторяющимися значениями, отвечающих оператору Q, по сравнению с двумя другими матрицами. Это согласуется с проведенным выше анализом $\mathbf{G}_3$. Значения в третьем столбце табл. 11.4 показывают, что максимальная корреляция достигается для $\mathbf{G}_2$; это говорит о том, что значения яркости второго исходного изображения сильно коррелированы. Повторение синусоидального изменения яркости снова и снова на рис. 11.30(*б*) объясняет, почему это так. Отметим, что для матрицы $\mathbf{G}_1$ корреляция практически нулевая, что указывает на отсутствие корреляции между значениями соседних пикселей — характерное свойство случайных изображений, подобных рис. 11.30(*а*).

Дескриптор контраста оказывается максимальным для $\mathbf{G}_1$ и минимальным для $\mathbf{G}_2$, исходя из чего можно сделать вывод, что чем менее случайным является изображение, тем обычно меньше значение его дескриптора контраста. Объяснение этому можно дать на основании рассмотрения матриц на рис. 11.31. Члены $(i-j)^2$ одинаковы для всех матриц, поэтому величина контраста определяется значениями вероятностей в нормированной матрице сочетаемости. Хотя для $\mathbf{G}_1$ значение максимума вероятности минимально, две другие матрицы содержат намного больше нулевых или близких к нулю элементов (темные области на рис. 11.31). Учитывая, что сумма всех элементов нормированной матрицы

Таблица 11.4. Значения дескрипторов, вычисленные для матриц сочетаемости на рис. 11.31

Нормированная матрица сочетаемости	Дескрипторы					
	Максимум вероятности	Корреляция	Контраст	Равномерность	Однородность	Энтропия
$\mathbf{G}_1/n_1$	0,00006	−0,0005	10838	0,00002	0,0366	15,75
$\mathbf{G}_2/n_2$	0,01500	0,9650	570	0,01230	0,0824	6,43
$\mathbf{G}_3/n_3$	0,06860	0,8798	1356	0,00480	0,2048	13,58

равна 1, легко понять, почему дескриптор контраста возрастает с увеличением случайности[12].

Поведение трех оставшихся дескрипторов можно объяснить аналогичным образом. Равномерность растет как квадрат значений вероятности, поэтому чем менее случайным является изображение, тем выше его равномерность, как показывает пятый столбец таблицы. Однородность характеризует сосредоточенность значений **G** вблизи главной диагонали. Величина знаменателя $(1 + |i - j|)$ одинакова для всех трех матриц сочетаемости и уменьшается по мере сближения значений i и j (т. е. с приближением к главной диагонали). Поэтому матрице с бóльшими значениями вероятности (стоящими в числителе) вблизи диагонали будет отвечать и бóльшая величина дескриптора однородности. Как говорилось выше, такая матрица характерна для изображения с большим содержанием полутонов и областей мало меняющейся яркости. С такой интерпретацией согласуются значения в шестом столбце табл. 11.4.

Дескриптор в последнем столбце характеризует степень случайности в матрице сочетаемости, что в свою очередь определяется случайностью исходного изображения. Как и следовало ожидать, наибольшее значение достигается для матрицы $\mathbf{G}_1$, полученной по абсолютно случайному изображению. Другие два значения также не нуждаются в объяснении. Обратите внимание, что энтропия $\mathbf{G}_1$ близка к теоретическому максимуму для 256-градационного изображения, равному 16; ($2 \log_2 256 = 16$). Изображение на рис. 11.30(*а*) образовано равномерным шумом, поэтому все значения яркости приблизительно равновероятны, что является условием максимума энтропии в табл. 11.3.

До сих пор рассматривались одиночные изображения и их матрицы сочетаемости. Пусть теперь мы хотим выяснить (не глядя на изображение), содержит ли оно области с повторяющимися компонентами, т. е. периодические текстуры. Одним из способов решения такой задачи может стать вычисление дескриптора корреляции для последовательности матриц сочетаемости, полученных по этому изображению с постепенным увеличением расстояния между «соседями» в операторе Q. Как уже отмечалось, при работе с последовательностями матриц сочетаемости обычно переходят к более грубому квантованию яркости, чтобы уменьшить размеры матриц и сопутствующий объем вычислений. Приводимые ниже результаты были получены при числе уровней квантования $L = 8$.

На рис. 11.32 приведены графики величины дескриптора корреляции в зависимости от горизонтального «смещения соседа» (т. е. расстояния по горизонтали между соседями) в интервале от 1 (примыкающие пиксели) до 50. На рис. 11.32(*а*) все значения корреляции близки к 0, что указывает на отсутствие какой-либо периодической текстуры в случайном изображении. Вид графика корреляции на рис. 11.32(*б*) ясно показывает, что исходное изображение содержит периодичность в горизонтальном направлении. Обратите внимание, что функция корреляции начинается с большого значения, которое уменьшается по мере увеличения расстояния между «соседями», затем снова возрастает и т. д.

---

[12] Значение дескриптора контраста возрастает, конечно, не с увеличением случайности, а с увеличением значений членов матрицы сочетаемости, отстоящих далеко от ее главной диагонали, которые соответствуют вероятностям появления пар пикселей с большой разницей в значениях яркостей. — *Прим. перев.*

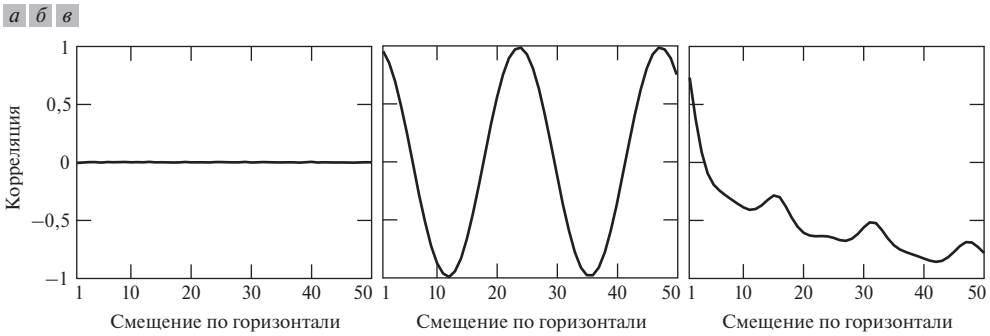


Рис. 11.32. Зависимость корреляционного дескриптора от смещения (расстояния между «соседними» пикселями) для (*а*) шумового изображения, (*б*) изображения с синусоидальной текстурой и (*в*) изображения монтажной платы, приведенных на рис. 11.30

Рис. 11.33. Увеличенный фрагмент изображения монтажной платы, на котором видна периодическая структура

Как показывает рис. 11.32(*в*), для изображения монтажной платы корреляционный дескриптор сначала уменьшается, но дальше демонстрирует отчетливый пик при величине смещения 16 пикселей. Анализ изображения на рис. 11.30(*в*) показывает, что расположенный в верхней части ряд точек пайки образует повторяющуюся структуру с периодом около 16 пикселей (рис. 11.33). Следующий заметный пик имеется при смещении 32 пикселей; причиной его является та же структура, но амплитуда пика меньше, так как число повторений на таком расстоянии меньше, чем при смещении 16 пикселей. Аналогично объясняется еще более слабый пик при смещении 48 пикселей. ■

На исходном изображении имеются и другие повторяющиеся структуры, но они пропали в результате огрубленного квантования исходных 256 градаций в 8 уровней.

Структурный подход

Как отмечалось в начале данного раздела, второй класс описаний текстур основан на структурном подходе. Пусть имеется правило, записанное в форме[13] $S \rightarrow aS$, которое указывает, что символ S разрешается переписывать в виде aS (например, троекратное применение такого правила дает строку символов $aaaS$). Если a

[13] Такая форма записи широко используется в теории формальных грамматик и определяет так называемые *правила переписывания* (или *продукции*). Смысл нотации в том, что цепочку символов, записанную слева от знака →, разрешается заменять на цепочку символов, стоящую справа от этого знака. — *Прим. перев.*

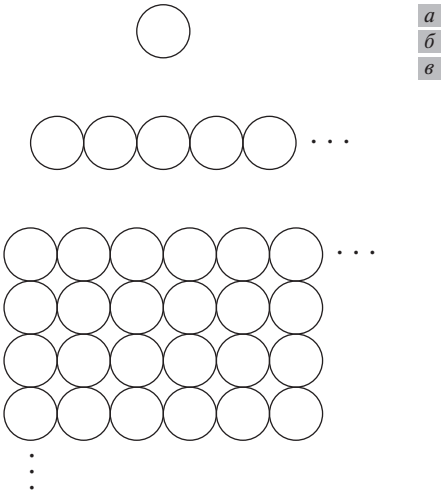


Рис. 11.34. (*а*) Базовый элемент текстуры. (*б*) Текстура, строящаяся применением правила $S \to aS$. (*в*) Двумерная текстура, сгенерированная с помощью этого и других правил

символически представляет круг (рис. 11.34(*а*)) и строке *aaa*... придан смысл «последовательность кругов вправо от начальной точки», то применение правила $S \to aS$ приводит к построению текстуры, изображенной на рис. 11.34(*б*).

Далее добавим к этой схеме еще несколько новых правил: $S \to bA$, $A \to cA$, $A \to c$, $A \to bS$, $S \to a$. Здесь появление *b* интерпретируется как «круг ниже», а *c* — как «круг слева». С помощью таких правил можно построить символьную строку *aaabccbaa*, которой соответствует построенная из кругов матрица 3×3. Таким же образом могут генерироваться и более крупные текстурные образы, например как на рис. 11.34(*в*). (Отметим, однако, что по этим правилам можно сгенерировать и структуры, не являющиеся прямоугольными.)

Основная идея вышеприведенного обсуждения состоит в том, что из относительно простых базовых элементов текстуры можно формировать более сложные текстурные образы с помощью некоторых правил, ограничивающих возможное взаимное расположение этих базовых элементов. Эти принципы лежат в основе реляционных дескрипторов — темы, которую мы подробнее рассмотрим в разделе 11.5.

Спектральный подход

Как указывалось в разделе 5.4, спектр Фурье идеально подходит для описания направленности присутствующих в изображении периодических или квазипериодических двумерных структур. Эти глобальные текстурные образы легко различаются на спектре в виде импульсов с высокой энергией. Мы рассмотрим здесь следующие три свойства Фурье-спектра, полезные для описания текстуры. (1) Угловая координата выступающего пика спектра (в полярном представлении) указывает направление соответствующей текстурной составляющей. (2) Местоположение этих пиков на частотной плоскости дает основной пространственный период текстуры. (3) После устранения всех периодических составляющих путем фильтрации в изображении остаются только непериодиче-

ские компоненты, которые затем могут описываться с помощью статистических методов. Напомним, что амплитуда спектра симметрична относительно начала координат, так что достаточно рассматривать только половину частотной плоскости. Таким образом, в нашем анализе каждая периодическая компонента текстуры связана только с одним пиком спектра, а не двумя.

Обнаружение и интерпретация вышеупомянутых спектральных признаков часто упрощается при переходе к полярным координатам, в которых спектральная функция выражается в виде $S(r, \theta)$, где r и θ — переменные данной системы координат. Для каждого угла θ функция $S(r, \theta)$ может рассматриваться как одномерная функция $S_\theta(r)$. Аналогично для каждого значения частоты r, $S_r(\theta)$ является одномерной функцией. Анализ функции $S_\theta(r)$ при фиксированном θ дает картину поведения спектра (скажем, наличие пиков) по направлению радиуса из начала координат, а исследуя $S_r(\theta)$при фиксированном r, получаем поведение спектральной функции вдоль окружности с центром в начале координат.

Более общее описание получается интегрированием этих функций (которое в рассматриваемом дискретном случае заменяется суммированием):

$$S(r) = \sum_{\theta=0}^{\pi} S_\theta(r) \tag{11.3-10}$$

и

$$S(\theta) = \sum_{r=1}^{R_0} S_r(\theta), \tag{11.3-11}$$

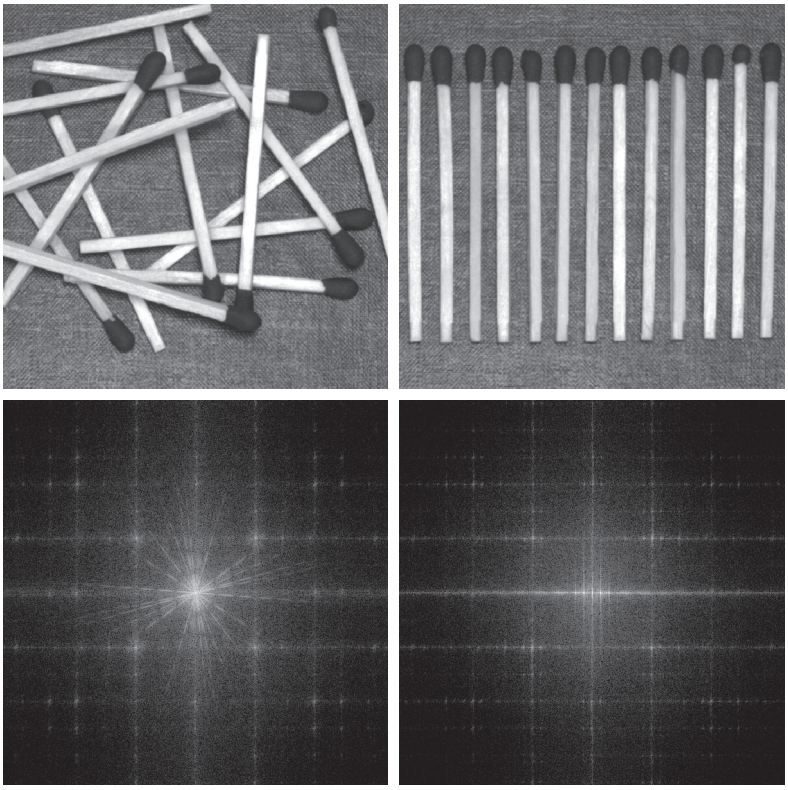
где R_0 — радиус круга с центром в начале координат.

Результатом вычислений по формулам (11.3-10) и (11.3-11) является получение пары значений ($S(r)$, $S(\theta)$) для *каждой* точки плоскости с координатами (r, θ). Варьируя эти координаты, можно построить две одномерные функции $S(r)$ и $S(\theta)$, описывающие текстуру всего изображения или области интереса в терминах энергии спектра. После этого уже можно вычислять те или иные дескрипторы самих этих функций, количественно характеризующие поведение последних. Для этих целей обычно используются такие дескрипторы, как среднее, положение максимума и дисперсия, а также разность между средним и максимальным значениями функции.

Пример 11.12. Спектральный анализ текстуры.

■ На рис. 11.35(*а*) показано изображение случайно разбросанных спичек, а на рис. 11.35(*б*) — изображение с теми же спичками, расположенными в виде периодической структуры. Рис. 11.35(*в*) и (*г*) демонстрируют соответствующие спектры Фурье. На обоих спектрах видны периодические всплески энергии, расположенные в виде крупной четырехугольной сетки по обоим координатным направлениям; они вызваны мелкой периодической структурой материала, на котором лежат спички. Прочие выраженные компоненты спектра в виде лучей из центра на рис. 11.35(*в*) появились вследствие случайной ориентации контуров объектов на рис. 11.35(*а*). Напротив, на рис. 11.35(*г*) основные энергетические компоненты, помимо вызванных фоном, располагаются вдоль горизонтальной оси, что соответствует мощным вертикальным контурам на рис. 11.35(*б*).

Графики функций $S(r)$ и $S(\theta)$ для случайно разбросанных спичек показаны соответственно на рис. 11.36(*а*) и (*б*), а для упорядоченных — на рис. 11.36(*в*) и (*г*).



а *б*
в *г*

Рис. 11.35. (*а*) и (*б*) Изображения объектов со случайным и упорядоченным расположением. (*в*) и (*г*) Соответствующие Фурье-спектры. Размеры всех изображений — 600×600 пикселей

В первом случае на графике $S(r)$ не видно периодических компонент, т. е. выступающих пиков спектра, за исключением пика в начале координат, вызванного постоянной составляющей. Напротив, график $S(r)$ в случае упорядоченных спичек демонстрирует выступающий пик вблизи значения $r = 15$ и несколько меньший вблизи $r = 25$, что соответствует периоду светлых (спички) и темных (фон) областей на рис. 11.35(*б*). Точно так же на графике $S(\theta)$ рис. 11.36(*б*) видна случайность расположения всплесков энергии[14]. Напротив, график $S(\theta)$ на рис. 11.36(*г*) демонстрирует мощные всплески энергии вблизи начала координат и при углах 90° и 180°, что согласуется с распределением энергии в спектре на рис. 11.35(*г*). ■

11.3.4. Инварианты моментов двумерных функций

Двумерный момент порядка $(p + q)$ для цифрового изображения $f(x, y)$ размерами $M \times N$ определяется как

$$m_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} x^p y^q f(x, y) \qquad (11.3\text{-}12)$$

[14] Положение этих пиков соответствует углам (нормали), под которыми спички расположены на изображении. — *Прим. перев.*

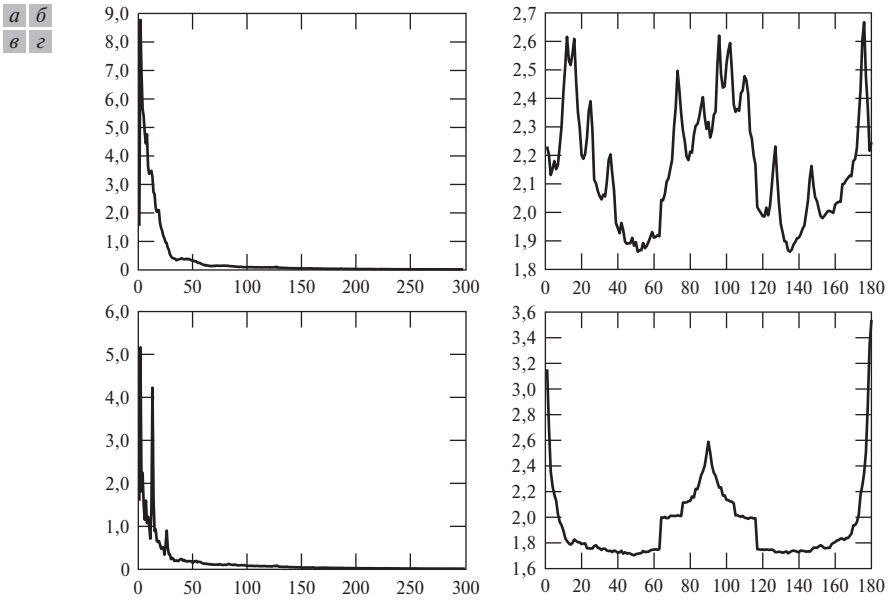


Рис. 11.36. (*а*) и (*б*) Графики функций $S(r)$ и $S(\theta)$ для изображения на рис. 11.35(*а*). (*в*) и (*г*) Графики функций $S(r)$ и $S(\theta)$ для изображения на рис. 11.35(*б*). Масштаб по вертикальным осям всюду $\times 10^5$

для целочисленных $p, q = 0, 1, 2, \ldots$. Соответствующие *центральные моменты* порядка $(p + q)$ определяются следующим образом:

$$\mu_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (x - \bar{x})^p (y - \bar{y})^q f(x, y), \tag{11.3-13}$$

для $p, q = 0, 1, 2, \ldots$, где

$$\bar{x} = \frac{m_{10}}{m_{00}} \quad \text{и} \quad \bar{y} = \frac{m_{01}}{m_{00}}. \tag{11.3-14}$$

Нормированные центральные моменты, обозначаемые η_{pq}, определяются как

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}}, \tag{11.3-15}$$

где

$$\gamma = \frac{p+q}{2} + 1 \tag{11.3-16}$$

для $p + q = 2, 3, \ldots$.

С использованием моментов второго и третьего порядков может быть выведен следующий набор из семи *инвариантных моментов*[15].

$$\phi_1 = \eta_{20} + \eta_{02} \tag{11.3-17}$$

[15] При выводе этих результатов используются понятия, выходящие за рамки нашего обсуждения, которые подробно рассмотрены в книге Белла [Bell, 1965] и статье Ху [Hu, 1962]. Для построения инвариантов моментов выше седьмого порядка см. [Flusser, 2000]. Инварианты моментов допускают обобщение и на n-мерный случай [Mamistvalov, 1998].

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \tag{11.3-18}$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (\eta_{03} - 3\eta_{21})^2 \tag{11.3-19}$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \tag{11.3-20}$$

$$\begin{aligned}\phi_5 = {} & (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] + \\ & + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})\left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right]\end{aligned} \tag{11.3-21}$$

$$\phi_6 = (\eta_{20} - \eta_{02})\left[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \tag{11.3-22}$$

$$\begin{aligned}\phi_7 = {} & (3\eta_{21} - \eta_{03})(\eta_{12} + \eta_{30})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] + \\ & + (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03})\left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right]\end{aligned} \tag{11.3-23}$$

Этот набор моментов является инвариантным по отношению к параллельному переносу, зеркальному отражению (с точностью до перемены знака), повороту и изменению масштаба.

Пример 11.13. Инварианты двумерных моментов.

■ Цель этого примера — вычислить и сравнить вышеприведенные инварианты моментов, используя изображение на рис. 11.37(*а*). Черный бордюр (с нуле-

а *б* *в*
г *д* *е*

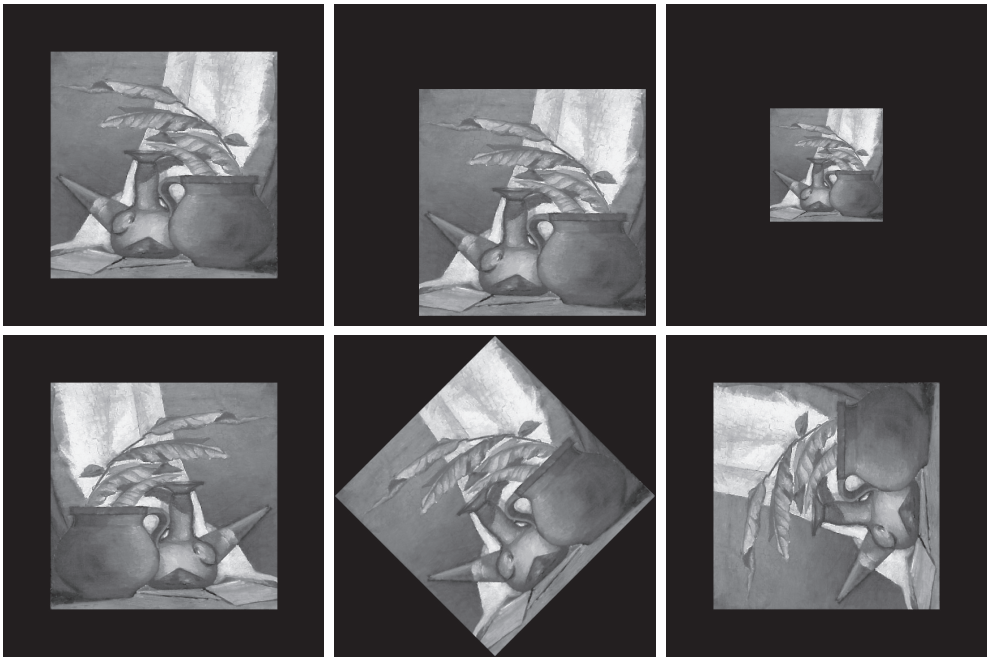


Рис. 11.37. (*а*) Исходное изображение. (*б*)—(*е*) Результаты сдвига, уменьшения в два раза, зеркального отражения, поворота на 45° и на 90° соответственно

Таблица 11.5. Инварианты моментов для изображений на рис. 11.37.

Инвариант	Исходное изображение	Сдвиг	Половинный размер	Зеркальное отражение	Поворот на 45°	Поворот на 90°
ϕ_1	2,8662	2,8662	2,8664	2,8662	2,8661	2,8662
ϕ_2	7,1265	7,1265	7,1257	7,1265	7,1266	7,1265
ϕ_3	10,4109	10,4109	10,4047	10,4109	10,4115	10,4109
ϕ_4	10,3742	10,3742	10,3719	10,3742	10,3742	10,3742
ϕ_5	21,3674	21,3674	21,3924	21,3674	21,3663	21,3674
ϕ_6	13,9417	13,9417	13,9383	13,9417	13,9417	13,9417
ϕ_7	–20,7809	–20,7809	–20,7724	20,7809	–20,7813	–20,7809

выми значениями) был добавлен, чтобы привести все изображения в примере к одному размеру; эти нулевые значения не влияют на вычисление инвариантов двумерных моментов. Рис. 11.37(*б*)—(*е*) демонстрируют сдвиг исходного изображения, уменьшение в два раза по обоим пространственным направлениям, зеркальное отражение, поворот на 45° и поворот на 90° соответственно. В табл. 11.5 представлены значения семи инвариантов двумерных моментов для этих шести изображений. Чтобы сузить динамический диапазон и тем самым упростить интерпретацию результатов, приводятся величины $\text{sgn}(\phi_i)\lg(|\phi_i|)$. Взятие модуля необходимо, поскольку многие значения являются дробными и/или отрицательными, а для сохранения знака служит функция sgn(·) (интерес здесь представляют прежде всего *инвариантность* и соотношение знаков моментов, а не фактические значения). Два ключевые момента, следующие из табл. 11.5, — это (1) близость значений моментов, несмотря на сдвиг, изменение масштаба, зеркальное отражение и поворот; и (2) тот факт, что знак ϕ_7 меняется при зеркальном отражении (это свойство применяется на практике для проверки, не является ли изображение зеркальным). ■

11.4. Использование главных компонент для описания

Материал этого раздела можно применять и для границ, и для областей. Кроме этого, на его основе можно описывать наборы изображений, которые пространственно совмещены в момент регистрации, но имеют отличающиеся значения пикселей (как например три цветовые составляющие RGB в цветном изображении). Предположим, что даны три такие цветовые компоненты. Эти изображения можно рассматривать как единое целое, если считать вектором любую группу, состоящую из трех соответственных пикселей в каждом из них. Например, пусть x_1, x_2 и x_3 — значения пикселя в каждом из трех изображений компонент RGB. Эти три элемента можно записать в форме *вектор-столбца* $\mathbf{x}$ размерности 3:

Краткий обзор, посвященный векторам и матрицам, приведен на интернет-сайте книги.

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}.$$

Один такой вектор представляет *один* общий пиксель во всех трех изображениях. Если эти изображения имеют одинаковый размер $M \times N$, то после представления аналогичным образом всех их пикселей получим $K = MN$ трехмерных векторов. Если регистрируется не 3, а n совмещенных изображений, векторы станут n-мерными:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}. \tag{11.4-1}$$

Всюду на протяжении этого раздела мы будем считать, что все векторы являются вектор-столбцами (т.е. матрицами порядка $n \times 1$). Записывая их в строке текста, будем использовать нотацию $\mathbf{x} = (x_1, x_2, ..., x_n)^T$, где T означает операцию транспонирования.

Будем считать эти векторы реализациями случайной величины точно так же, как это делалось при построении гистограммы яркостей. Разница состоит только в том, что теперь вместо таких характеристик, как среднее значение и дисперсия случайной величины, мы будем говорить о *векторе математического ожидания* и *ковариационной матрице* случайных векторов. Вектор математического ожидания для генеральной совокупности определяется как

$$\mathbf{m}_\mathbf{x} = E\{\mathbf{x}\}, \tag{11.4-2}$$

где $E\{\cdot\}$ есть ожидаемое значение аргумента, а индекс означает, что $\mathbf{m}$ связан с генеральной совокупностью векторов $\mathbf{x}$. Напомним, что ожидаемое значение вектора или матрицы формируется как набор независимых математических ожиданий их компонентов.

Ковариационная матрица для генеральной совокупности векторов определяется как

$$\mathbf{C}_\mathbf{x} = E\left\{(\mathbf{x} - \mathbf{m}_\mathbf{x})(\mathbf{x} - \mathbf{m}_\mathbf{x})^T\right\}. \tag{11.4-3}$$

Поскольку $\mathbf{x}$ есть n-мерный вектор, то $(\mathbf{x} - \mathbf{m}_\mathbf{x})(\mathbf{x} - \mathbf{m}_\mathbf{x})^T$ и $\mathbf{C}_\mathbf{x}$ — матрицы порядка $n \times n$. Элемент c_{ii} матрицы $\mathbf{C}_\mathbf{x}$ есть дисперсия x_i, т.е. i-й компоненты векторов $\mathbf{x}$ генеральной совокупности, а элемент c_{ij} матрицы $\mathbf{C}_\mathbf{x}$ есть ковариация[16] компонент x_i и x_j этих векторов. Значения матрицы $\mathbf{C}_\mathbf{x}$ действительные и симметричны относительно главной диагонали. Если компоненты x_i и x_j являются некоррелированными, то значение их ковариации равно нулю и, следовательно, $c_{ij} = c_{ji} = 0$. Все эти определения при $n = 1$ сводятся к хорошо знакомым одномерным эквивалентам.

---

[16] Напомним, что дисперсия случайной величины x, имеющей среднее значение m, определяется как $E\{(x - m)^2\}$. Ковариация (второй смешанный момент) двух случайных величин x_i и x_j определяется как $E\{(x_i - m_i)(x_j - m_j)\}$.

На основании выборки K векторов из генеральной совокупности приближенная оценка вектора математического ожидания находится с помощью обычного усреднения:

$$\mathbf{m}_\mathbf{x} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{x}_k . \tag{11.4-4}$$

Аналогично, раскрывая произведение $(\mathbf{x} - \mathbf{m}_\mathbf{x})(\mathbf{x} - \mathbf{m}_\mathbf{x})^T$ и используя равенства (11.4-2) и (11.4-4), получаем следующую выборочную оценку для ковариационной матрицы:

$$\mathbf{C}_\mathbf{x} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{x}_k\mathbf{x}_k^T - \mathbf{m}_\mathbf{x}\mathbf{m}_\mathbf{x}^T . \tag{11.4-5}$$

Пример 11.14. Вычисление вектора математического ожидания и матрицы ковариации.

■ В качестве иллюстрации соотношений (11.4-4) и (11.4-5) рассмотрим четыре вектора $\mathbf{x}_1 = (0, 0, 0)^T$, $\mathbf{x}_2 = (1, 0, 0)^T$, $\mathbf{x}_3 = (1, 1, 0)^T$ и $\mathbf{x}_4 = (1, 0, 1)^T$. Применяя равенство (11.4-4), получаем следующий вектор математического ожидания:

$$\mathbf{m}_\mathbf{x} = \frac{1}{4}\begin{bmatrix}3\\1\\1\end{bmatrix}.$$

Аналогичным образом с помощью соотношения (11.4-5) вычисляем следующую матрицу ковариации:

$$\mathbf{C}_\mathbf{x} = \frac{1}{16}\begin{bmatrix}3 & 1 & 1\\1 & 3 & -1\\1 & -1 & 3\end{bmatrix}.$$

Все элементы на главной диагонали матрицы равны между собой; это означает, что дисперсия всех компонент векторов генеральной совокупности одинакова. Кроме того, корреляция компонент x_1 и x_2, как и x_1, и x_3, положительная, а корреляция компонент x_2 и x_3 отрицательная. ■

Поскольку матрица $\mathbf{C}_\mathbf{x}$ является действительной и симметричной, то для нее всегда существует ортонормированный базис, состоящий из n собственных векторов [Noble, Daniel, 1988]. Пусть $\mathbf{e}_i$ и λ_i, $i = 1, 2,..., n$, — набор собственных векторов и соответствующие им собственные значения[17] матрицы $\mathbf{C}_\mathbf{x}$, которые для удобства упорядочим по убыванию, так что $\lambda_j \geq \lambda_{j+1}$ для $j = 1, 2,..., n-1$. Образуем матрицу $\mathbf{A}$ из собственных векторов $\mathbf{C}_\mathbf{x}$, располагая их по строкам таким образом, чтобы в первой строке записывался вектор, которому отвечает наибольшее собственное значение, а в последней — собственный вектор, соответствующий наименьшему собственному значению.

Будем использовать $\mathbf{A}$ в качестве матрицы преобразования, которое отображает векторы $\mathbf{x}$ в векторы $\mathbf{y}$ по следующему закону:

$$\mathbf{y} = \mathbf{A}(\mathbf{x} - \mathbf{m}_\mathbf{x}). \tag{11.4-6}$$

Это выражение называется *преобразованием Хотеллинга*, которое, как мы скоро увидим, обладает рядом интересных и полезных свойств.

---

[17] По определению, собственные векторы и собственные значения матрицы $\mathbf{C}$ порядка $n \times n$ удовлетворяют условиям $\mathbf{C}\mathbf{e}_i = \lambda_i\mathbf{e}_i$ для $i = 1, 2,..., n$.

Преобразование Хотеллинга идентично дискретному преобразованию Карунена—Лоэва [Karhunen, 1947], поэтому в литературе можно встретить как одно, так и другое название.

Нетрудно показать, что получаемые в результате такого преобразования векторы **y** имеют нулевое математическое ожидание, т. е.

$$\mathbf{m}_{\mathbf{y}} = E\{\mathbf{y}\} = \mathbf{0}. \tag{11.4-7}$$

Из элементарной теории матриц следует, что ковариационная матрица генеральной совокупности векторов **y** выражается через матрицы **A** и $\mathbf{C}_{\mathbf{x}}$ следующим образом:

$$\mathbf{C}_{\mathbf{y}} = \mathbf{A}\mathbf{C}_{\mathbf{x}}\mathbf{A}^T. \tag{11.4-8}$$

Более того, учитывая способ построения матрицы **A**, ковариационная матрица $\mathbf{C}_{\mathbf{y}}$ является диагональной, и элементы, находящиеся на главной диагонали, суть собственные значения матрицы $\mathbf{C}_{\mathbf{x}}$, т. е.

$$\mathbf{C}_{\mathbf{y}} = \begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_n \end{bmatrix}. \tag{11.4-9}$$

Все элементы этой ковариационной матрицы, расположенные вне главной диагонали, равны 0, поэтому компоненты векторов **y** являются некоррелированными.

Учитывая, что все λ_j являются собственными значениями матрицы $\mathbf{C}_{\mathbf{x}}$, а элементы любой диагональной матрицы, находящиеся на ее главной диагонали, — собственными значениями этой матрицы [Noble, Daniel, 1988], делаем вывод, что матрицы $\mathbf{C}_{\mathbf{x}}$ и $\mathbf{C}_{\mathbf{y}}$ имеют совпадающий набор собственных значений.

Другое важное свойство преобразования Хотеллинга связано с восстановлением **x** по **y**. Поскольку строки матрицы **A** представляют собой ортонормированные векторы, отсюда следует, что $\mathbf{A}^{-1} = \mathbf{A}^T$, и, значит, любой вектор **x** может быть восстановлен по соответствующему вектору **y** с помощью соотношения

$$\mathbf{x} = \mathbf{A}^T\mathbf{y} + \mathbf{m}_{\mathbf{x}}. \tag{11.4-10}$$

Предположим, однако, что вместо использования всех собственных векторов $\mathbf{C}_{\mathbf{x}}$ строится матрица преобразования $\mathbf{A}_k$, состоящая лишь из k собственных векторов, которым отвечают k наибольших собственных значений; т. е. матрица $\mathbf{A}_k$ имеет размеры $k \times n$. Тогда векторы **y** будут иметь размерность k, и восстановление по формуле (11.4-10) перестанет быть точным (подобно тому, как это было в рассмотренной в разделе 11.2.3 процедуре описания границы с помощью части коэффициентов Фурье).

С использованием матрицы $\mathbf{A}_k$ восстанавливаться будет вектор

$$\hat{\mathbf{x}} = \mathbf{A}_k^T\mathbf{y} + \mathbf{m}_{\mathbf{x}}. \tag{11.4-11}$$

Можно показать, что средний квадрат ошибки между **x** и $\hat{\mathbf{x}}$ задается выражением

$$e_{\text{ms}} = \sum_{j=1}^{n}\lambda_j - \sum_{j=1}^{k}\lambda_j = \sum_{j=k+1}^{n}\lambda_j. \tag{11.4-12}$$

Первое равенство в (11.4-12) указывает, что ошибка равна 0, если $k = n$ (т. е. когда в преобразовании используются все собственные векторы ковариационной матрицы). Коль скоро значения λ_j в нашем построении монотонно убывают (при увеличении j), из соотношения (11.4-12) также видно, что при заданном k ошибку можно минимизировать, выбирая k собственных векторов, которым соответствуют наибольшие собственные значения. Таким образом, преобразование Хотеллинга оптимально в том смысле, что оно минимизирует средний квадрат ошибки между векторами $\mathbf{x}$ и их приближениями $\hat{\mathbf{x}}$. Учитывая лежащий в основе преобразования Хотеллинга принцип использования собственных векторов, отвечающих наибольшим собственным значениям, его также называют приведением *к главным компонентам*.

Пример 11.15. Описание изображений с помощью главных компонент.

■ На рис. 11.38 приведены шесть компонент многозонального спутникового изображения, которые соответствуют шести диапазонам спектра: видимому синему (450—520 нм), видимому зеленому (520—600 нм), видимому красному (630—690 нм), ближнему инфракрасному (760—900 нм), среднему инфракрасному (1550—1750 нм) и тепловому инфракрасному (10400—12500 нм). Целью этого примера является иллюстрация применения главных компонент для описания изображения.

а *б* *в*
г *д* *е*

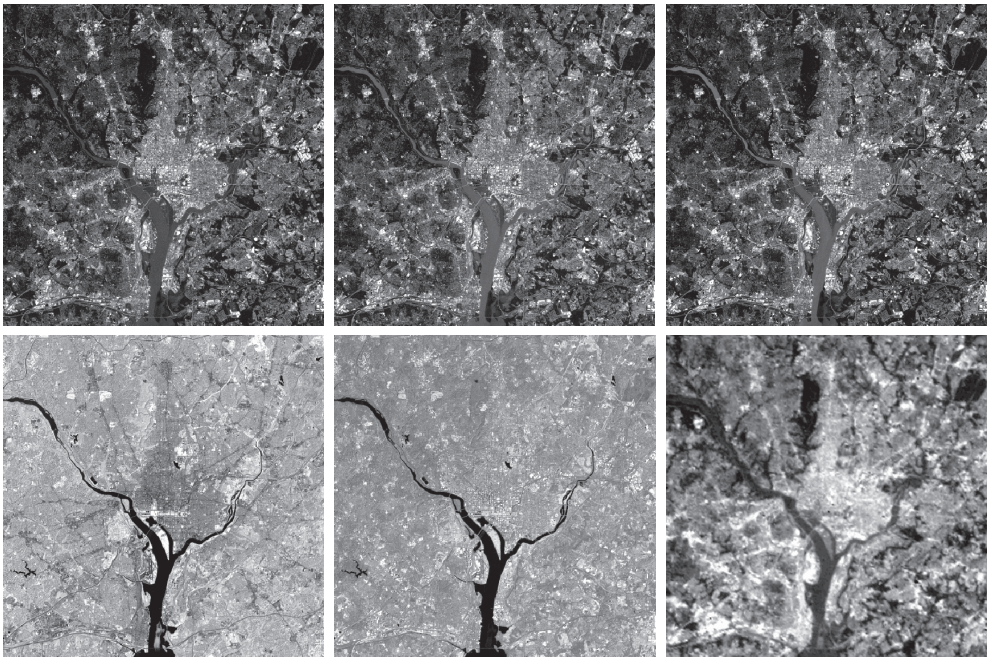


Рис. 11.38. Компоненты мультиспектрального изображения в (*а*) видимом синем, (*б*) видимом зеленом, (*в*) видимом красном, (*г*) ближнем инфракрасном, (*д*) среднем инфракрасном и (*е*) тепловом инфракрасном диапазонах. (Изображения предоставлены агентством NASA)

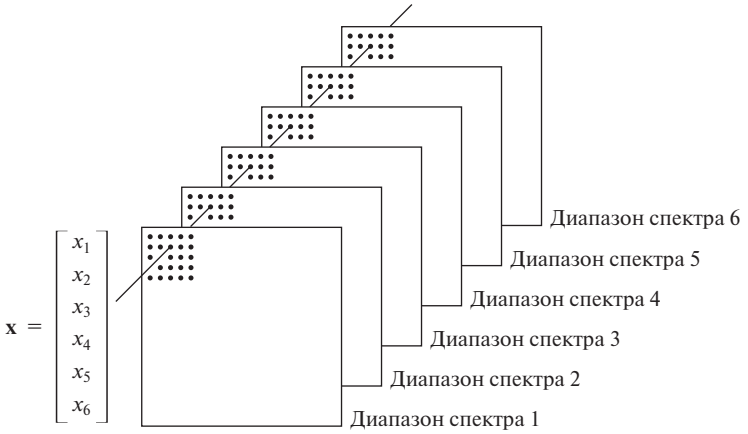


Рис. 11.39. Построение вектора из соответственных пикселей шести изображений

Рассматривая эти изображения в соответствии с рис. 11.39, из каждого набора их соответственных пикселей можно построить 6-мерный вектор $\mathbf{x} = (x_1, x_2, ..., x_6)^T$, как описывалось выше. В данном примере изображения имели размеры 564×564, так что выборка состояла из $(564)^2 = 318096$ векторов, по которым рассчитывались вектор математического ожидания, ковариационная матрица и соответствующие собственные значения и собственные векторы. Затем эти собственные векторы использовались как строки матрицы **A** и с помощью (11.4-6) был получен набор векторов **y**. Аналогично с помощью (11.4-8) была получена матрица $\mathbf{C}_y$. В табл. 11.6 указаны вычисленные собственные значения этой матрицы. Обратите внимание на явное преобладание двух первых собственных значений.

Набор изображений главных компонент был построен с помощью найденных векторов **y** непосредственным обращением схемы на рис. 11.39; полученные результаты представлены на рис. 11.40. Рис. 11.40(*а*) сформирован из первой компоненты 318096 векторов **y**, рис. 11.40(*б*) — из второй компоненты тех же векторов и т. д., так что полученные изображения по размерам совпадают с исходными на рис. 11.38. Самым очевидным свойством изображений главных компонент является то, что значительная доля контрастных деталей содержится в первых двух изображениях, а затем она быстро падает. Причина становится понятной, если взглянуть на собственные значения. Как показывает табл. 11.6, первые два собственных значения намного больше остальных. Поскольку собственные значения есть дисперсии элементов векторов **y**, а дисперсия является мерой яркостного контраста, естественно ожидать, что изображения, построенные из элементов векторов, отвечающих наибольшим собственным значениям, обладают и бóльшим контрастом. Фактически первые два изображения на рис. 11.40 отвечают приблизительно за 89 % от общей дисперсии[18]. Остальные четыре изображения демонстрируют меньший контраст деталей, поскольку они ответственны лишь за оставшиеся 11 % дисперсии.

[18] Для сопоставления амплитуд полученных компонент было бы правильнее сравнивать среднеквадратические отклонения. Такая оценка показывает, что первые две компоненты составляют вместе лишь 70 % суммы сигналов, что хотя и снижает заявленную точность, но не меняет общего вывода. — *Прим. перев.*

а *б* *в*
г *д* *е*

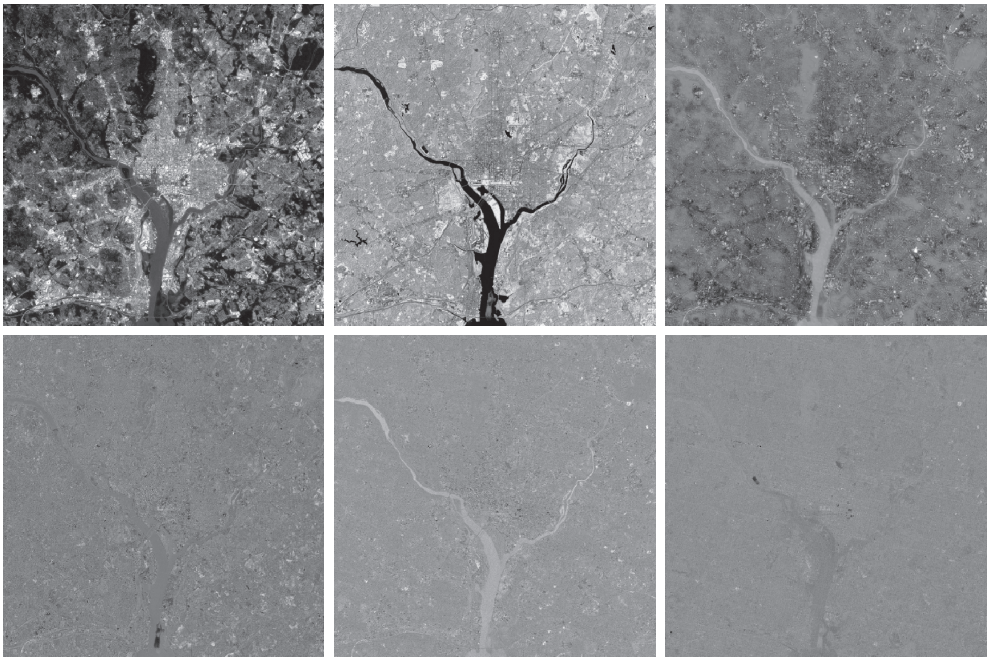


Рис. 11.40. Шесть изображений главных компонент, построенных из векторов, которые были получены с помощью формулы (11.4-6). Векторы превращались в изображения непосредственным обращением схемы на рис. 11.39

В соответствии с соотношениями (11.4-11) и (11.4-12), если использовать все собственные векторы в матрице **A**, можно восстановить исходные изображения (векторы) по изображениям (векторам) главных компонент с нулевой ошибкой между исходными и восстановленными изображениями, т. е. идентично. Однако нет смысла хранить и/или передавать изображения главных компонент и матрицу преобразования с целью последующего точного восстановления исходных изображений, поскольку это не дает никакого выигрыша. Теперь предположим, что сохраняются и/или передаются только изображения первых двух главных компонент, в которых содержится большинство деталей контраста. Это дает существенную экономию объема данных (матрица **A** имеет в этом случае размеры 2×6, так что ее вкладом можно пренебречь).

Говоря об изображении, мы зачастую обозначаем его словом «вектор», поскольку в данном контексте между ними имеется взаимно однозначное соответствие.

Таблица 11.6. Собственные значения ковариационной матрицы, полученной для набора изображений на рис. 11.38

λ_1	λ_2	λ_3	λ_4	λ_5	λ_6
10344	2966	1401	203	94	31

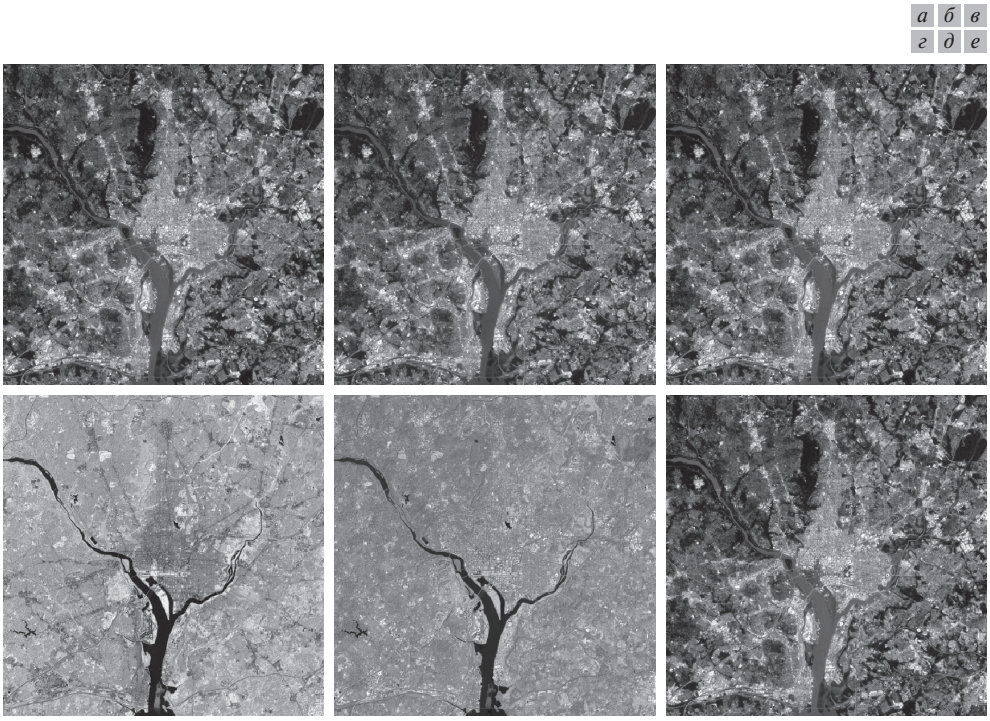


Рис. 11.41. Результат восстановления компонент мультиспектрального изображения только по двум изображениям главных компонент, которым соответствуют наибольшие собственные значения (дисперсии). Сравните с исходными изображениями на рис. 11.38

На рис. 11.41 представлены результаты восстановления всех шести компонент мультиспектрального изображения по изображениям лишь двух главных компонент, соответствующих наибольшим собственным значениям. Первые пять изображений по своему виду весьма близки к оригиналам на рис. 11.38, однако шестое заметно отличается. Причина этого в том, что оригинал шестого изображения фактически является размытым, а использованные при восстановлении первые две главные компоненты — резкие изображения, почему «детали» размывания и теряются. На рис. 11.42 показаны разности между исходными и восстановленными изображениями (чтобы сделать различия более заметными, разностные изображения подвергнуты градационной коррекции; в противном случае первые пять изображений выглядели бы совершенно черными). Как и следовало ожидать, шестое разностное изображение демонстрирует наибольшие отличия. ■

Пример 11.16. Использование главных компонент для нормализации описания безотносительно к сдвигу, масштабированию и повороту.

■ Как уже отмечалось в данной главе, представление и описание объектов должны быть максимально независимы от изменений размеров, сдвига и поворота. Использование метода главных компонент представляет собой удобный способ нормализации границ и/или областей в отношении изменений трех этих пара-

а *б* *в*
г *д* *е*

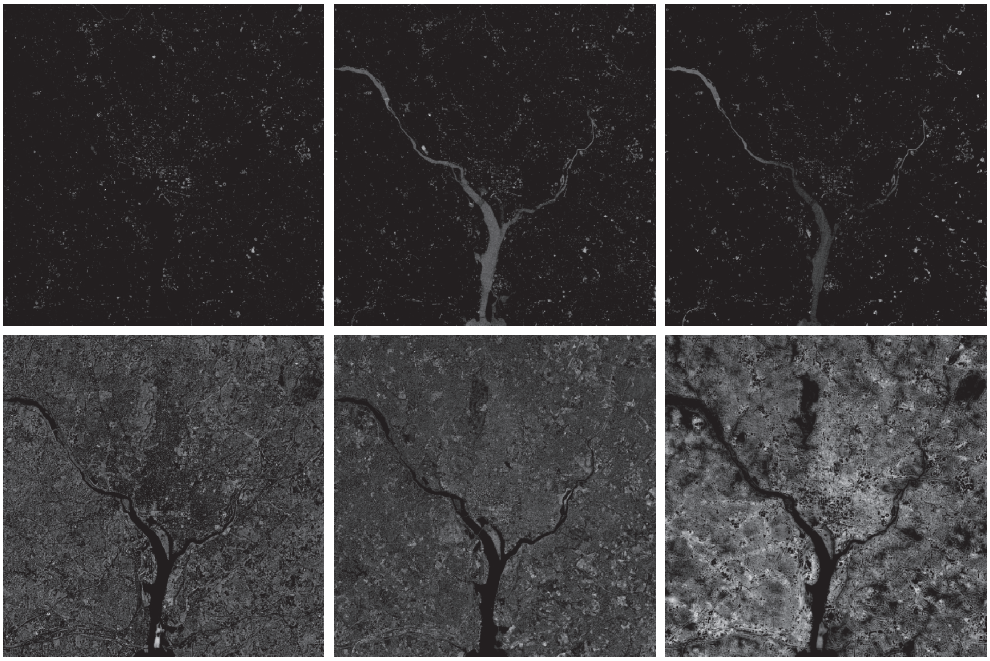


Рис. 11.42. Разности между восстановленными и исходными компонентами мультиспектрального изображения (для облегчения визуального анализа контраст разностных изображений усилен до полного диапазона яркостей [0, 255])

метров. Рассмотрим объект, изображенный на рис. 11.43, и предположим, что его размеры, местоположение и ориентация могут произвольно меняться. Каждую точку области (или ее границы) можно рассматривать как двухкомпонентный вектор $\mathbf{x} = (x_1, x_2)^T$, где x_1 и x_2 — значения координат этой точки по осям x_1 и x_2. Совокупность всех точек области или границы образует выборку двумерных векторов, которая используется для вычисления вектора математического ожидания $\mathbf{m}_x$ и ковариационной матрицы $\mathbf{C}_x$, как это делалось выше. Один собственный вектор $\mathbf{C}_x$ совпадает с направлением максимальной дисперсии (разброса данных) выборки, а другой ему перпендикулярен, как показано на рис. 11.43(*б*). Применительно к текущему обсуждению преобразование к главным компонентам согласно (11.4-6) делает две вещи: (1) переносит начало координат в центр тяжести (центроид) выборки, поскольку из каждого $\mathbf{x}$ вычитается $\mathbf{m}_x$; и (2) строит векторы $\mathbf{y}$ путем поворота исходных векторов $\mathbf{x}$, раскладывая данные по направлениям собственных векторов. Определяя новую систему координат (y_1, y_2) так, что направление оси y_1 совпадает с первым собственным вектором, а оси y_2 — со вторым, получим геометрическую схему, изображенную на рис. 11.43(*в*), в которой преобладающие направления данных совпадают с координатными осями. Независимо от сдвига и поворота объекта будет получен один и тот же результат, при условии что все точки области или границы претерпевают одинаковые изменения. Для нормализации результата относительно изменений

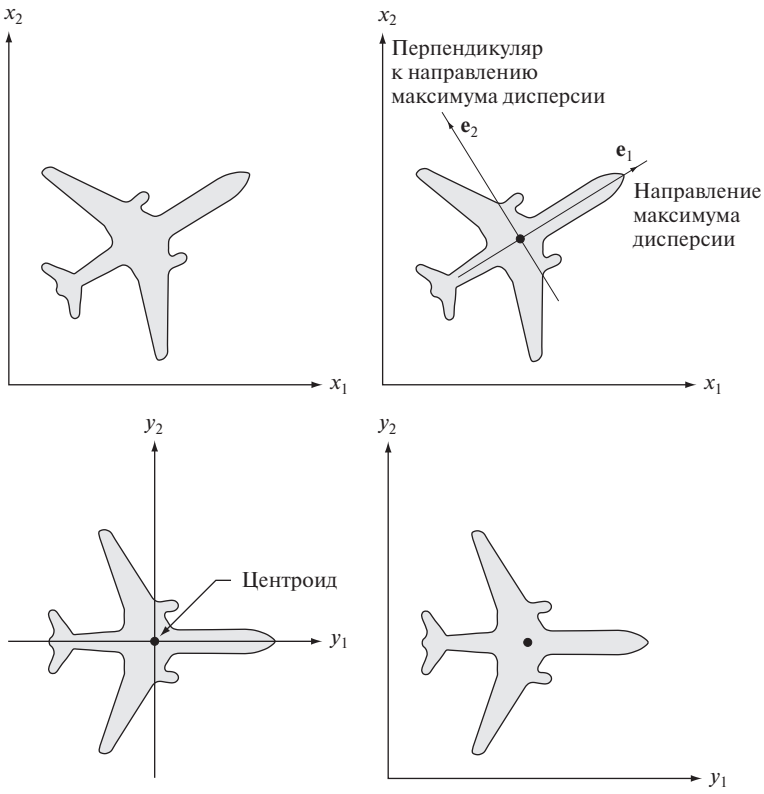


Рис. 11.43. (*а*) Изображение объекта. (*б*) Объект с наложенными собственными векторами его ковариационной матрицы. (*в*) Объект, преобразованный согласно уравнению (11.4-6). (*г*) Сдвиг объекта для обеспечения положительных значений всех координат

размеров (масштаба) исходного объекта достаточно в преобразованных данных разделить координаты на соответствующие собственные значения.

> В зависимости от ориентации объекта, система координат (y_1, y_2) может иметь направления осей, противоположные указанным на рис. 11.43(*в*). Например, если бы нос самолета был ориентирован в обратную сторону, вычисленные собственные векторы указывали бы влево и вниз.

Заметим, что в системе координат (y_1, y_2) на рис. 11.43(*в*) точки могут иметь как положительные, так и отрицательные координаты. Чтобы координаты всегда были неотрицательными, из всех векторов **y** просто вычитается вектор $(y_{1\min}, y_{2\min})^T$. Чтобы сдвинуть полученные точки в строго положительную область, как показано на рис. 11.43(*г*), к ним прибавляется вектор $(a, b)^T$, компоненты которого больше 0.

Хотя проведенное выше обсуждение в принципе является несложным, непосредственная техника выполнения операций часто приводит к путанице. Поэтому завершим этот пример простой иллюстрацией вычислений вручную. На рис. 11.44(*а*) показаны четыре точки с координатами (1, 1), (2, 4), (4, 2)

и (5, 5). Для такой выборки вектор математического ожидания, ковариационная матрица и нормированные (до единичной длины) собственные векторы будут следующими:

$$\mathbf{m}_x = \begin{bmatrix} 3 \\ 3 \end{bmatrix}, \quad \mathbf{C}_x = \begin{bmatrix} 3{,}333 & 2{,}000 \\ 2{,}000 & 3{,}333 \end{bmatrix}$$

и

$$\mathbf{e}_1 = \begin{bmatrix} 0{,}707 \\ 0{,}707 \end{bmatrix}, \quad \mathbf{e}_2 = \begin{bmatrix} -0{,}707 \\ 0{,}707 \end{bmatrix}.$$

Соответствующие собственные значения равны $\lambda_1 = 5{,}333$ и $\lambda_2 = 1{,}333$. На рис. 11.44(*б*) собственные векторы наложены на исходные данные. В соответствии с (11.4-6) преобразованные координаты точек (векторы **y**) будут (–2,828, 0), (0, 1,414), (0, –1,414) и (2,828, 0). Эти точки изображены на рис. 11.44(*в*); они располагаются на осях и имеют дробные координаты. При работе с изображениями координаты обычно целочисленные, поэтому их необходимо округлить до ближайшего целого. На рис. 11.44(*г*) показаны точки после округления и сдвига в положительную область, так что все значения координат положительные и целые, как и на исходном изображении. ■

а б
в г

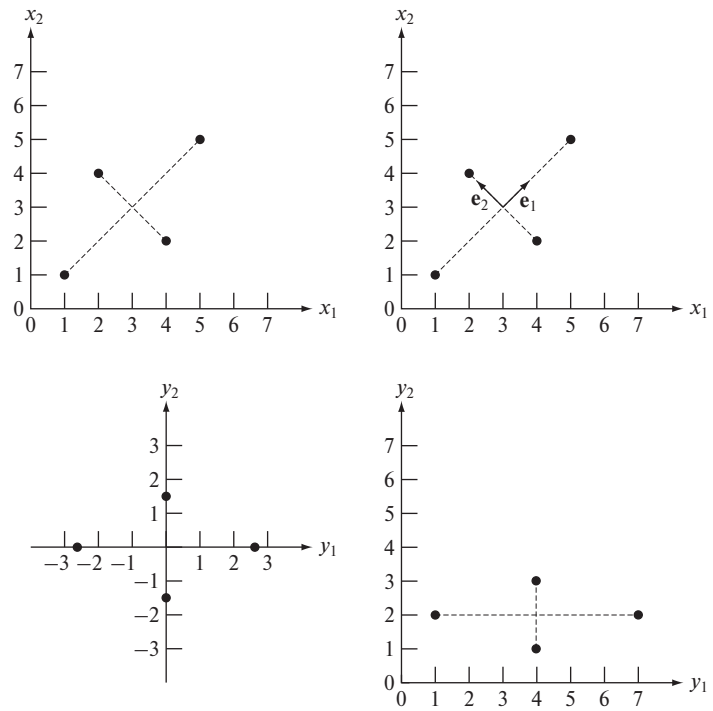


Рис. 11.44. Пример для ручного счета. (*а*) Исходные точки. (*б*) Собственные векторы ковариационной матрицы совокупности точек (*а*). (*в*) Точки, преобразованные согласно уравнению (11.4-6). (*г*) Точки после округления значений и сдвига для обеспечения положительности всех координат. Пунктирные линии добавлены для удобства; они не являются частью данных

11.5. Реляционные дескрипторы

В разделе 11.3.3 мы рассматривали принцип использования правил подстановки при построении описания текстуры. В данном разделе эта идея получит развитие в связи с реляционными дескрипторами. Они с равным успехом могут применяться и для границ, и для областей; главная цель таких дескрипторов — зафиксировать в форме правил подстановки элементарные конфигурации, которые повторяются на границе или внутри области.

Рассмотрим простую структуру в виде лестницы, показанную на рис. 11.45(*а*). Предположим, что в результате сегментации на изображении была выделена такая структура, и теперь мы хотим описать ее некоторым формальным способом. Определив два *непроизводных элемента*[19] *a* и *b*, как показано на рисунке, мы можем закодировать фигуру на рис. 11.45(*а*) в форме, показанной на рис. 11.45(*б*). Самым очевидным свойством закодированной структуры является чередование элементов *a* и *b*. Следовательно, простой способ символьного описания состоит в построении формальной рекурсивной зависимости, в которой участвовали бы эти базисные элементы. В частности, это можно сделать с помощью следующих *правил подстановки*:

(1) $S \to aA$,
(2) $A \to bS$,
(3) $A \to b$,

где S и A — переменные, а элементы a и b — константы, соответствующие введенным непроизводным элементам. Правило 1 указывает, что так называемый *начальный символ S* разрешается заменять на непроизводный элемент *a* со следующей за ним переменной *A*. Эта переменная, в свою очередь, в соответствии с правилом 2 может заменяться на *b* или (правило 3) сочетание *b* и *S*. Первый вариант позволяет снова применить правило 1 и повторить ту же процедуру. При замене *A* на *b* по правилу 3 процедура заканчивается, поскольку в выражении больше не остается переменных. Рис. 11.46 демонстрирует несколько примеров порождения с помощью указанных правил, где под каждой из фигур цифрами указана очередность применения правил 1, 2 и 3. Связь между *a* и *b* во всех случаях сохраняется, поскольку по этим правилам вслед за непроизводным элементом *a* всегда должен появиться *b*. Примечательно, что с помощью этих трех простых правил подстановки может порождаться (и значит, описываться) бесконечно много «похожих» структур.

Поскольку строки символов являются одномерными структурами, применение их для описания изображений требует построения подходящего метода, позволяющего свести двумерные пространственные отношения к одномерной форме. Большинство применений строк символов к описанию изображения базируется на идее выделения из интересующего объекта линии, составленной из соединенных друг с другом отрезков. Один из таких подходов основан на прослеживании контура объекта с кодированием результата отрезками заданного направления и/или длины. Рис. 11.47 иллюстрирует такую процедуру.

Другой, в чем-то более общий подход состоит в том, чтобы описывать фрагменты изображения (например однородные области малого размера) направ-

[19] В оригинале — *primitive element. — Прим. перев.*

а *б*

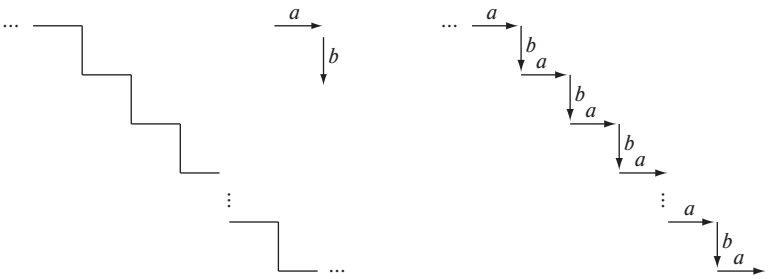


Рис. 11.45. (*а*) Простая ступенчатая структура. (*б*) Структура в закодированном виде

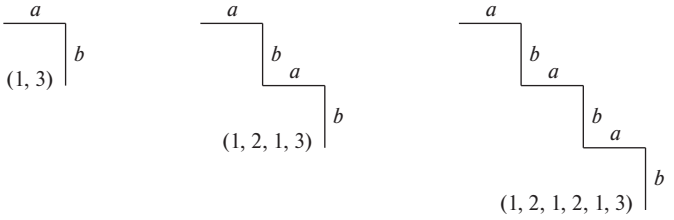


Рис. 11.46. Примеры порождения для системы правил $S \to aA$, $A \to bS$ и $A \to b$

ленными отрезками (см. рис. 11.48(*а*)), которые могут соединяться не только как начало одного с концом другого, но и другими способами, как показано на рис. 11.48(*б*), где представлены некоторые типовые операции, которые можно определить для абстрактных непроизводных элементов. На рис. 11.48(*в*) предлагается конкретный набор таких элементов, состоящий из отрезков четырех направлений, а рис. 11.48(*г*) демонстрирует последовательные шаги построения определенной фигуры, где обозначение (~*d*) используется для указания на непроизводный элемент *d* с противоположным направлением. Заметим, что у каждой такой составной структуры имеется одно начало и один конец. Искомым результатом является последняя строка, которая описывает законченную структуру.

Описания в виде строк символов лучше всего подходят для приложений, в которых способы соединения непроизводных элементов могут быть выраже-

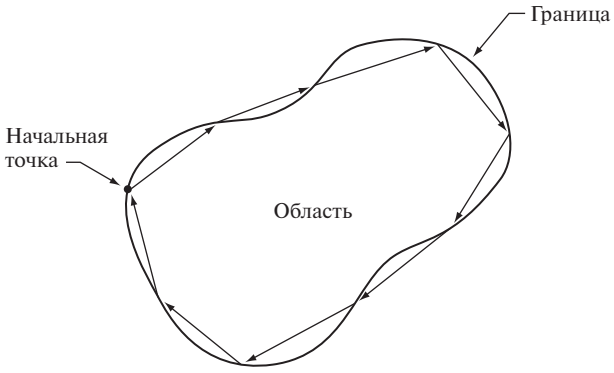


Рис. 11.47. Кодирование границы области с помощью направленных отрезков

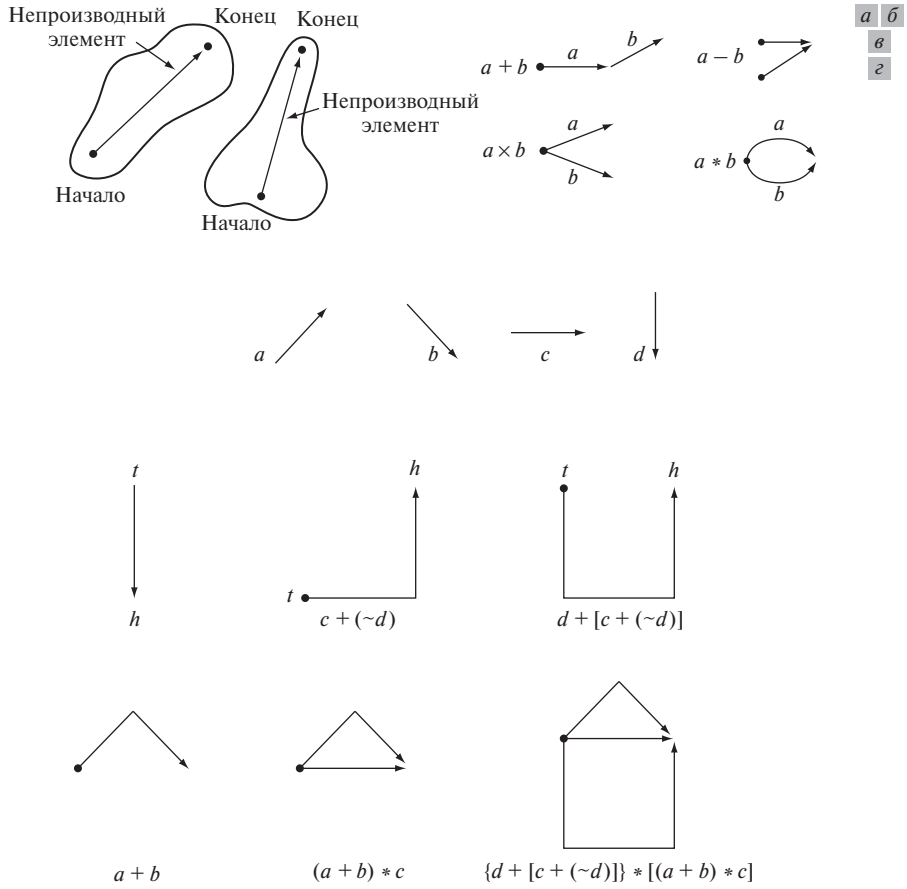


Рис. 11.48. (*а*) Абстрактные непроизводные элементы. (*б*) Операции над элементами. (*в*) Конкретный набор непроизводных элементов. (*г*) Шаги построения некоторой структуры

ны в терминах «конец с началом» или иным непрерывным способом. Иногда области, сходные по текстурным признакам или другим дескрипторам, оказываются несмежными, и для описания таких ситуаций требуются особые способы. Здесь одним из наиболее продуктивных подходов является использование деревьев в качестве дескрипторов.

Дерево T есть конечное множество, состоящее из одной или более вершин, такое, что: (а) существует единственная вершина $, принимаемая за *корень* дерева, и (б) остальные вершины разделяются на *m* непересекающихся мно-

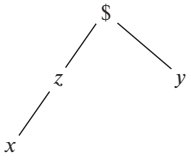


Рис. 11.49. Простое дерево с корнем $ и *концевыми вершинами xy*

а б

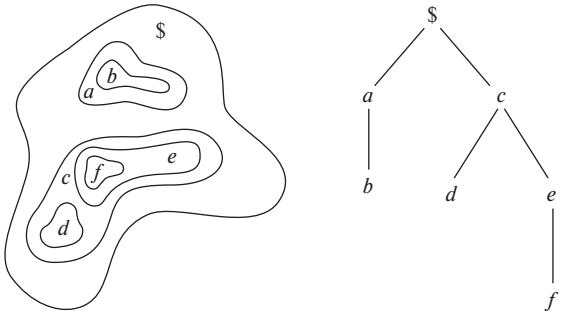


Рис. 11.50. (*а*) Простая область, состоящая из вложенных подобластей. (*б*) Представление в виде дерева, полученное с помощью отношения «находиться внутри»

жеств, каждое из которых в свою очередь является деревом, называемым *поддеревом* дерева *T*.

Концевые вершины графа — это набор вершин внизу дерева (которые также называются *листьями*), взятых в фиксированном порядке (слева направо). Например, изображенное на рис. 11.49 дерево имеет корень $ и *концевые вершины xy*.

Вообще говоря, есть два вида существенной информации об элементах (вершинах) дерева: (1) информация о само́й вершине, хранящаяся в виде некоторых параметров, описывающих эту вершину, и (2) информация о связях вершины с ее соседями, хранимая в виде набора указателей на эти соседние вершины. Применительно к описанию изображений первый вид информации определяет фрагмент структуры изображения (например часть области или границы), тогда как второй вид информации определяет физическую связь этого фрагмента структуры с другими фрагментами (т. е. отношение между ними). Например, объект на рис. 11.50(*а*) можно представить в виде дерева с помощью отношения «находиться внутри». Тогда, если за корень дерева принять область, обозначенную на рисунке $, видно, что на первом уровне сложности внутри $ находятся области *a* и *c*, что дает две ветви, исходящие от корня, как показано на рис. 11.50(*б*). На следующем уровне появляется область *b*, находящаяся внутри *a*, а также области *d* и *e* внутри *c*. Наконец, областью *f*, находящейся внутри *e*, завершается построение дерева.

Заключение

Представление и описание объектов или областей, выделенных при сегментации изображения, составляют начальные этапы работы большинства автоматизированных процессов, в которых участвуют цифровые изображения. Такие описания, например, являются входной информацией для методов распознавания объектов, обсуждаемых в следующей главе. Как показывает обсуждавшийся выше набор методов описания, выбор того или иного подхода определяется рассматриваемой задачей. Цель состоит в том, чтобы выбрать дескрипторы, способные «ухватить» существенные различия между объектами или классами

объектов и которые в то же время были бы максимально независимыми от изменений местоположения объекта, его размеров и ориентации.

Ссылки и литература для дальнейшего изучения

Описанный в разделе 11.1.1 алгоритм прослеживания границы был впервые предложен Муром [Moore,1968]. Рассмотренное в разделе 11.1.2 представление границ области с помощью цепного кода впервые было предложено Фрименом [Freeman, 1961, 1974]. Современное состояние исследований по цепным кодам представлено в работах Брибиеска [Bribiesca, 1999], где цепные коды распространены также и на трехмерный случай [Bribiesca, 2000]. Подробное обсуждение и алгоритм нахождения аппроксимирующего границу многоугольника с минимальным периметром (раздел 11.1.3) проводится в работах [Klette, Rosenfeld, 2004]; см. также [Sloboda et al., 1998] и [Coeurjolly, Klette, 2004]. Среди интересного дополнительного материала к разделу 11.1.4 отметим построение инвариантного приближения ломаной линией [Voss, Suesse, 1997], методы оценки качества и производительности алгоритмов кусочно-линейной аппроксимации [Rosin, 1997], типовые методы реализации [Huang, Sun, 1999] и исследование скорости вычислений [Davis, 1999].

Рассмотрение сигнатур (раздел 11.1.5) проводится в книге [Ballard, Brown, 1982] и статье [Gupta, Srinath, 1988]. Фундаментальные построения для нахождения выпуклой оболочки и дефицита выпуклости (раздел 11.1.6) проводятся в [Preparata, Shamos, 1985]. См. также статью [Liu-Yu, Antipolis, 1993]. В работе [Katzir et al., 1994] рассматривается обнаружение частично смыкающихся кривых. В статье [Zimmer et al., 1997] рассматривается улучшенный алгоритм построения выпуклой оболочки, а статья [Latecki, Lakämper, 1999] обсуждает правило выпуклости применительно к декомпозиции фигуры.

Алгоритм скелетонизации, рассмотренный в разделе 11.1.7, основан на работе [Zhang, Suen, 1984]. Ряд полезных дополнительных замечаний о свойствах и реализации этого алгоритма приводятся в статье [Lu, Wang, 1986]. Статья [Jang, Chin, 1990] проводит интересную связь между предметом рассмотрения раздела 11.1.7 и морфологическим принципом утончения, представленным в разделе 9.5.5. По поводу методов утончения в присутствии шума см. статьи [Shi, Wong, 1994] и [Chen, Yu, 1996]. В работе [Shaked, Bruckstein, 1998] описывается алгоритм отсечения, полезный для удаления паразитных отростков остова области. Быстрый алгоритм вычисления преобразования срединной оси рассматривается в работах [Sahni, Jenq, 1992] и [Ferreira, Ubéda, 1999]. Представляет интерес обзорная статья [Loncaric, 1998], касающаяся многих методов, обсуждаемых в разделе 11.1.

В статье [Freeman, Shapira, 1975] дается алгоритм нахождения базового прямоугольника для замкнутой кривой, представленной с помощью цепного кода (раздел 11.2.1). Обсуждение номеров фигур в разделе 11.2.2 базируется на работах [Bribiesca, Guzman, 1980] и [Bribiesca, 1981]. В качестве дополнительного материала по Фурье-дескрипторам (раздел 11.2.3) рекомендуются ранние работы [Zahn, Roskies, 1972] и [Persoon, Fu, 1977]. См. также [Aguado et al., 1998] и [Sonka et al., 1999]. В статье [Reddy, Chatterji, 1996] обсуждается интересный подход,

в котором с помощью БПФ достигается инвариантность к параллельному переносу, повороту и изменению масштаба. Материал раздела 11.2.4 базируется на элементарной теории вероятности (см., например, [Peebles, 1993] и [Popoulis, 1991]).

В качестве дополнительной литературы по разделу 11.3.2 см. книги [Rosenfeld, Kak, 1982] и [Ballard, Brown, 1982]. Прекрасным введением в анализ текстуры (раздел 11.3.3) является книга [Haralick, Shapiro, 1992]. Ранний обзор по анализу текстуры приводится в статье [Wechsler, 1980]. Современные работы в этой области представлены статьями [Murino et al., 1998] и [Garcia, 1999], а также монографией [Shapiro, Stockman, 2001].

Подход с использованием инвариантов для моментов, изложенный в разделе 11.3.4, взят из статьи [Hu, 1962]. См. также [Bell, 1965]. Чтобы получить представление о диапазоне возможных приложений инвариантов моментов, см. работы [Hall, 1979] о сравнении изображений и [Cheung, Teoh, 1999] об описании симметрии с помощью моментов. Инварианты моментов для n-мерного случая обобщены в работе [Mamistvalov, 1998]. Относительно обобщения моментов произвольного порядка см. [Flusser, 2000].

Хотеллинг первым вывел и опубликовал метод преобразования дискретных переменных в некоррелированные коэффициенты, названный им *методом главных компонент* [Hotelling, 1933]. Эта статья демонстрирует глубокое понимание метода и заслуживает прочтения. Преобразование Хотеллинга было заново «открыто» в статье [Kramer, Mathews, 1956], а затем в [Huang, Schultheiss, 1963]. Главные компоненты все еще остаются одним из основных инструментов описания изображений в многочисленных прикладных задачах, примерами чего могут служить работы [Swets, Weng, 1996] и [Duda, Heart, Stork, 2001]. В качестве дополнительных ссылок по материалу раздела 11.5 укажем монографии [Gonzalez, Thomason, 1978] и [Fu, 1982]. См. также книгу [Sonka et al., 1999]. В качестве дополнительного материала по тематике данной главы, ориентированного на вопросы реализации, рекомендуются работы [Nixon, Aguado, 2002] и [Gonzalez, Woods, Eddins, 2004].

Задачи

11.1 ★ **(а)** Покажите, что переопределение начальной точки цепного кода, так чтобы получаемая последовательность цифр образовывала наименьшее целое число, делает код не зависящим от выбора начальной точки на границе.

(б) Найдите начальную точку, которая нормализует цепной код 10176722335422.

11.2 ★ **(а)** Покажите, что вычисление разности первого порядка цепного кода делает его инвариантным по отношению к повороту, как говорится в разделе 11.1.2.

(б) Вычислите разность первого порядка кода 0110233210332322111.

11.3 ★ **(а)** Покажите, что описанный в разделе 11.1.3 метод аппроксимации замкнутой ломаной линией, построенной с помощью резиновой ленты, приводит к многоугольнику с минимальным периметром.

(б) Покажите, что если каждая клетка соответствует центру пикселя, то максимально возможная ошибка аппроксимации в этой клетке составляет $d/\sqrt{2}$, где d — минимальное расстояние между соседними пикселями по горизонтали или по вертикали (т. е. шаг сетки дискретизации, использованной для получения цифрового изображения).

11.4 Объясните, как описанный в разделе 11.1.3 алгоритм построения ЛМД поведет себя в следующих ситуациях:

★ **(а)** Ступеньки шириной 1 пиксель и глубиной 1 пиксель.

★ **(б)** Ступеньки шириной 1 пиксель и глубиной n пикселей.

(в) Выступ шириной 1 пиксель и длиной 1 пиксель.

(г) Выступ шириной 1 пиксель и длиной n пикселей.

11.5 ★ **(а)** Опишите, как повлияет на полученный прямоугольник выбор нулевого порога для ошибки в методе слияния, рассмотренном в разделе 11.1.4.

(б) Как такой же порог повлияет на метод разбиения?

11.6 ★ **(а)** Постройте для границы квадратной формы график сигнатуры при использовании метода угла наклона касательной к контуру, описанного в разделе 11.1.5.

(б) Повторите то же самое для функции плотности угла наклона.

Полагайте, что стороны квадрата параллельны осям координат и в качестве опорной прямой используется ось x. Начните с угла, ближайшего к началу координат.

11.7 Найдите аналитические выражения для сигнатуры каждой из следующих границ и постройте их графики.

★ **(а)** Равносторонний треугольник.

(б) Квадрат.

(в) Окружность.

11.8 Нарисуйте срединные оси

★ **(а)** эллипса,

★ **(б)** правильного пятиугольника,

(в) прямоугольника,

(г) равностороннего треугольника.

11.9 Для каждого из показанных ниже рисунков:

★ **(а)** Опишите действия, выполняемые в точке p на первом шаге алгоритма скелетонизации, описанного в разделе 11.1.7.

(б) Повторите то же для второго шага алгоритма, предполагая, что $p = 0$ во всех случаях.

1	1	0
1	p	0
1	1	0

0	0	0
1	p	0
0	0	0

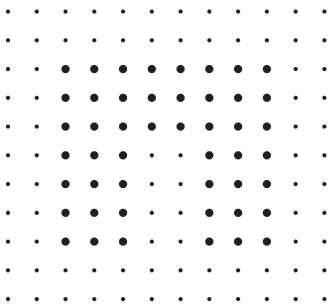
0	1	0
1	p	1
0	1	0

1	1	0
0	p	1
0	0	0

11.10 Применительно к алгоритму скелетонизации из раздела 11.1.7 как будет выглядеть показанная ниже фигура после выполнения:

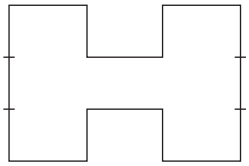
★ **(а)** Одного прохода первого шага алгоритма?

(б) Одного прохода второго шага (для результата, полученного на шаге 1, а не исходного изображения)?



11.11 ★ **(а)** Каков будет порядок номера фигуры для изображенной ниже фигуры?

(б) Получите сам номер фигуры.



11.12 Рассмотренная в разделе 11.2.3 процедура использования дескрипторов Фурье состоит в выражении координат точек контура комплексными числами, вычислении ДПФ для этих чисел и сохранении затем лишь части членов ДПФ в качестве описателей формы границы. При выполнении обратного ДПФ будет получено некоторое приближение исходного контура. Для какого класса форм контуров полученное ДПФ будет состоять из действительных чисел и как следует провести координатные оси на рис. 11.19, чтобы получить этот результат?

11.13 Докажите, что если для восстановления границы с помощью формулы (11.2-5) используются только два Фурье-дескриптора ($u = 0$ и $u = 1$), то результатом всегда будет окружность. (*Подсказка:* воспользуйтесь представлением окружности в параметрической форме на комплексной плоскости и запишите уравнение окружности в полярных координатах.)

★**11.14** Укажите наименьшее число дескрипторов — статистических моментов, необходимых, чтобы различать сигнатуры фигур, показанных на рис. 11.10.

11.15 Приведите пример двух границ различной формы, у которых дескрипторы математического ожидания и третьего момента были бы одинаковы, а вторые моменты отличались.

★**11.16** Предложите набор дескрипторов, с помощью которых можно было бы различать по форме символы 0, 1, 8, 6 и *Z*. (*Подсказка:* воспользуйтесь топологическими дескрипторами в соединении с выпуклой оболочкой.)

11.17 Рассмотрите двоичное изображение размерами 100×100, содержащее вертикальную черную полосу в столбцах 1—49 и белую — в столбцах 50—100.

(а) Постройте для этого изображения матрицу сочетаемости, используя оператор позиционирования «на один пиксель вправо».

★ **(б)** Осуществите нормировку этой матрицы так, чтобы ее элементы стали оценками вероятности, как объяснялось в разделе 11.3.1.

(с) Используйте матрицу, полученную на шаге (б), для вычисления шести дескрипторов, приведенных в табл. 11.3.

11.18 Рассмотрите изображение шахматного поля, состоящего из чередующихся белых и черных квадратов со стороной m. Укажите оператор позиционирования, который бы порождал матрицу сочетаемости яркостей диагонального вида.

11.19 Постройте матрицу сочетаемости яркостей для изображения 8×8, содержащего шахматное поле из чередующихся единиц и нулей, если оператор позиционирования Q определен следующим образом:

★ **(а)** «на один пиксель вправо»,

(б) «на два пикселя вправо».

Предполагайте, что левый верхний элемент изображения имеет значение 0.

11.20 Докажите справедливость равенств (11.4-7), (11.4-8) и (11.4-9).

★**11.21** В примере 11.13 отмечалось, что достоверное восстановление шести исходных изображений может быть выполнено с помощью всего двух изображений главных компонент, которым отвечают наибольшие собственные значения. Какая при этом будет внесена среднеквадратическая ошибка? Выразите свой ответ в процентах от максимально возможного значения ошибки.

11.22 **П**редположите, что для набора изображений размерами 32×32 ковариационная матрица в (11.4-9) оказывается единичной матрицей. Какова будет среднеквадратическая ошибка при восстановлении исходных изображений согласно уравнению (11.4-11), если используется только половина от всех собственных векторов?

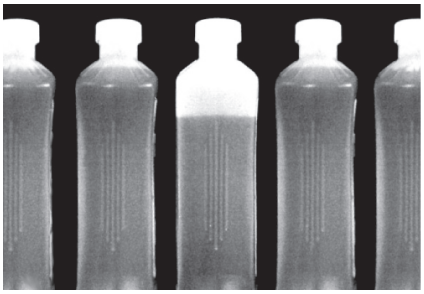
★**11.23** При соблюдении каких условий можно ожидать, что введенные в разделе 11.2.1 главные оси границы будут совпадать с ее собственными осями?

11.24 Предложите пространственное отношение и постройте соответствующее ему представление в виде дерева для шахматного поля из черных и белых квадратов. Считайте, что левый верхний элемент черный и ему соответствует корень дерева. В построенном дереве из каждой вершины должно исходить не более двух ветвей.

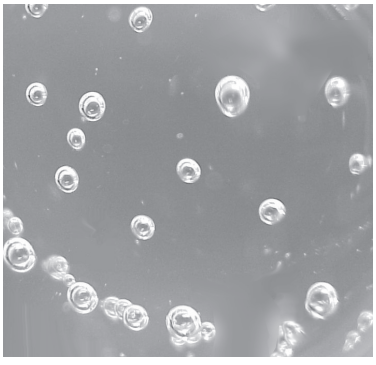
★**11.25** С Вами заключен контракт на проектирование системы обработки изображений для обнаружения дефектов внутри твердых пластмассовых дисков. Диски обследуются с помощью рентгеновской системы, дающей 8-битовые изображения с разрешением 1024×1024 элементов. При отсутствии дефектов изображение выглядит «чистым», характеризуясь средней яркостью 100 с дисперсией 200. Дефекты проявляются в виде округлых областей, внутри которых около 70 % пикселей имеют отклонение по яркости до 40 градаций от среднего значения 100. Диск считается дефектным, если такая область занимает площадь более 20×20 элементов. Предложите систему, основанную на анализе текстуры.

11.26 Компания, фасующая в пластмассовые бутылки различные промышленные химикаты, прослышала о Ваших успехах в решении задач анализа изображений и нанимает Вас для разработки метода обнаружения не до конца заполненных бутылок во время их движения на конвейерной линии (см. рисунок). Бутылка считается заполненной не до конца, если уровень жидкости ниже середины между началом сужения и низом горлышка. Хотя бутылки движутся,

система регистрации изображений оборудована лампой-вспышкой, что создает эффект неподвижного изображения, которое выглядит очень похоже на показанный пример. Основываясь на уже изученном материале, предложите решение для обнаружения дефектных бутылок. Четко сформулируйте все сделанные предположения, которые влияют на выбор предлагаемого решения.



11.27 Узнав о том, как Вы успешно справились с задачей обнаружения неполных бутылок, к Вам обращается компания, желающая автоматизировать подсчет пузырьков в определенных технологических процессах с целью контроля качества. Компания уже решила проблему регистрации изображений и может получать 8-битовые изображения с разрешением 800×800 элементов, подобных примеру на рисунке ниже. Изображение представляет площадь 8 см^2. Для каждого изображения компания хочет решить две следующие задачи: (1) определить долю площади, занимаемой пузырьками, по отношению ко всей площади изображения; и (2) подсчитать число отдельных пузырьков. Основываясь на уже изученном материале, предложите решение этих задач. В этом решении обязательно укажите минимальные размеры пузырька, который сможет обнаруживаться предлагаемой системой. Четко сформулируйте все сделанные предположения, которые влияют на выбор Вашего решения.



ГЛАВА 12

РАСПОЗНАВАНИЕ ОБЪЕКТОВ

Одно из самых интересных свойств мира — это то, что его можно рассматривать как составленный из образов. Образ — это по сути структура, которая характеризуется не столько внутренней сущностью составляющих ее элементов, сколько их расположением.

Норберт Винер

Введение

Завершим наше изучение сферы цифровой обработки изображений знакомством с методами распознавания объектов. Как отмечалось в разделе 1.1, мы включили в тематику цифровой обработки изображений распознавание *отдельных* областей изображения, которые в этой главе и будем называть *объектами* или *образами*.

Излагаемые в данной главе методы распознавания образов делятся на две основные категории: методы, основанные на теории принятия решений, и структурные методы. Первая категория имеет дело с образами, описанными с помощью количественных дескрипторов, таких как длина, площадь, текстура. Вторая категория методов ориентирована на образы, для описания которых лучше подходят качественные дескрипторы, например реляционные, обсуждавшиеся в разделе 11.5.

В распознавании образов центральную роль играет принцип «обучения» на выборке известных образов. Далее рассматриваются и иллюстрируются методы обучения, применимые как в структурном подходе, так и в теории принятия решений.

12.1. Образы и классы образов

Под *образом*[1] подразумевается некоторая *упорядоченная совокупность дескрипторов*, подобных рассмотренным в главе 11. В литературе по распознаванию

[1] В первом же абзаце главы авторы отождествили распознаваемые объекты (в нашем случае области изображения) и их образы (т. е. представления в виде совокупности дескрипторов). Такое отождествление часто встречается в литературе по распознаванию образов. Следуя этой укоренившейся традиции, мы будем в дальнейшем использовать термин «объект» наравне с понятием «образ», не делая между ними принципиального различия. Отметим также, что переход от объекта к его образу в общем случае необратим, и само изображение в дальнейшем содержательно не используется. — *Прим. перев.*

образов эти дескрипторы часто называют *признаками*. *Классом образов* (или просто *классом*) называется совокупность образов, обладающих некоторыми общими свойствами. Будем обозначать классы символами ω_1, ω_2, ..., ω_W, где W — число классов. Под машинным распознаванием образов понимаются методы, позволяющие относить образы к тем или иным классам — автоматически или с минимальным вмешательством человека.

В практических задачах получили распространение три формы упорядоченного представления признаков: в виде *векторов признаков* (для количественных дескрипторов), в виде *символьных строк*, а также в виде *деревьев* (строки и деревья применяются для структурных описаний). Образы, представленные векторами признаков, обозначаются жирными строчными буквами, например **x**, **y**, **z**, и имеют форму

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \tag{12.1-1}$$

где каждая из компонент x_i представляет i-й дескриптор, а n — общее число дескрипторов, связанных с данным образом. Образы представляются вектор-столбцами (т. е. матрицами порядка $n \times 1$) вида (12.1-1) или в эквивалентной форме $\mathbf{x} = (x_1, x_2, ..., x_n)^T$, где T — операция транспонирования. Мы уже использовали эту запись в разделе 11.4.

Краткий обзор векторов и матриц приведен на интернет-сайте книги.

Содержательное наполнение компонент вектора признаков **x** зависит от применяемого подхода к описанию самого физического объекта. Проиллюстрируем это на простом примере, который в то же время позволит ощутить историю этого научного направления, идущего от классификации результатов измерений. В ставшей классической статье Фишера [Fisher, 1936] сообщалось об использовании нового метода, получившего название *дискриминантный анализ* (мы рассмотрим его в разделе 12.2), для распознавания трех видов цветков ириса (*Iris setosa*, *Iris virginica* и *Iris versicolor*) по данным измерений длины и ширины их лепестков (рис. 12.1). В нашей сегодняшней терминологии каждый цветок *описывается* результатами двух измерений, что приводит к двумерному вектору признаков вида

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \tag{12.1-2}$$

где x_1 и x_2 соответствуют длине и ширине лепестка соответственно. В данном случае три класса, обозначенные ω_1, ω_2 и ω_3, отвечают разновидностям *setosa*, *virginica* и *versicolor* соответственно.

Поскольку для лепестков характерна изменчивость по длине и ширине, описывающие их векторы признаков варьируются не только от класса к классу, но также в пределах одного и того же класса. На рис. 12.1 приведены результаты измерения длины и ширины для нескольких экземпляров каждой из разновидностей ириса. После того, как выбран набор измеряемых величин (состоящий

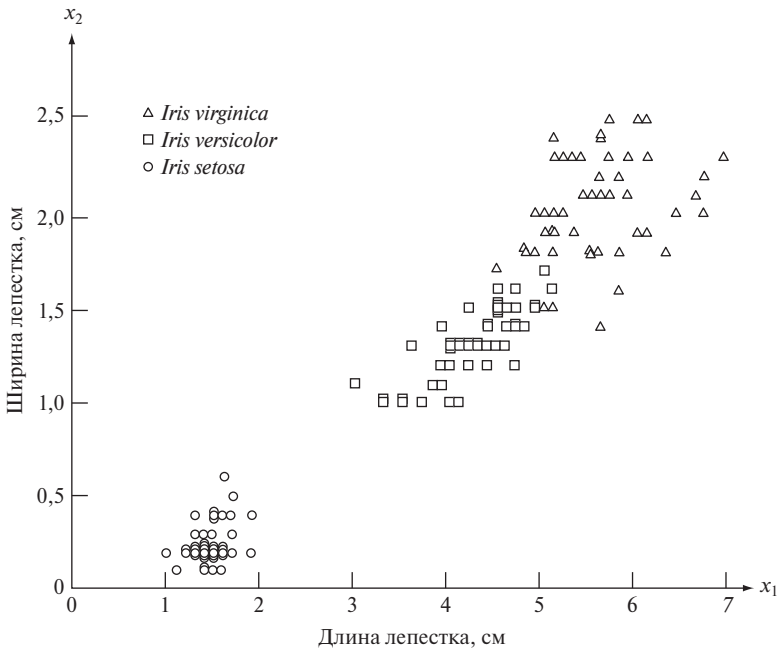


Рис. 12.1. Три вида цветков ириса, описанные двумя измерениями

из двух характеристик в рассматриваемом случае), компоненты вектора признаков становятся исчерпывающим описанием каждого физического образца. Таким образом, в этом примере каждый цветок представляется некоторой точкой в двумерном евклидовом пространстве. Заметим также, что в данном случае по измерениям длины и ширины лепестков класс *Iris setosa* оказывается достаточно обособленным от двух других классов; однако столь же успешного отделения друг от друга разновидностей *virginica* и *versicolor* не наблюдается. Это явление является иллюстрацией классической проблемы *выбора признаков*, которая проявляется в том, что в конкретном приложении *степень разделимости* классов сильно зависит от выбора дескрипторов. Этот вопрос будет подробнее рассматриваться в разделах 12.2 и 12.3.

Рис. 12.2 демонстрирует другой пример построения вектора признаков. Здесь интересующими объектами являются различные фигуры искаженной формы, типичный пример которых изображен на рис. 12.2(*а*). Если в качестве способа представления объекта выбрать его сигнатуру (см. раздел 11.1.5), то получим одномерные сигналы с формой, подобной показанной на рис. 12.2(*б*). Предположим, что каждую сигнатуру решено описывать как набор значений амплитуды, для чего с заданным шагом проводится ее дискретизация по углу θ, которая дает последовательность точек отсчета $\theta_1, \theta_2, ..., \theta_n$. Тогда векторы признаков можно формировать, присваивая их компонентам значения $x_1 = r(\theta_1)$, $x_2 = r(\theta_2), ..., x_n = r(\theta_n)$. Этим векторам соответствуют точки в n-мерном евклидовом пространстве, а класс образов можно представить в виде n-мерного «облака» в этом пространстве.

Вместо того чтобы использовать значения амплитуды сигнатуры непосредственно, мы могли бы вычислять, скажем, первые n статистических моментов

а б

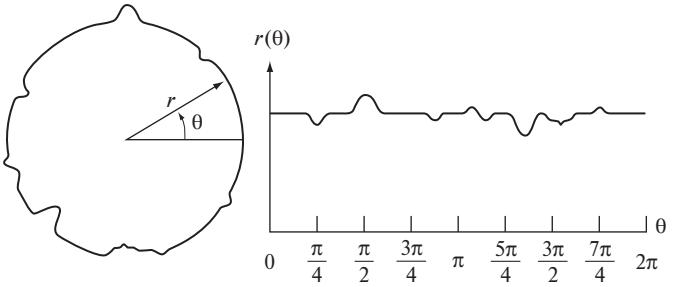


Рис. 12.2. Искаженный объект и соответствующая ему сигнатура

данной сигнатуры (см. раздел 11.2.4), а затем использовать эти дескрипторы в качестве компонент вектора признаков. Как уже ясно, векторы признаков можно построить и многими другими способами; некоторые из них будут представлены в этой главе. На данный момент важно отчетливо понять, что выбор дескрипторов, на которых базируются компоненты вектора признаков, оказывает глубочайшее влияние на конечные характеристики системы распознавания, использующей этот вектор признаков.

Описанная выше техника построения векторов признаков приводит к классам образов, характеризующихся количественной информацией. Однако в ряде практических областей характеристики образов лучше описываются структурными связями. Например, распознавание отпечатков пальцев основывается на взаимосвязях признаков отпечатка, называемых *локальными признаками*[2]. Эти признаки играют роль непроизводных элементов (примитивов), которые вместе с их относительными размерами и расположением описывают свойства линий отпечатка, такие как окончания, ветвления, слияния и разрывы. Подобные задачи распознавания, в которых принадлежность к классу определяется не только данными количественных измерений признаков, но и пространственными отношениями между признаками, обычно успешно решаются применением структурных методов. Такой подход был нами впервые рассмотрен в разделе 11.5; кратко напомним его здесь применительно к дескрипторам образов.

На рис. 12.3(*а*) представлен объект ступенчатой структуры. Такой объект, конечно, может быть дискретизован, а последовательность его отсчетов выражена в виде вектора признаков, аналогично тому, как это делалось для объекта на рис. 12.2. Однако при таком описании была бы потеряна базовая структура, образуемая повторением двух простых непроизводных элементов. Более содержательным способом описания было бы определение примитивов *a* и *b* и формирование образа в виде символьной строки $w = ...ababababab...$, как это показано на рис. 12.3(*б*). Такое описание отражает структуру данного конкретного класса объектов, требуя, чтобы связь осуществлялась по правилу «начало одного примитива с концом другого», причем с обязательным чередованием символов. Построенная подобным образом конструкция годится для ступенчатых структур произвольной длины, но исключает другие виды структур, которые могли бы быть построены из тех же непроизводных элементов *a* и *b* в других сочетаниях.

---

[2] В оригинале — *minutiae*; в русскоязычной литературе локальные признаки часто называют *минуциями*. — *Прим. перев.*

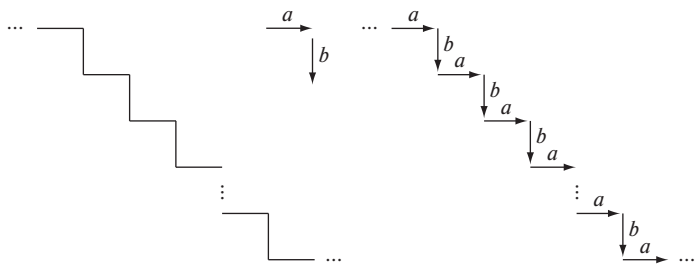


Рис. 12.3. (*а*) Ступенчатая структура. (*б*) Структура, закодированная с помощью непроизводных элементов *a* и *b*, в форме символьной строки *...ababab...*

Описания в форме символьных строк порождают адекватные образы для таких объектов, структура которых базируется на относительно несложных способах соединения примитивов; такая связь обычно соответствует формам границ фигур. Для многих прикладных задач применяется более мощный подход, основанный на описании с помощью деревьев, который рассматривался в разделе 11.5. По существу, большинство иерархически упорядоченных схем приводят к древовидным структурам. Например, на рис. 12.4 представлено спутниковое изображение плотно застроенной центральной части города и окружающих ее жилых пригородов. Обозначим символом $ всю область изображения. Показанное на рис. 12.5 представление в виде дерева, обращенного корнем вверх,

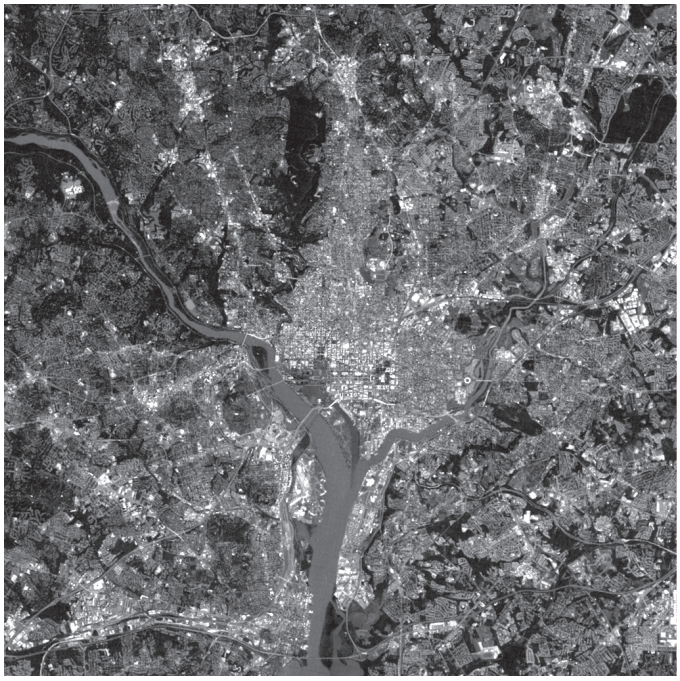


Рис. 12.4. Спутниковое изображение плотно застроенного центра г. Вашингтон (округ Колумбия) и окружающих жилых районов. (Изображение предоставлено агентством NASA)

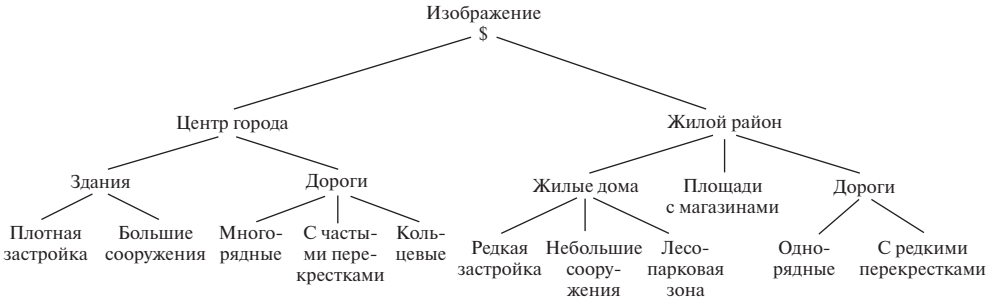


Рис. 12.5. Описание изображения на рис. 12.4 в виде дерева

было получено с помощью структурного отношения «состоять из». Итак, корень дерева представляет изображение целиком. Следующий уровень показывает, что изображение составлено из центральной части города и жилой зоны, которая, в свою очередь, состоит из жилых домов, шоссейных дорог и площадей с магазинами. Следующий уровень предлагает дальнейшую детализацию описания для жилых зданий и шоссейных дорог. Разбиение подобного вида может продолжаться вплоть до предела нашей способности разграничивать отличающиеся области на изображении.

В последующих разделах мы познакомимся с методами распознавания объектов, описанных рассмотренными выше способами.

12.2. Распознавание на основе методов теории принятия решений

Подход к задачам распознавания образов с позиций теории принятия решений основан на использовании *решающих* (или *дискриминантных*) *функций*. Пусть $\mathbf{x} = (x_1, x_2, ..., x_n)^T$ — n-мерный вектор признаков объекта, обсуждавшийся в разделе 12.1. Основная задача распознавания в теории принятия решений формулируется следующим образом. Предположим, что существует W классов образов $\omega_1, \omega_2, ..., \omega_W$. Требуется найти W дискриминантных функций $d_1(\mathbf{x}), d_2(\mathbf{x}), ..., d_W(\mathbf{x})$, таких, что если образ $\mathbf{x}$ принадлежит классу ω_i, то

$$d_i(\mathbf{x}) > d_j(\mathbf{x}) \quad j = 1,2,...,W;\ j \neq i\,. \tag{12.2-1}$$

Другими словами, неизвестный образ $\mathbf{x}$ относят к i-му классу, если при подстановке $\mathbf{x}$ во все дискриминантные функции наибольшее численное значение дает функция $d_i(\mathbf{x})$. В случае неоднозначности решение принимается произвольным образом.

Разделяющая поверхность между классами ω_i и ω_j задается множеством значений $\mathbf{x}$, для которых $d_i(\mathbf{x}) = d_j(\mathbf{x})$, или, что то же самое, такими $\mathbf{x}$, для которых

$$d_i(\mathbf{x}) - d_j(\mathbf{x}) = 0\,. \tag{12.2-2}$$

Общепринятая практика состоит в том, чтобы описывать разделяющую поверхность между двумя классами единой функцией $d_{ij}(\mathbf{x}) = d_i(\mathbf{x}) - d_j(\mathbf{x}) = 0$. Тогда $d_{ij}(\mathbf{x}) > 0$ для образов из класса ω_i и $d_{ij}(\mathbf{x}) < 0$ для образов из класса ω_j. Главная

цель этого раздела состоит в рассмотрении различных подходов, применяемых для отыскания дискриминантных функций, удовлетворяющих неравенству (12.2-1).

12.2.1. Сопоставление

В методах распознавания, основанных на сопоставлении, каждый класс представляется вектором признаков некоторого образа, являющегося *прототипом* этого класса. Неизвестный образ приписывается к тому классу, к прототипу которого он оказывается ближайшим в смысле заранее заданной метрики. Простейший подход состоит в использовании классификатора, основанного на минимальном расстоянии, который, как ясно из названия, вычисляет евклидовы расстояния между вектором признаков неизвестного объекта и вектором признаков каждого прототипа. Решение о принадлежности объекта к определенному классу принимается по наименьшему из таких расстояний. Далее также будет рассмотрен корреляционный подход, который формулируется непосредственно в терминах изображений и поэтому вполне нагляден.

Классификатор по минимуму расстояния

Предположим, что прототип каждого класса определяется как математическое ожидание векторов образов этого класса, оцениваемое формулой

$$\mathbf{m}_j = \frac{1}{N_j}\sum_{x\in\omega_j}\mathbf{x}_j \qquad j=1,2,...,W, \tag{12.2-3}$$

где N_j — число векторов признаков объектов класса ω_j; суммирование ведется по всем таким векторам. Как и прежде, W обозначает число классов. Как уже отмечалось выше, один из способов отнести неизвестный объект с вектором признаков $\mathbf{x}$ к какому-то классу состоит в выборе того класса, прототип которого окажется ближайшим. При использовании евклидова расстояния в качестве меры близости задача сводится к вычислению расстояний

$$D_j(\mathbf{x}) = \left\|\mathbf{x}-\mathbf{m}_j\right\| \qquad j=1,2,...,W, \tag{12.2-4}$$

где $\|\mathbf{a}\| = (\mathbf{a}^T\mathbf{a})^{1/2}$ — евклидова норма. После этого объект $\mathbf{x}$ относится к тому классу ω_i, для которого значение $D_i(\mathbf{x})$ оказывается наименьшим. Таким образом, в данной постановке минимальное расстояние до прототипа означает наилучшее совпадение. Нетрудно показать (задача 12.2), что выбор кратчайшего расстояния эквивалентен вычислению функций

$$d_j(\mathbf{x}) = \mathbf{x}^T\mathbf{m}_j - \frac{1}{2}\mathbf{m}_j^T\mathbf{m}_j \qquad j=1,2,...,W \tag{12.2-5}$$

и отнесению затем $\mathbf{x}$ к тому классу ω_i, для которого $d_i(\mathbf{x})$ принимает наибольшее численное значение. Такая формулировка согласуется с понятием дискриминантной функции, определяемой согласно соотношению (12.2-1).

Из соотношений (12.2-2) и (12.2-5) следует, что в случае классификатора по минимуму расстояния разделяющая поверхность между классами ω_i и ω_j задается уравнением

$$d_{ij}(\mathbf{x}) = d_i(\mathbf{x}) - d_j(\mathbf{x}) = \mathbf{x}^T(\mathbf{m}_i - \mathbf{m}_j) - \frac{1}{2}(\mathbf{m}_i - \mathbf{m}_j)^T(\mathbf{m}_i - \mathbf{m}_j) = 0. \qquad (12.2\text{-}6)$$

Заданная уравнением (12.2-6) разделяющая поверхность представляет собой перпендикуляр к отрезку, соединяющему точки $\mathbf{m}_i$ и $\mathbf{m}_j$ в пространстве признаков, проведенный через его середину (см. задачу 12.3). В случае $n = 2$ перпендикуляр представляет собой линию, при $n = 3$ — плоскость, а при $n > 3$ он называется *гиперплоскостью.*

Пример 12.1. Иллюстрация классификатора по минимуму расстояния.

■ На рис. 12.6 показаны два класса образов, выделенных из выборки измерений цветков ириса на рис. 12.1. Для этих двух классов, *Iris versicolor* и *Iris setosa*, обозначенных соответственно ω_1 и ω_2, выборочные оценки векторов математического ожидания равны $\mathbf{m}_1 = (4{,}3,\ 1{,}3)^T$ и $\mathbf{m}_2 = (1{,}5,\ 0{,}3)^T$. Согласно выражению (12.2-5), дискриминантные функции имеют вид

$$d_1(\mathbf{x}) = \mathbf{x}^T\mathbf{m}_1 - \frac{1}{2}\mathbf{m}_1^T\mathbf{m}_1 = 4{,}3x_1 + 1{,}3x_2 - 10{,}1$$

и

$$d_2(\mathbf{x}) = \mathbf{x}^T\mathbf{m}_2 - \frac{1}{2}\mathbf{m}_2^T\mathbf{m}_2 = 1{,}5x_1 + 0{,}3x_2 - 1{,}17.$$

В соответствии с (12.2-6), уравнение разделяющей поверхности принимает вид

$$d_{12}(\mathbf{x}) = d_1(\mathbf{x}) - d_2(\mathbf{x}) = 2{,}8x_1 + 1{,}0x_2 - 8{,}9 = 0.$$

На рис. 12.6 показан график этой разделяющей поверхности (обратите внимание, что масштаб по координатным осям неодинаков). Подстановка в полученное уравнение любого вектора признаков объекта из класса ω_1 приводит к ре-

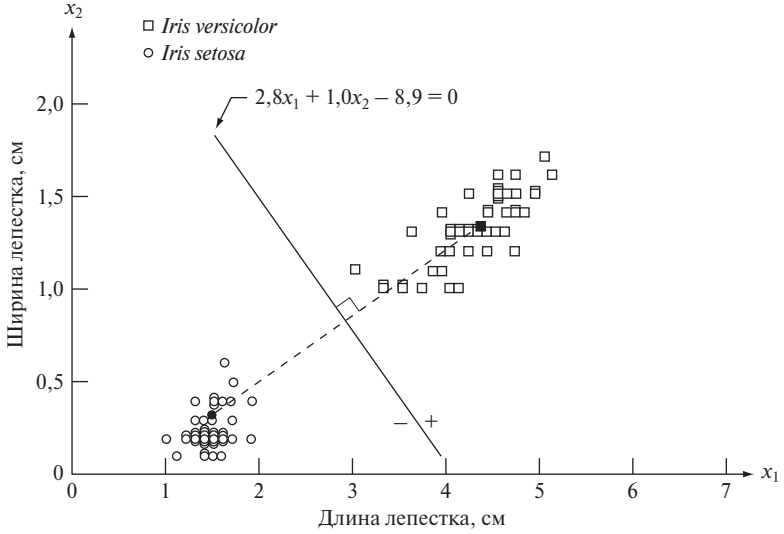


Рис. 12.6. Разделяющая поверхность классификатора по минимуму расстояния для классов *Iris versicolor* и *Iris setosa*. Кружок и квадратик черного цвета указывают положения математических ожиданий обоих классов

зультату $d_{12}(\mathbf{x}) > 0$, а из класса ω_2 — напротив, к $d_{12}(\mathbf{x}) < 0$. Другими словами, чтобы определить, к какому из двух этих классов относится неизвестный образ, достаточно исследовать знак функции $d_{12}(\mathbf{x})$. ■

Классификатор по минимуму расстояния хорошо работает в тех практических задачах, где расстояния между точками математических ожиданий классов велики по сравнению с размахом значений векторов признаков объектов каждого класса. В разделе 12.2.2 мы покажем, что оптимальные (в смысле минимизации среднего риска ошибочного распознавания) характеристики классификатора по минимуму расстояния достигаются, когда распределение каждого класса имеет форму гиперсферы в n-мерном пространстве признаков с центром в точке его математического ожидания.

На практике редко встречаются случаи, когда одновременно и значения математических ожиданий классов далеко разнесены друг от друга, и размах значений объектов каждого класса достаточно мал; разве что сама природа исходных данных находится под контролем проектировщика системы. Прекрасным примером такого рода могут служить системы автоматического чтения знаков стилизованного печатного шрифта, например хорошо известного шрифта E-13B, используемого Американской банковской ассоциацией[3]. Как видно из рис. 12.7, этот шрифт состоит из 14 символов, которые были специальным образом нарисованы на сетке 9×7 элементов, чтобы упростить их считывание. Эти знаки обычно печатаются типографской краской, содержащей мелкий порошкообразный магнитный материал. Перед считыванием документ с краской попадает в магнитное поле, которое усиливает каждый символ, упрощая его выделение. Иначе говоря, задача сегментации решается с помощью искусственного подчеркивания ключевых характеристик каждого символа.

Символы обычно сканируются справа налево в горизонтальном направлении магнитной считывающей головкой с одиночным зазором, ширина которого много меньше ширины символа, а высота превышает высоту символа. При прохождении головки над символом она выдает одномерный электрический сигнал (сигнатуру), который нормируется так, чтобы его амплитуда была пропорциональна скорости убывания или возрастания суммарной площади линий символа в зоне зазора головки. Например, рассмотрим на рис. 12.7 форму сигнала, связанного с цифрой 0. При движении головки справа налево суммарная площадь символа под зазором сначала увеличивается, что дает положительную производную (положительную скорость изменения площади). По мере того как зазор головки выходит за правую боковину цифры, площадь начинает уменьшаться, давая отрицательную производную. Когда головка находится в средней зоне символа между боковинами, площадь остается примерно постоянной, что соответствует нулевой производной. Такая же последовательность повторяется при проходе головки над левой боковиной символа. Особое начертание шрифта гарантирует, что форма сигнала для каждого знака отличается от других, и этим также обеспечивается то, что положения пиков и нулевых значений сигнала приблизительно совпадают с вертикальными линиями сетки, как это показано на рис. 12.7. Особенностью шрифта E-13B является то, что отсчеты сигнала

[3] Таким шрифтом, в частности, печатается код банка и номер счета на банковских чеках в США. — *Прим. перев.*

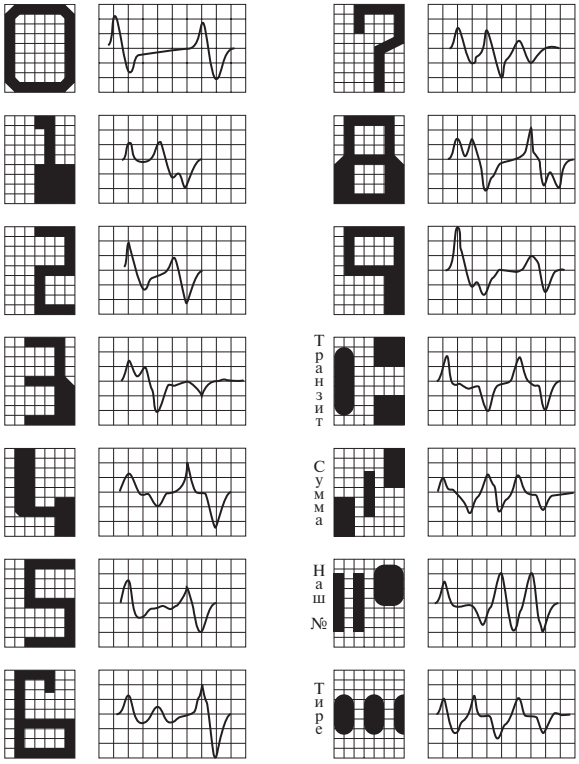


Рис. 12.7. Набор символов шрифта E-13B Американской банковской ассоциации и соответствующие им формы сигналов

только в этих точках дают достаточно информации для правильной классификации знаков. Применение магнитной краски помогает получать сигналы чистой формы с минимальным разбросом.

Классификатор по минимуму расстояния для такого приложения можно построить достаточно просто. Запомним набор значений каждого сигнала в точках, соответствующих сетке дискретизации, придав этому набору смысл вектора прототипа $\mathbf{m}_j$, $j = 1, 2,..., 14$. При поступлении неизвестного символа, подлежащего классификации, он сканируется описанным выше способом, и выходной сигнал дискретизуется по той же сетке. Набор полученных значений образует вектор $\mathbf{x}$, после чего символ будет отнесен к тому классу, для которого вектор прототипа максимизирует значение $d_i(\mathbf{x})$ в (12.2-5). При использовании аналоговых схем, составленных из блоков резисторов (см. задачу 12.4), может быть достигнута достаточно высокая скорость классификации.

Корреляционное сопоставление

Теоретические основы пространственной корреляции были изложены в разделе 3.4.2, где корреляция широко использовалась для фильтрации в пространственной области. Теорема о корреляции также упоминалась в разделе 4.6.7 и табл. 4.3. Как известно из формулы (3.4-1), корреляция маски $w(x, y)$ размерами $m \times n$ с изображением $f(x, y)$ размерами $M \times N$ задается выражением

Для большей строгости следовало бы говорить об *автокорреляции*, когда w и f — одна и та же функция, или о *взаимной корреляции*, когда функции различные. Однако обычно используют общий термин «корреляция», если из контекста конкретной задачи ясно, совпадают функции или отличаются.

$$c(x,y)=\sum_s\sum_t w(s,t)\,f(x+s,y+t), \tag{12.2-7а}$$

где суммирование ведется по области пересечения w и f. Это выражение вычисляется для всевозможных значений переменных сдвига x и y, так чтобы каждый элемент w в какой-то момент попадал на каждый пиксель f, в предположении, что размеры изображения больше размеров маски. Аналогично тому, как (согласно теореме о свертке) пространственная свертка связана с преобразованием Фурье, пространственная корреляция связана с преобразованием функций согласно теореме о корреляции

$$f(x,y)\star w(x,y)\Leftrightarrow F^*(u,v)W(u,v), \tag{12.2-7б}$$

где символ ☆ означает пространственную корреляцию, а * — операцию комплексного сопряжения. Вторая половина теоремы о корреляции, приведенная в табл. 4.3, в данном обсуждении не используется. Соотношение (12.2-7б) определяет Фурье-пару, интерпретация которой аналогична обсуждению формулы (4.6-24), за исключением того, что для одной из функций берется комплексно сопряженное значение. Применение обратного преобразования Фурье к соотношению (12.2-7б) дает двумерную круговую корреляцию, аналогичную соотношению (4.6-23), и вопрос о расширении исходного изображения, рассмотренный в разделе 4.6.6 в связи со сверткой, также относится и к корреляции.

Мы не будем подробно останавливаться на вышеприведенных формулах, поскольку они обе зависят от изменения масштабов значений f и w. Вместо этого будем использовать нормированную величину — *коэффициент корреляции*, имеющий следующий вид:

$$\gamma(x,y)=\frac{\sum_s\sum_t\left[w(s,t)-\bar{w}\right]\left[f(x+s,y+t)-\bar{f}_{xy}\right]}{\left\{\sum_s\sum_t\left[w(s,t)-\bar{w}\right]^2\sum_s\sum_t\left[f(x+s,y+t)-\bar{f}_{xy}\right]^2\right\}^{\frac{1}{2}}}, \tag{12.2-8}$$

Читателю рекомендуется освежить в памяти раздел 3.4.2, в котором описывается механизм пространственной корреляции.

где суммирование ведется по всем парам координат, общим для f и w, $\bar{w}$ — среднее значение элементов маски w (вычисляемое только один раз), а $\bar{f}_{xy}$ — среднее значение элементов изображения f по области[4], совпадающей с текущим положением w. Маску w часто называют *эталоном*, а вычисление корреляции — *сопоставлением с эталоном*. Коэффициент корреляции $\gamma(x, y)$ изменяется в диа-

[4] Также предполагается, что f и w на области пересечения не являются константами. — *Прим. перев.*

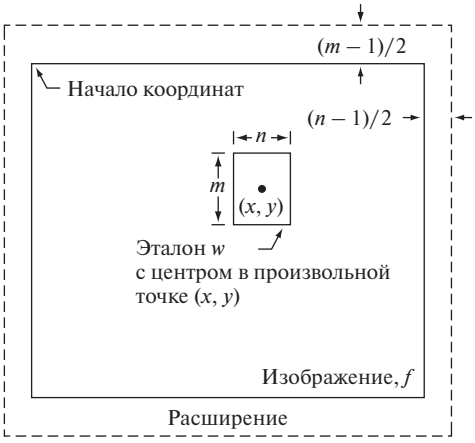


Рис. 12.8. Схема корреляционного сопоставления с эталоном

пазоне от –1 до 1 и не зависит от изменения масштаба значений f и w (см. задачу 12.5). Максимальное значение $\gamma(x, y)$ достигается, когда нормализованный (вычитанием среднего значения) эталон и соответствующая область в f (также нормализованная) имеют наибольшее сходство в смысле соотношения (12.2-28). Минимум достигается, когда обе нормализованные функции имеют наименьшее сходство. Коэффициент корреляции нельзя вычислить с помощью преобразования Фурье ввиду наличия нелинейных членов в выражении (возведение в квадрат и извлечение корня).

Рис. 12.8 иллюстрирует выполнение описанной процедуры корреляции. Добавление элементов на краях f является необходимым расширением для ситуаций, когда центр w оказывается вблизи края f, как объяснялось в разделе 3.4.2. (Когда производится сопоставление с эталоном, при выходе центра эталона за край изображения значение корреляции обычно не представляет интереса, поэтому величина расширения в каждую сторону ограничивается половиной соответствующего размера маски.) Как обычно, для удобства обозначений мы ограничимся рассмотрением эталонов с нечетными размерами.

На рис. 12.8 показан эталон размерами $m \times n$ с центром в произвольной точке (x, y). В этой точке с помощью формулы (12.2-8) вычисляется значение коэффициента корреляции, после чего центр эталона перемещается в соседнюю точку и повторяется та же процедура. Полное множество коэффициентов корреляции $\gamma(x, y)$ находится перемещением центра эталона (т. е. перебором значений x и y) так, чтобы *центр* w пробегал все пиксели f. По окончании этой процедуры отыскивается максимальное значение $\gamma(x, y)$, чтобы определить точку (x, y), в которой достигается наилучшее совпадение области изображения с эталоном. Могут найтись несколько точек с одинаковым максимальным значением $\gamma(x, y)$, что указывает на наличие в f нескольких областей одинаковой степени совпадения с w.

Пример 12.2. Сопоставление с помощью корреляции.

■ На рис. 12.9(*a*) приведено полученное со спутника изображение урагана Эндрю (размеры изображения 913×913), на котором отчетливо виден «глаз» урагана. В качестве примера использования корреляции мы хотим найти на этом

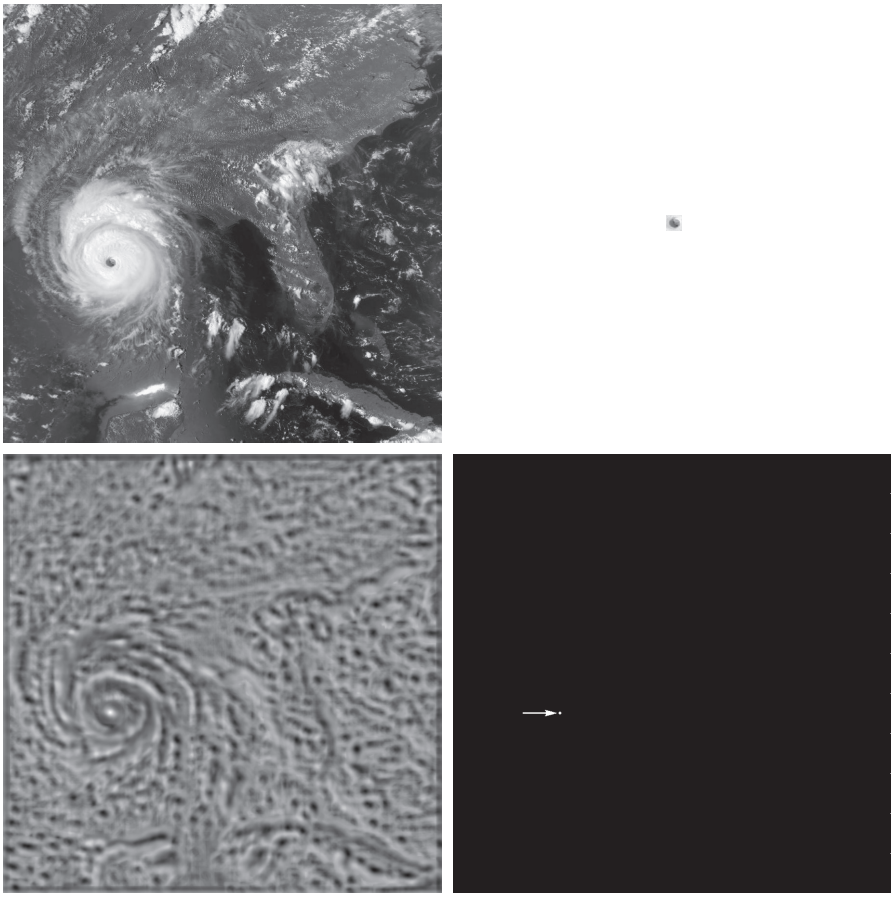


Рис. 12.9. (*а*) Космическое изображение урагана Эндрю, полученное 24 августа 1992 г. (*б*) Эталон «глаза» урагана. (*в*) Коэффициент корреляции в виде изображения (обратите внимание на самую яркую точку). (*г*) Точка наилучшего совпадения; она представляет собой одиночный пиксель, поэтому искусственно увеличена для большей наглядности. (Исходное изображение предоставлено службой NOOA)

изображении точку наилучшего сопоставления с эталоном, показанным на рис. 12.9(*б*), — небольшим (31×31 пиксель) фрагментом изображения глаза урагана. Результат вычисления коэффициента корреляции согласно формуле (12.2-8) показан как изображение на рис. 12.9(*в*), которое имело размеры 943×943 (вследствие расширения, см. рис. 12.8), но для наглядности было обрезано до исходных размеров. Яркости на этом изображении пропорциональны значению коэффициента корреляции, причем все отрицательные значения устанавливались в 0 (черное) для удобства визуального анализа. Самая яркая точка корреляционного изображения отчетливо видна вблизи глаза урагана. На рис. 12.9(*г*) положение максимума корреляции отмечено белой точкой (в данном случае достигается единственное максимальное значение, равное 1), которая находится вблизи местоположения глаза урагана на исходном изображении. ■

Хотя выше было показано, что корреляционная функция может быть нормирована относительно изменений амплитуды обрабатываемых функций путем перехода к коэффициенту корреляции, достичь нормировки относительно поворота или изменения размера не так просто. Нормировка относительно размеров связана с пространственным масштабированием, которое, как объяснялось в разделах 2.6.5 и 4.5.4, влечет за собой необходимость повторной дискретизации изображения или эталона. Для осмысленного изменения масштаба необходимо знать правильные соотношения их размеров, что может представлять серьезные трудности при отсутствии дополнительной пространственной информации. Например, в прикладных задачах дистанционного зондирования, где геометрические параметры поля зрения сенсора известны (что является обычной ситуацией), знания высоты сенсора над наблюдаемой поверхностью может быть достаточно для нормировки размера изображения, если угол съемки постоянный. Аналогично для нормировки относительно поворота необходимо знать угол, на который следует повернуть изображение (или эталон), что опять-таки требует пространственных подсказок. В упомянутой задаче дистанционного зондирования информации о направлении полета может быть достаточно для поворота получаемых изображений до стандартной ориентации. В общем случае, когда дополнительная пространственная информация недоступна, нормировка относительно размеров и поворота может стать весьма трудной задачей; для ее решения необходимо с помощью методов, описанных в главе 11, автоматически находить на изображении признаки, которые могли бы служить пространственной подсказкой.

12.2.2. Статистически оптимальные классификаторы

В этом разделе излагается вероятностный подход к распознаванию. Как и в большинстве областей, связанных с измерением и интерпретацией физических явлений, вероятностные подходы оказываются важными в задаче распознавания образов из-за случайностей, влияющих на порождение классов образов. Как будет видно из дальнейшего рассмотрения, можно выработать метод классификации, оптимальный в том смысле, что при его использовании достигается наименьшая (в среднем) вероятность появления ошибок классификации (см. задачу 12.10).

Основы

Обозначим через $p(\omega_i|\mathbf{x})$ вероятность того, что поступивший образ $\mathbf{x}$ принадлежит классу ω_i. Если классификатор относит к классу ω_j образ $\mathbf{x}$, в действительности принадлежащий классу ω_i, это приводит к потерям, которые обозначаются L_{ij}. Поскольку образ $\mathbf{x}$ может принадлежать любому из рассматриваемых W классов, средняя величина потерь, связанных с отнесением $\mathbf{x}$ к классу ω_j, равна

$$r_j(\mathbf{x}) = \sum_{k=1}^{W} L_{kj}\, p(\omega_k|\mathbf{x}). \tag{12.2-9}$$

Согласно терминологии теории принятия решений, эта величина называется (*условным*) *средним риском* (или *потерями*).

Из элементарной теории вероятностей известно, что при $p(A) > 0$ и $p(B) > 0$ справедливо равенство $p(A|B) = [p(A)\,p(B|A)] / p(B)$. С его использованием перепишем выражение (12.2-9) в следующей форме:

Краткий обзор теории вероятностей имеется на интернет-сайте книги.

$$r_j(\mathbf{x}) = \frac{1}{p(\mathbf{x})} \sum_{k=1}^{W} L_{kj}\, p(\mathbf{x}|\omega_k) P(\omega_k), \tag{12.2-10}$$

где $p(\mathbf{x}|\omega_k)$ — функция плотности распределения вероятностей образов класса ω_k, а $P(\omega_k)$ — вероятность появления образа из класса ω_k (эти вероятности иногда называют *априорными*). Поскольку множитель $1/p(\mathbf{x})$ положителен и одинаков для всех $r_j(\mathbf{x})$, $j = 1, 2,..., W$, его в (12.2-10) можно опустить, при этом упорядоченность значений функций $r_j(\mathbf{x})$ не изменится. Тогда выражение для условных средних потерь (с точностью до постоянного множителя) сводится к

$$r_j(\mathbf{x}) = \sum_{k=1}^{W} L_{kj}\, p(\mathbf{x}|\omega_k) P(\omega_k). \tag{12.2-11}$$

Классификатор имеет возможность отнести поступивший неизвестный образ к любому из W классов. Если для каждого образа $\mathbf{x}$ вычислить функции $r_1(\mathbf{x}), r_2(\mathbf{x}),..., r_W(\mathbf{x})$ и приписать этот образ к тому классу, для которого потери минимальны, то суммарное значение средних потерь по всем решениям будет минимальным. Такой классификатор, минимизирующий суммарную величину средних потерь, называется *байесовским классификатором*. Итак, байесовский классификатор относит неизвестный образ $\mathbf{x}$ к классу ω_i, если $r_i(\mathbf{x}) < r_j(\mathbf{x})$ для $j = 1, 2,..., W$; $j \neq i$. Последнее неравенство можно записать в виде

$$\sum_{k=1}^{W} L_{ki}\, p(\mathbf{x}|\omega_k) P(\omega_k) < \sum_{q=1}^{W} L_{qj}\, p(\mathbf{x}|\omega_q) P(\omega_q) \tag{12.2-12}$$

для всех $j \neq i$. Обычно величину потерь L_{ij} при правильном выборе класса принимают равной нулю, а при ошибочном решении потери предполагают одинаковыми и приписывают им некоторое ненулевое значение, скажем, 1. При таких условиях функция потерь принимает вид

$$L_{ij} = 1 - \delta_{ij}, \tag{12.2-13}$$

где $\delta_{ij} = 1$ при $i = j$ и $\delta_{ij} = 0$ при $i \neq j$, то есть потери равны 1 при ошибочном выборе класса и 0 при правильном решении. Функцию δ_{ij} называют *симметричной* или *нуль-единичной* функцией потерь. Подставляя (12.2-13) в (12.2-11), получаем

$$r_j(\mathbf{x}) = \sum_{k=1}^{W} (1 - \delta_{kj})\, p(\mathbf{x}|\omega_k) P(\omega_k) = p(\mathbf{x}) - p(\mathbf{x}|\omega_j) P(\omega_j). \tag{12.2-14}$$

Тогда байесовский классификатор приписывает образ $\mathbf{x}$ к классу ω_i, если для всех $j \neq i$,

$$p(\mathbf{x}) - p(\mathbf{x}|\omega_i) P(\omega_i) < p(\mathbf{x}) - p(\mathbf{x}|\omega_j) P(\omega_j), \tag{12.2-15}$$

или, что то же самое, если

$$p(\mathbf{x}|\omega_i)P(\omega_i) > p(\mathbf{x}|\omega_j)P(\omega_j) \quad j = 1,2,...,W;\ j \neq i. \tag{12.2-16}$$

С учетом обсуждения, которое привело к неравенству (12.2-1), мы видим, что байесовский классификатор в случае нуль-единичной функции потерь есть не что иное, как вычисление дискриминантных функций вида

$$d_j(\mathbf{x}) = p(\mathbf{x}|\omega_j)P(\omega_j) \quad j = 1,2,...,W \tag{12.2-17}$$

с отнесением образа $\mathbf{x}$ к тому классу ω_i, для которого значение дискриминантной функции $d_i(\mathbf{x})$ оказывается наибольшим.

Дискриминантные функции, заданные соотношениями (12.2-17), являются оптимальными в том смысле, что они минимизируют величину средних потерь из-за ошибочной классификации. Однако для достижения оптимальности должны быть известны как функции плотности распределения вероятностей образов каждого класса, так и вероятности появления каждого из классов. Последнее требование обычно не создает трудностей. Например, если все классы являются равновероятными, то $P(\omega_i) = 1/W$. Даже если это и не так, вероятности классов обычно можно оценить на основе априорных сведений о задаче. Совершенно по-другому обстоит дело с оценкой функции плотности распределения $p(\mathbf{x}|\omega_k)$. Если векторы признаков $\mathbf{x}$ являются n-мерными, то $p(\mathbf{x}|\omega_k)$ есть функция n переменных, для оценки которой (в случае функции неизвестного вида) необходимы методы теории вероятностей, описывающие многомерные случайные величины. Эти методы трудно применить на практике, особенно в случаях недостаточно представительных выборок образов из каждого класса или недостаточно хороших форм функций плотности распределения вероятностей. По этим причинам при использовании байесовского классификатора для различных функций плотности распределения вероятностей обычно исходят из предполагаемых аналитических выражений общего вида, оценивая затем их параметры по выборке образов из каждого класса. Наиболее часто используется предположение о том, что функция $p(\mathbf{x}|\omega_k)$ описывается гауссовой функцией плотности распределения вероятностей (т. е. класс имеет *нормальное* распределение). Чем ближе предположения о функциях распределения к действительности, тем точнее достигается минимум средних потерь в результате использования байесовского классификатора.

Байесовский классификатор для классов с нормальным распределением

Для начала рассмотрим одномерную ($n = 1$) задачу распознавания образов в случае двух классов ($W = 2$) с нормальными распределениями, характеризующимися математическими ожиданиями m_1 и m_2 и стандартными отклонениями σ_1 и σ_2. Согласно (12.2-17), байесовские дискриминантные функции имеют вид

$$d_j(x) = p(x|\omega_j)P(\omega_j) = \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{(x-m_j)^2}{2\sigma_j^2}} P(\omega_j) \quad j = 1,2, \tag{12.2-18}$$

где образы представляют собой скалярную величину x. На рис. 12.10 изображены графики функций плотности распределения вероятностей для этих двух классов. Разделяющая поверхность между классами есть точка x_0, такая, что $d_1(x_0) = d_2(x_0)$. Если указанные два класса являются равновероятными, то $P(\omega_1) = P(\omega_2) = 1/2$

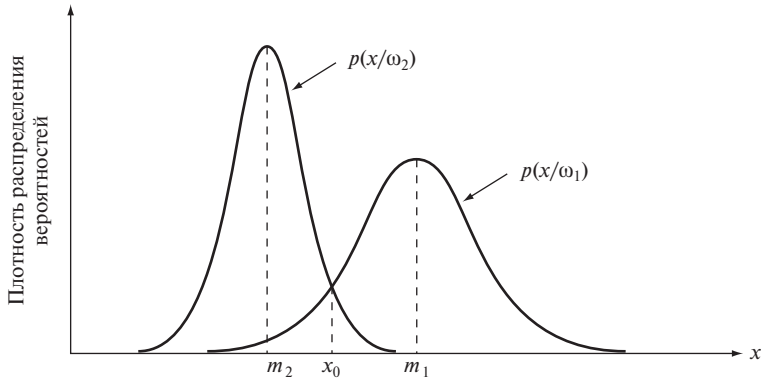


Рис. 12.10. Функции плотности распределения вероятностей для двух одномерных классов. Точка x_0 указывает положение разделяющей поверхности в случае, когда классы равновероятны

и координата точки x_0 определяется из условия $p(x_0|\omega_1) = p(x_0|\omega_2)$, что соответствует точке пересечения графиков функций плотности распределения вероятностей, как показано на рис. 12.10. Любой образ (т. е. точка), лежащий правее x_0, классифицируется как принадлежащий классу ω_1, а расположенный левее x_0 — классу ω_2. Если вероятности появления образов каждого класса неодинаковы, то точка x_0 сдвигается влево, если бóльшую вероятность имеет класс ω_1, и вправо, если более вероятным является класс ω_2. Этого результата следовало ожидать, поскольку данный классификатор стремится минимизировать потери от ошибочной классификации. Например, в предельном случае, если объекты класса ω_2 не встречаются вообще, то классификатор не совершит ошибки, относя все образы к классу ω_1, т. е. считая $x_0 = -\infty$.

См. примечания в конце данного раздела касательно того, что байесовский классификатор для одной переменной является оптимальной функцией *порогового разделения*, как отмечалось в разделе 10.3.3.

В n-мерном случае гауссова функция плотности распределения вероятностей векторов j-го класса имеет вид

$$p(\mathbf{x}|\omega_j) = \frac{1}{(2\pi)^{n/2}|\mathbf{C}_j|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\mathbf{m}_j)^T \mathbf{C}_j^{-1}(\mathbf{x}-\mathbf{m}_j)}, \qquad (12.2\text{-}19)$$

где каждая функция $p(\mathbf{x}|\omega_j)$ полностью задается вектором математического ожидания $\mathbf{m}_j$ и ковариационной матрицей $\mathbf{C}_j$, которые определяются как

$$\mathbf{m}_j = E_j\{\mathbf{x}\} \qquad (12.2\text{-}20)$$

и

$$\mathbf{C}_j = E_j\{(\mathbf{x}-\mathbf{m}_j)(\mathbf{x}-\mathbf{m}_j)^T\}, \qquad (12.2\text{-}21)$$

где запись $E_j\{\cdot\}$ означает математическое ожидание значения аргумента на образах из класса ω_j. В формуле (12.2-19) n есть размерность пространства признаков, а $|\mathbf{C}_j|$ — определитель матрицы $\mathbf{C}_j$. Приближая математические ожидания E_j

средними значениями, получаем следующие оценки для вектора математического ожидания и ковариационной матрицы:

$$\mathbf{m}_j = \frac{1}{N_j}\sum_{\mathbf{x}\in\omega_j}\mathbf{x} \tag{12.2-22}$$

и

$$\mathbf{C}_j = \frac{1}{N_j}\sum_{\mathbf{x}\in\omega_j}\mathbf{x}\mathbf{x}^T - \mathbf{m}_j\mathbf{m}_j^T, \tag{12.2-23}$$

где N_j — число образов из класса ω_j, по которым проводится суммирование. Ниже в этом разделе мы приведем пример использования этих двух выражений.

Ковариационная матрица является симметрической и положительно полуопределенной. Как объяснялось в разделе 11.4, ее диагональные элементы c_{kk} являются дисперсиями компонент x_k вектора признаков, а лежащие вне диагонали элементы c_{jk} — ковариациями компонент x_j и x_k этого вектора. Если все недиагональные элементы матрицы ковариации нулевые, то многомерная гауссова функция распределения разлагается на произведение одномерных гауссовых плотностей каждой компоненты вектора **x**. Это происходит в том случае, если компоненты x_j и x_k этого вектора (признаки j и k) некоррелированы.

Краткий обзор векторов и матриц имеется на интернет-сайте книги.

В соответствии с (12.2-17) байесовская дискриминантная функция для класса ω_j есть $d_j(\mathbf{x}) = p(\mathbf{x}|\omega_j)P(\omega_j)$. Однако, учитывая экспоненциальный вид гауссовой плотности распределения, удобнее иметь дело с натуральным логарифмом этой функции. Другими словами, мы будем использовать дискриминантные функции вида

$$d_j(\mathbf{x}) = \ln\left[p(\mathbf{x}|\omega_j)P(\omega_j)\right] = \ln p(\mathbf{x}|\omega_j) + \ln P(\omega_j). \tag{12.2-24}$$

Данное выражение эквивалентно (12.2-17) с точки зрения классификации, поскольку логарифм — монотонно возрастающая функция. Иначе говоря, *упорядоченность* значений дискриминантных функций, задаваемых выражениями (12.2-17) и (12.2-24), будет одинаковой. Подставляя (12.2-19) в (12.2-24), получаем

$$d_j(\mathbf{x}) = \ln P(\omega_j) - \frac{n}{2}\ln 2\pi - \frac{1}{2}\ln|\mathbf{C}_j| - \frac{1}{2}\left[(\mathbf{x}-\mathbf{m}_j)^T\mathbf{C}_j^{-1}(\mathbf{x}-\mathbf{m}_j)\right]. \tag{12.2-25}$$

Член $(n/2)\ln 2\pi$ одинаков для всех классов, так что из (12.2-25) его можно исключить, и тогда получим

$$d_j(\mathbf{x}) = \ln P(\omega_j) - \frac{1}{2}\ln|\mathbf{C}_j| - \frac{1}{2}\left[(\mathbf{x}-\mathbf{m}_j)^T\mathbf{C}_j^{-1}(\mathbf{x}-\mathbf{m}_j)\right] \quad j = 1,2,...,W. \tag{12.2-26}$$

Эта формула выражает байесовские дискриминантные функции для классов с нормальным распределением при условии нуль-единичной функции потерь.

Дискриминантные функции, заданные формулами (12.2-26), определяют поверхности второго порядка, поскольку являются квадратичными функциями в n-мерном пространстве, не содержащими членов с компонентами векто-

ра **x** выше второй степени. Поэтому ясно, что для нормально распределенных классов байесовский классификатор строит между каждой парой классов разделяющую поверхность второго порядка общего вида. Если генеральные совокупности образов каждого класса действительно описываются нормальными распределениями, то никакая другая поверхность не позволит получить меньшую величину средних потерь от ошибочной классификации.

Если ковариационные матрицы для всех классов одинаковы, т. е. $\mathbf{C}_j = \mathbf{C}$ для $j = 1, 2, \ldots, W$, то, раскрывая выражение в правой части (12.2-26) и отбрасывая все члены, не зависящие от j, получим

$$d_j(\mathbf{x}) = \ln P(\omega_j) + \mathbf{x}^T\mathbf{C}^{-1}\mathbf{m}_j - \frac{1}{2}\mathbf{m}_j{}^T\mathbf{C}^{-1}\mathbf{m}_j \quad j = 1,2,\ldots,W, \tag{12.2-27}$$

т. е. дискриминантные функции оказываются линейными и задают разделяющие *гиперплоскости*.

Если вдобавок $\mathbf{C} = \mathbf{I}$ (где $\mathbf{I}$ — единичная матрица) и $P(\omega_j) = 1/W$ для $j = 1, 2, \ldots, W$ (классы равновероятны), то

$$d_j(\mathbf{x}) = \mathbf{x}^T\mathbf{m}_j - \frac{1}{2}\mathbf{m}_j^T\mathbf{m}_j \quad j = 1,2,\ldots,W. \tag{12.2-28}$$

Это выражение задает дискриминантные функции для классификатора по минимуму расстояния и совпадает с ранее приведенной формулой (12.2-5). Таким образом, классификатор по минимуму расстояния является оптимальным в байесовском смысле, если (1) классы имеют нормальное распределение, (2) все ковариационные матрицы единичные и (3) все классы равновероятны. Удовлетворяющие перечисленным требованиям нормально распределенные классы имеют в n-мерном пространстве признаков форму одинаковых *гиперсфер*. Классификатор по минимуму расстояния строит разделяющую гиперплоскость для каждой пары классов, проходящую перпендикулярно через середину отрезка, соединяющего центры этих двух классов. В двумерном случае классы образуют области круглой формы, а разделяющие поверхности имеют форму линий, перпендикулярных отрезкам, соединяющим центры каждой пары таких кругов, и проведенных через середины таких отрезков.

Пример 12.3. Байесовский классификатор для трехмерных образов.

■ На рис. 12.11 показан простой пример расположения двух классов образов в трехмерном пространстве. Проиллюстрируем с помощью этих образов технику реализации байесовского классификатора, предполагая, что образы каждого класса являются выборкой из гауссова распределения.

Применяя формулу (12.2-22) к образам на рис. 12.11, получаем:

$$\mathbf{m}_1 = \frac{1}{4}\begin{bmatrix}3\\1\\1\end{bmatrix} \quad \text{и} \quad \mathbf{m}_2 = \frac{1}{4}\begin{bmatrix}1\\3\\3\end{bmatrix}.$$

Аналогично, применяя (12.2-23) к этим двум классам, получаем их ковариационные матрицы, которые в данном случае оказываются одинаковыми:

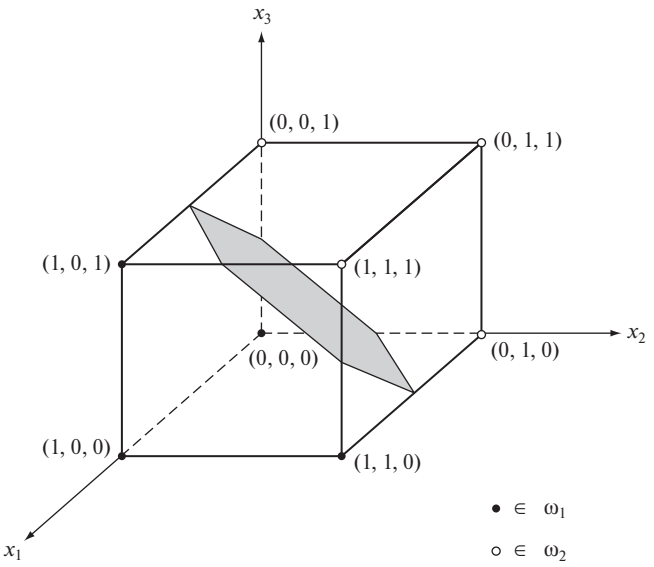


Рис. 12.11. Два простых класса образов (черные и белые точки в вершинах куба) и их байесовская разделяющая поверхность (показана серым цветом)

$$\mathbf{C}_1 = \mathbf{C}_2 = \frac{1}{16}\begin{bmatrix} 3 & 1 & 1 \\ 1 & 3 & -1 \\ 1 & -1 & 3 \end{bmatrix}.$$

Поскольку ковариационные матрицы одинаковы, байесовские дискриминантные функции определяются выражениями (12.2-27). Если предположить, что $P(\omega_1) = P(\omega_2) = 1/2$, то из (12.2-28) получаем

$$d_j(\mathbf{x}) = \mathbf{x}^T\mathbf{C}^{-1}\mathbf{m}_j - \frac{1}{2}\mathbf{m}_j{}^T\mathbf{C}^{-1}\mathbf{m}_j,$$

где

$$\mathbf{C}^{-1} = \begin{bmatrix} 8 & -4 & -4 \\ -4 & 8 & 4 \\ -4 & 4 & 8 \end{bmatrix}.$$

Проведя для $d_j(\mathbf{x})$ матричные вычисления, получаем следующие дискриминантные функции:

$$d_1(\mathbf{x}) = 4x_1 - 1{,}5 \quad \text{и} \quad d_2(\mathbf{x}) = -4x_1 + 8x_2 + 8x_3 - 5{,}5.$$

Уравнение разделяющей поверхности для этих двух классов имеет вид

$$d_1(\mathbf{x}) - d_2(\mathbf{x}) = 8x_1 - 8x_2 - 8x_3 + 4 = 0.$$

На рис. 12.11 показано сечение единичного куба этой поверхностью, которое демонстрирует, что классы эффективно разделяются. ■

Одним из наиболее успешных применений байесовского классификатора является его использование в задаче классификации данных дистанционного зондирования, регистрируемых при помощи мультиспектральных сканеров, устанавливаемых на борту самолета, спутника или орбитальной станции. Учитывая огромный объем данных, передаваемых подобным оборудованием, задача автоматического анализа и классификации изображений вызывает значительный интерес. Прикладные задачи дистанционного зондирования весьма разнообразны и включают землепользование, прогнозирование урожая, обнаружение заболеваний пищевых растений, наблюдение за лесными ресурсами, мониторинг качества воздуха и воды, геологические исследования, прогнозирование погоды, а также множество других приложений, важных для контроля окружающей среды. Ниже рассматривается пример типичной прикладной задачи.

Пример 12.4. Классификация мультиспектральных данных с помощью байесовского классификатора.

■ Как уже говорилось в разделах 1.3.4 и 11.4, мультиспектральный сканер регистрирует энергию электромагнитного излучения в заданных участках спектра. Таковыми могут быть, например, участки 0,45—0,52, 0,52—0,60, 0,63—0,69 и 0,76—0,90 мкм, которые находятся в диапазонах синего, зеленого, красного и ближнего инфракрасного излучения соответственно. При сканировании земной поверхности в указанных диапазонах регистрируются четыре цифровых изображения заданного участка — по одному для каждого диапазона спектра. Сформированные изображения являются несмещенными (наложенными друг на друга), что иллюстрирует рис. 12.12. Следовательно, в точности, как это делалось в разделе 11.4, каждая точка земной поверхности может быть представлена 4-компонентным вектором признаков вида $\mathbf{x} = (x_1, x_2, x_3, x_4)^T$, где x_1 — яркость в синем участке спектра, x_2 — в зеленом и т.д. Если размеры изображений составляют 512×512 пикселей, то каждый комплект мультиспектральных изображений можно представить с помощью 262144 четырехмерных

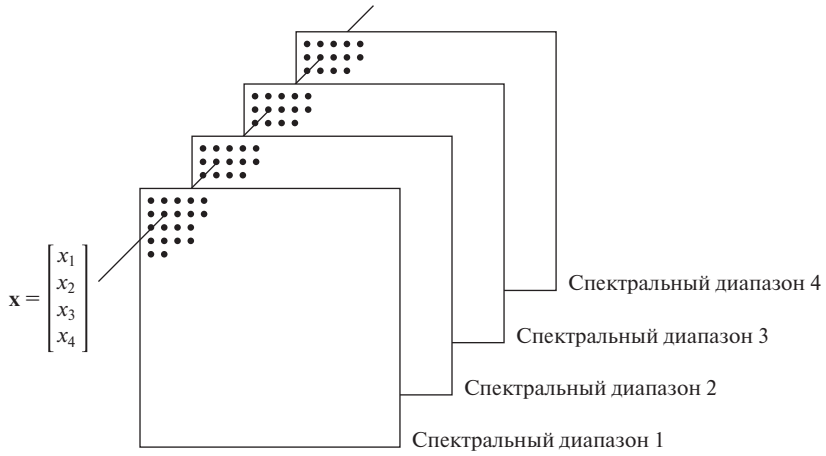


Рис. 12.12. Построение вектора признаков из соответственных пикселей четырех цифровых изображений, зарегистрированных с помощью мультиспектрального сканера

образов. Как отмечалось выше, при построении байесовского классификатора образов с гауссовым распределением требуется получить оценки вектора математического ожидания и ковариационной матрицы для каждого из классов.

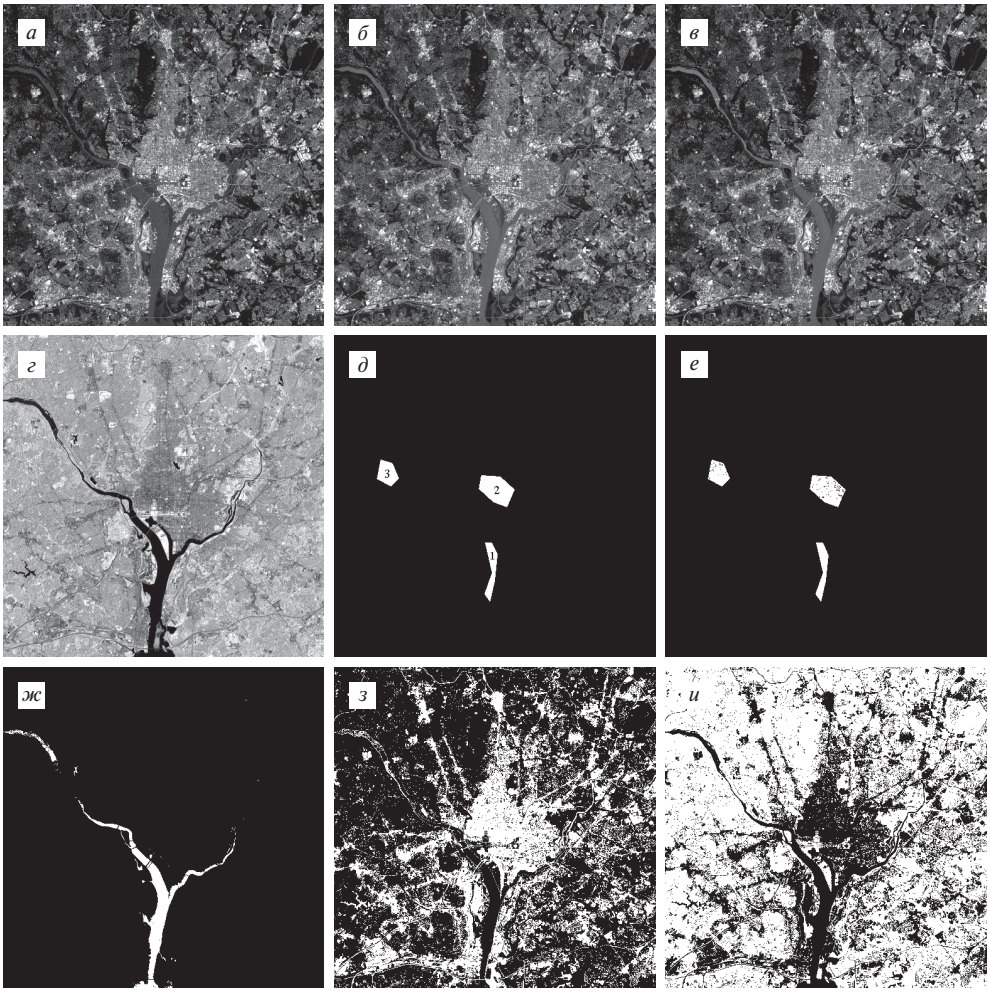


Рис. 12.13. Байесовская классификация мультиспектральных данных. (*а*)—(*г*) Изображения синей, зеленой, красной и ближней инфракрасной компонент. (*д*) Маски представительных областей для формирования выборок точек: водной поверхности (1), городской застройки (2) и растительности (3). (*е*) Результаты классификации, полученные на двух выборках; черные точки отмечают ошибки классификации, белые соответствуют правильной классификации. (*ж*) Совокупность пикселей изображения, отнесенных к классу водной поверхности (белые точки). (*з*) Совокупность пикселей изображения, отнесенных к классу городской застройки (белые точки). (*и*) Совокупность пикселей изображения, отнесенных к классу участков растительности (белые точки)

В прикладных задачах дистанционного зондирования эти оценки вычисляются путем сбора мультиспектральных данных для каждого интересующего класса областей. Полученная выборка векторов затем используется для получения оценок векторов математического ожидания и ковариационных матриц, как это делалось в примере 12.3.

На рис. 12.13(*а*)—(*г*) приведены четыре компоненты одного мультиспектрального изображения (размерами 512×512) окрестностей г. Вашингтон (округ Колумбия), полученные съемкой в вышеуказанных диапазонах спектра. В данном случае задача состоит в классификации пикселей всей области изображения на три категории: водная поверхность, городская застройка и растительность. На рис. 12.13(*д*) показаны маски, с помощью которых из всего изображения выделяются области, представительные для каждого из трех классов. Половина каждой выборки используется для обучения (т. е. вычисления оценок векторов математического ожидания и ковариационных матриц), а другая половина — для независимого контроля с целью предварительной оценки качества построенного классификатора. Априорные вероятности классов $P(\omega_i)$ в задачах классификации мультиспектральных данных без дополнительных ограничений редко известны заранее, поэтому мы предполагаем их одинаковыми: $P(\omega_i) = 1/3$, $i = 1, 2, 3$.

В табл. 12.1 сведены результаты распознавания, полученные на обучающей и независимой контрольной[5] выборках. Процент правильно распознанных векторов примерно одинаков для обеих выборок, что говорит об устойчивости оценок параметров. В обоих случаях больше всего ошибок классификации связано с образами из класса городской застройки, что неудивительно, потому что растительность там тоже присутствует (отметим, что никакие образы из классов городской застройки и растительности не были ошибочно классифицированы как водная поверхность). На рис. 12.13(*е*) белые точки соответствуют правильному распознаванию, а черные — ошибкам классификации. В области 1 черных точек не видно, что говорит о надежном распознавании водной поверхности (встретившиеся 7 ошибочных точек находятся непосредственно вблизи границы белой области и потому незаметны).

Рис. 12.13(*ж*)—(*и*) еще более интересны. Здесь с помощью рассчитанных по обучающей выборке векторов математического ожидания и ковариационных матриц проведена классификация *всех* пикселей изображения на три упомянутые категории. На рис. 12.13(*ж*) белым цветом отмечены все пиксели, классифицированные как водная поверхность, прочие пиксели показаны черным. Как видим, байесовский классификатор отлично справился с выделением на изображении участков водной поверхности. На рис. 12.13(*з*) белым цветом отмечены все пиксели, классифицированные как городская застройка. Обратите внимание, насколько хорошо система распознала детали городского пейзажа, например дороги и мосты. На рис. 12.13(*и*) показаны пиксели, которые были распознаны как участки, покрытые растительностью. В центральной части рис. 12.13(*з*) отмечается высокая концентрация белых пикселей в деловом центре города, по мере удаления от которого концентрация уменьшается. Рис. 12.13(*и*) демонстрирует обратное явление: по мере приближения к центру города количество растительности уменьшается, все больше сменяясь городской застройкой. ■

[5] Ее также иногда называют *экзаменационной* выборкой. — *Прим. перев.*

Таблица 12.1. Байесовская классификация данных мультиспектрального изображения

Обучающая выборка						Независимая контрольная выборка					
Класс	Размер выборки	Отнесены к классу			% правильных	Класс	Размер выборки	Отнесены к классу			% правильных
		1	2	3				1	2	3	
1	484	482	2	0	99,6	1	483	478	3	2	98,9
2	933	0	885	48	94,9	2	932	0	880	52	94,4
3	483	0	19	464	96,1	3	482	0	16	466	96,7

В начале раздела 12.3.3 отмечалось, что пороговое разделение можно рассматривать как задачу байесовской классификации, при которой образы оптимальным способом разделяются на два или более классов. Фактически, как показывает последний пример, поэлементная классификация пикселей изображения в действительности сегментирует изображение на области, относящиеся к различным классам. Если используется только одна переменная (например яркость), то формула (12.2-17) определяет оптимальную функцию порогового разделения изображения по яркости пикселей, подобно тому, как это делалось в разделе 10.3. Следует учитывать, что для оптимального разделения необходимо знать и функцию плотности распределения, и априорную вероятность каждого класса, оценка которых, как уже отмечалось, является нетривиальной задачей. Если делаются какие-то предположения (например о гауссовой функции плотности), то достигаемая степень оптимальности сегментации напрямую зависит от того, насколько эти предположения близки к реальности.

12.2.3. Нейронные сети

Методы, рассмотренные в двух предыдущих разделах, основывались на использовании выборочной совокупности образов для оценивания статистических параметров каждого класса. Классификатор по минимуму расстояния полностью задается векторами математического ожидания всех классов. Аналогично байесовский классификатор для нормально распределенных совокупностей образов полностью определяется векторами математического ожидания и ковариационными матрицами каждого класса. Образы, принадлежащие к *известным* классам и используемые для оценивания упомянутых параметров, называются *обучающими*, а множество таких образов для каждого класса — *обучающей выборкой* этого класса. Процесс, в ходе которого с помощью обучающей выборки строятся дискриминантные функции, называется *обучением*.

В двух рассмотренных выше подходах сущность обучения проста. Обучающие образы каждого класса используются для вычисления параметров дискриминантной функции, соответствующей этому классу. После того, как оценки необходимых параметров получены, структура классификатора становится фиксированной, и его окончательное качество зависит лишь от того, насколько хорошо реальные совокупности образов отвечают статистическим предположениям, изначально сделанным при выводе используемого метода классификации.

В реальных задачах статистические свойства классов образов зачастую неизвестны или не поддаются оценке (вспомним упоминавшиеся выше трудности работы с многомерными статистиками). На практике для таких задач теории принятия решений более эффективными оказываются методы, в которых необходимые дискриминантные функции строятся непосредственно в ходе обучения. Это устраняет необходимость использовать предположения о функциях плотности распределения вероятностей или о каких-то других вероятностных параметрах рассматриваемых классов. В этом разделе мы обсудим различные методы, отвечающие такому критерию.

Предпосылки

Главной особенностью излагаемого ниже материала является использование большого числа простейших нелинейных вычислительных элементов (называемых *нейронами*), которые организованы в виде сетей, напоминающих предположительный способ соединения нейронов в мозге человека. Применяемые модели известны под различными названиями, в частности *нейронные сети*, *нейрокомпьютеры*, *модели параллельной распределенной обработки*, *нейроморфные системы*, *многослойные самонастраивающиеся сети* и *нейросетевые модели*. Здесь мы будем придерживаться термина *нейронные сети*. Мы воспользуемся этими сетями в качестве среды, в которой осуществляется адаптивная настройка параметров дискриминантных функций путем последовательного предъявления обучающих выборок образов из различных классов.

Интерес к нейронным сетям восходит к началу 40-х годов, чему примером является работа Мак-Каллока и Питтса [McCulloch, Pitts, 1943]. Они предложили модель нейрона в виде двоичного порогового устройства, а в качестве основы для моделирования нейронных систем — стохастические алгоритмы, в которых происходят внезапные переходы нейронов из состояния 0 в состояние 1 и наоборот. В последующей работе Хебба [Hebb, 1949] на основе математических моделей была сделана попытка ухватить концепцию обучения посредством усиления или ослабления связи.

В середине 50-х — начале 60-х годов Розенблаттом был создан новый класс так называемых *обучающихся машин* [Rosenblatt, 1959, 1962], что пробудило значительный интерес исследователей и инженеров к теории распознавания образов. Причиной большого интереса к таким машинам, названным *персептронами*, было построение математических доказательств того факта, что при обучении персептрона с помощью линейно разделимых обучающих выборок (т. е. выборок образов, для которых разделяющей поверхностью может быть гиперплоскость) сходимость к решению достигается за конечное число итеративных шагов. Решение имеет вид набора коэффициентов уравнений гиперплоскостей, правильно разделяющих классы, представленные образами из обучающей выборки.

К сожалению, те ожидания, которые последовали за открытием казавшейся хорошо обоснованной теоретической модели обучения, вскоре сменились разочарованием. Простой персептрон, как и некоторые его обобщения, были для того времени просто недостаточно мощными для решения большинства практически важных задач распознавания образов. Последующие попытки увеличить мощность машин, подобных персептрону, за счет рассмотрения несколь-

ких слоев таких устройств в принципе выглядели привлекательными, однако им не хватало эффективных алгоритмов обучения, вроде тех, которые вызвали интерес к самому персептрону. Состояние дел в области обучающихся машин в середине 60-х годов было обрисовано Нильсоном [Nilsson, 1965]. Несколькими годами позже Минский и Пейперт представили обескураживающий анализ ограниченности машин, подобных персептрону [Minsky, Papert, 1969]. Такая точка зрения продержалась до середины 80-х годов, о чем свидетельствует критический разбор Симона [Simon, 1986]. В этой работе, первоначально опубликованной на французском языке в 1984 г., Симон развенчал персептрон под заглавием «Рождение и смерть мифа».

Более поздние результаты в области разработки новых алгоритмов обучения для многослойных персептронов, полученные Румельхартом, Хинтоном и Уильямсом [Rumelhart, Hinton, Williams, 1986], существенно изменили положение дел. Предложенный ими основной метод, который часто называют *обобщенным дельта-правилом обучения посредством обратного распространения ошибки*, предлагает эффективный способ обучения многослойных машин. Хотя для такого алгоритма обучения не удается доказать конечную сходимость к правильному решению, как это было сделано для однослойного персептрона, обобщенное дельта-правило успешно было применено для решения большого числа задач, представляющих практический интерес. Благодаря этим успехам многослойные машины, подобные персептрону, стали одной из главных моделей нейронных сетей, используемых в настоящее время.

Персептрон для двух классов

В самой простой форме обучение персептрона заключается в построении линейной дискриминантной функции, осуществляющей дихотомию (т. е. деление на две части) двух линейно разделимых обучающих выборок. Рис. 12.14(*а*) схематически показывает модель персептрона в случае двух классов образов. Выходной сигнал (реакция) этого элементарного устройства базируется на взвешенной сумме его входных сигналов, которая имеет вид

$$d(\mathbf{x}) = \sum_{i=1}^{n} w_i x_i + w_{n+1} \tag{12.2-29}$$

и является линейной дискриминантной функцией по отношению к компонентам вектора признаков. Коэффициенты w_i, $i = 1, 2, ..., n, n+1$, называемые *весами*, изменяют (масштабируют) входные сигналы перед тем, как они суммируются и подаются на пороговое устройство. В этом смысле веса аналогичны синапсам в нервной системе человека. Функцию, которая отображает результат суммирования в конечный выходной сигнал устройства, иногда называют *активирующей функцией*.

Если $d(\mathbf{x}) > 0$, пороговое устройство устанавливает на выходе персептрона сигнал +1, указывающий, что объект $\mathbf{x}$ опознан как принадлежащий классу ω_1; при $d(\mathbf{x}) < 0$ на выходе устанавливается сигнал –1. Такой режим работы согласуется с замечанием, сделанным ранее при обсуждении уравнения (12.2-2), об использовании единой дискриминантной функции в случае двух классов. Если $d(\mathbf{x}) = 0$, то объект $\mathbf{x}$ лежит на разделяющей поверхности между двумя классами, что является условием неопределенности. Уравнение разделяющей

поверхности, реализуемой персептроном, получается приравниванием нулю выражения (12.2-29)

$$d(\mathbf{x}) = \sum_{i=1}^{n} w_i x_i + w_{n+1} = 0 \tag{12.2-30}$$

или

$$w_1 x_1 + w_2 x_2 + \ldots + w_n x_n + w_{n+1} = 0, \tag{12.2-31}$$

что представляет собой уравнение гиперплоскости в n-мерном пространстве признаков. С геометрической точки зрения первые n коэффициентов задают направление гиперплоскости, а последний, свободный член w_{n+1} пропорционален расстоянию от начала координат до гиперплоскости в перпендикулярном направлении. Следовательно, при $w_{n+1} = 0$ разделяющая гиперплоскость прохо-

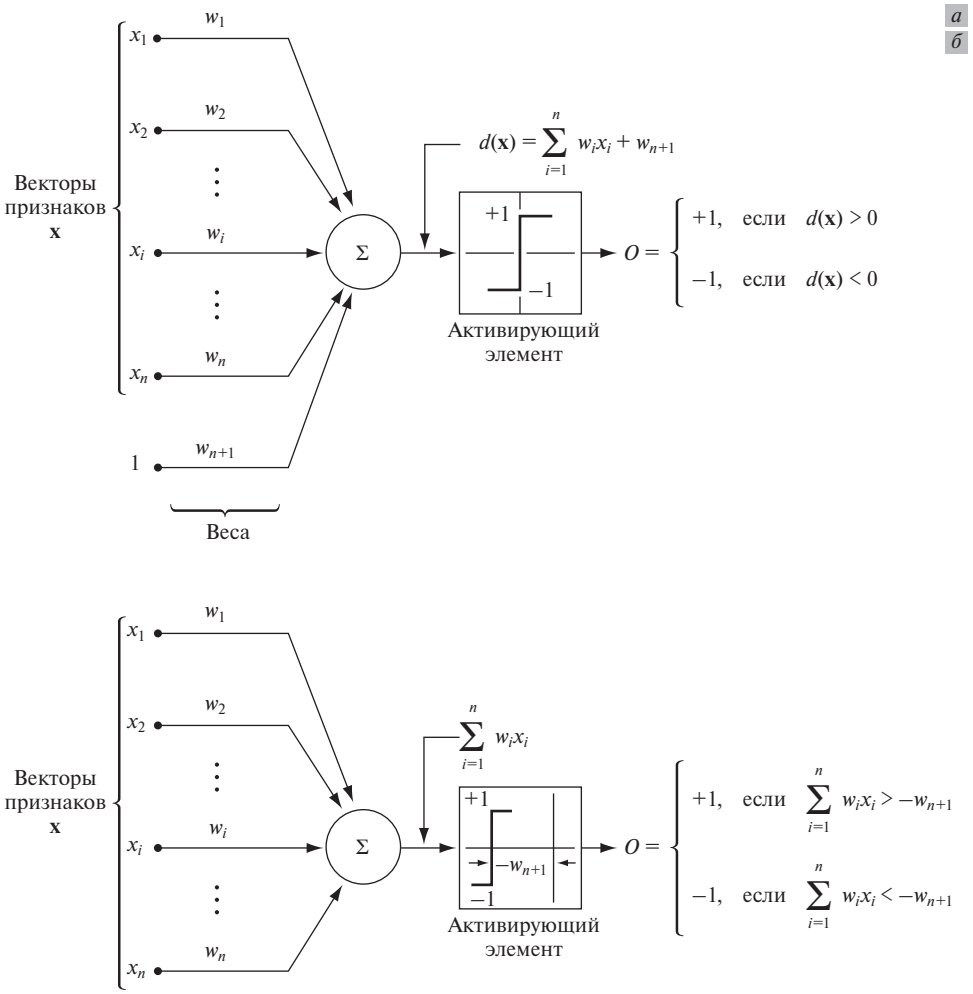


Рис. 12.14. Два эквивалентных представления модели персептрона для двух классов образов

дит через начало координат в пространстве признаков. Аналогично, если $w_j = 0$, гиперплоскость проходит параллельно координатной оси x_j.

Выходной сигнал порогового устройства на рис. 12.14(*а*) зависит от знака функции $d(\mathbf{x})$. Вместо того чтобы исследовать знак всей функции, можно сравнивать взвешенную сумму n входов в правой части выражения (12.2-29) со свободным членом w_{n+1}; в этом случае выходной сигнал системы формируется в соответствии с законом

$$O = \begin{cases} +1, & \text{если } \sum_{i=1}^{n} w_i x_i > -w_{n+1} \\ -1, & \text{если } \sum_{i=1}^{n} w_i x_i < -w_{n+1}. \end{cases} \qquad (12.2\text{-}32)$$

Такая реализация эквивалентна модели, изображенной на рис. 12.14(*а*), и показана на рис. 12.14(*б*). Единственное отличие состоит в том, что пороговая функция смещается на величину $-w_{n+1}$ и константа больше не присутствует в числе входов сумматора. Мы вернемся к эквивалентности этих двух построений позже в этом разделе — при обсуждении вопросов реализации многослойных нейронных сетей.

При другом часто применяемом построении вектор признаков расширяется посредством добавления к нему еще одной (n+1)-й компоненты, которая всегда равна 1, независимо от класса, к которому принадлежит объект. Иначе говоря, строится расширенный вектор признаков объекта $\mathbf{y}$, такой, что $y_i = x_i$, $i = 1, 2, ..., n$; $y_{n+1} = 1$. Тогда выражение (12.2-29) примет вид

$$d(\mathbf{y}) = \sum_{i=1}^{n+1} w_i y_i = \mathbf{w}^T \mathbf{y}, \qquad (12.2\text{-}33)$$

где $\mathbf{y} = (y_1, y_2, ..., y_n, 1)^T$ — *расширенный вектор признаков*, а $\mathbf{w} = (w_1, w_2, ..., w_n, w_{n+1})^T$ называется *весовым вектором*. Такая запись обычно является более удобной с точки зрения обозначений. Однако какая бы формулировка ни использовалась, главная задача всегда состоит в нахождении вектора $\mathbf{w}$ по данной обучающей выборке образов каждого из двух классов.

Алгоритмы обучения

Рассматриваемые ниже алгоритмы являются типичными представителями многочисленных подходов к обучению персептрона, предложенных за годы исследования данного вопроса.

Линейно разделимые классы

Ниже излагается простой итерационный алгоритм получения весового вектора $\mathbf{w}$, являющегося решением для двух линейно разделимых обучающих выборок. Пусть имеются две обучающие выборки расширенных векторов признаков объектов, принадлежащих классам ω_1 и ω_2 соответственно, и пусть $\mathbf{w}(1)$ — некоторый начальный весовой вектор, выбираемый произвольно. На k-м шаге итерации осуществляются следующие действия. Если $\mathbf{y}(k) \in \omega_1$ и $\mathbf{w}^T(k)\mathbf{y}(k) \le 0$, то $\mathbf{w}(k)$ заменяется на

$$\mathbf{w}(k+1) = \mathbf{w}(k) + c\mathbf{y}(k), \qquad (12.2\text{-}34)$$

где c — положительный коэффициент коррекции. Наоборот, если $\mathbf{y}(k) \in \omega_2$ и $\mathbf{w}^T(k)\mathbf{y}(k) \geq 0$, то $\mathbf{w}(k)$ заменяется на

$$\mathbf{w}(k+1) = \mathbf{w}(k) - c\mathbf{y}(k). \tag{12.2-35}$$

В остальных случаях $\mathbf{w}(k)$ остается неизменным:

$$\mathbf{w}(k+1) = \mathbf{w}(k). \tag{12.2-36}$$

Данный алгоритм вносит изменения в вектор $\mathbf{w}$ только в тех случаях, когда на k-м итерационном шаге обработки обучающей последовательности очередной рассматриваемый объект классифицируется ошибочно. Корректирующий коэффициент c считается положительным и в данном случае постоянным. Такой алгоритм иногда называют *правилом постоянного коэффициента коррекции*.

Сходимость алгоритма наступает, когда обучающие выборки обоих классов целиком проходят через последовательность итераций без единой ошибки. Алгоритм с постоянным коэффициентом коррекции сходится за конечное число шагов, если две используемые обучающие выборки являются линейно разделимыми. Доказательство этого результата, который называют *теоремой о сходимости персептрона*, можно найти в книгах [Duda, Hart, Stork, 2001], [Tou, Gonzalez, 1974] и [Nilsson, 1965].

Пример 12.5. Иллюстрация алгоритма обучения персептрона.

■ Рассмотрим две обучающие выборки, представленные на рис. 12.15(*а*), каждая из которых состоит из двух образов. Описанный алгоритм обучения должен завершиться успешно, так как эти две выборки линейно разделимы. Прежде чем применять алгоритм, выполним расширение образов, в результате чего получаем обучающие выборки $\{\mathbf{y}(1), \mathbf{y}(2)\} = \{(0, 0, 1)^T, (0, 1, 1)^T\}$ для класса ω_1 и $\{\mathbf{y}(3), \mathbf{y}(4)\} = \{(1, 0, 1)^T, (1, 1, 1)^T\}$ для класса ω_2. Полагая $c = 1$ и $\mathbf{w}(1) = \mathbf{0}$, будем предъявлять образы в порядке нижеприведенной последовательности шагов:

$$\mathbf{w}^T(1)\mathbf{y}(1) = \begin{bmatrix}0,0,0\end{bmatrix}\begin{bmatrix}0\\0\\1\end{bmatrix} = 0 \quad \Rightarrow \quad \mathbf{w}(2) = \mathbf{w}(1) + \mathbf{y}(1) = \begin{bmatrix}0\\0\\1\end{bmatrix},$$

$$\mathbf{w}^T(2)\mathbf{y}(2) = \begin{bmatrix}0,0,1\end{bmatrix}\begin{bmatrix}0\\1\\1\end{bmatrix} = 1 \quad \Rightarrow \quad \mathbf{w}(3) = \mathbf{w}(2) = \begin{bmatrix}0\\0\\1\end{bmatrix},$$

$$\mathbf{w}^T(3)\mathbf{y}(3) = \begin{bmatrix}0,0,1\end{bmatrix}\begin{bmatrix}1\\0\\1\end{bmatrix} = 1 \quad \Rightarrow \quad \mathbf{w}(4) = \mathbf{w}(3) - \mathbf{y}(3) = \begin{bmatrix}-1\\0\\0\end{bmatrix},$$

$$\mathbf{w}^T(4)\mathbf{y}(4) = \begin{bmatrix}-1,0,0\end{bmatrix}\begin{bmatrix}1\\1\\1\end{bmatrix} = -1 \quad \Rightarrow \quad \mathbf{w}(5) = \mathbf{w}(4) = \begin{bmatrix}-1\\0\\0\end{bmatrix}.$$

По причине ошибок классификации на первом и третьем шагах в весовой вектор были внесены изменения согласно рекуррентным соотношениям (12.2-34) и (12.2-35). Поскольку решение считается полученным только в том случае,

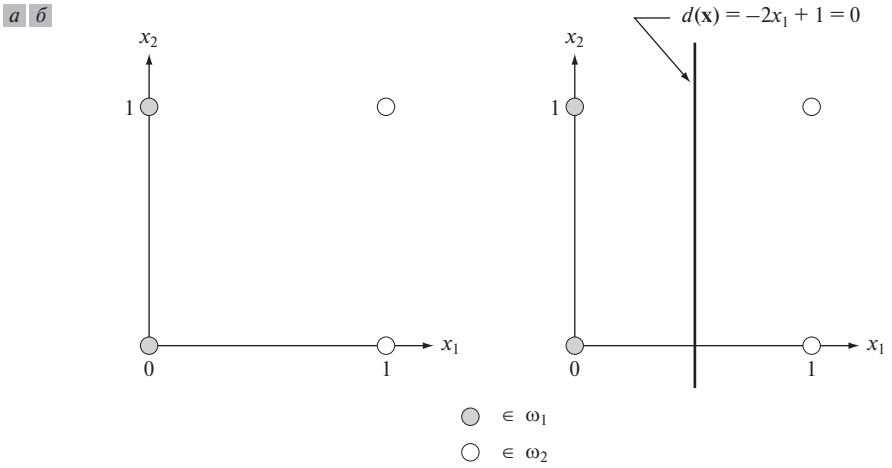


Рис. 12.15. (*а*) Образы из двух классов (серые и белые точки). (*б*) Разделяющая поверхность, построенная в результате обучения

если при предъявлении обученному алгоритму всей выборки целиком ошибки классификации не возникают, то выборку необходимо предъявить снова. Процесс обучения продолжается вышеописанным способом, считая $\mathbf{y}(5) = \mathbf{y}(1)$, $\mathbf{y}(6) = \mathbf{y}(2)$, $\mathbf{y}(7) = \mathbf{y}(3)$, $\mathbf{y}(8) = \mathbf{y}(4)$ и так далее. Сходимость достигается при $k = 14$; полученным решением является весовой вектор $\mathbf{w}(14) = (-2, 0, 1)^T$. Соответствующая дискриминантная функция задается выражением $d(\mathbf{y}) = -2y_1 + 1$. Полагая $x_i = y_i$, возвращаемся к исходному пространству признаков, где дискриминантная функция приобретает вид $d(\mathbf{x}) = -2x_1 + 1$. Приравнивая дискриминантную функцию к нулю, получаем уравнение разделяющей поверхности, показанной на рис. 12.15(*б*). ■

Неразделимые классы

На практике линейно разделимые классы являются скорее редким исключением, чем правилом. Поэтому в 60—70-х годах значительные усилия исследователей были направлены на разработку методов, предназначенных для работы с неразделимыми классами объектов. С успехами последних продвижений в области обучения нейронных сетей многие из таких методов для неразделимых классов стали представлять лишь исторический интерес; однако один из ранних методов имеет непосредственное отношение к теме обсуждения: это дельта-правило в его первоначальной формулировке. Данный метод обучения персептронов, известный как метод Уидроу—Хоффа или как *дельта-правило наименьшего среднего квадрата*, на каждом шаге обучения минимизирует ошибку между фактической и желаемой реакциями.

Рассмотрим целевую функцию

$$J(\mathbf{w}) = \frac{1}{2}\left(r - \mathbf{w}^T\mathbf{y}\right)^2, \tag{12.2-37}$$

где r — желаемая реакция, т. е. $r = +1$, если $\mathbf{y}$ (расширенный вектор признаков объекта из обучающей выборки) принадлежит классу ω_1, и $r = -1$, если $\mathbf{y}$ принадлежит классу ω_2. Задача состоит в том, чтобы путем последовательных при-

ращений корректировать весовой вектор $\mathbf{w}$ в обратном направлении по отношению к вектору градиента функции $J(\mathbf{w})$, чтобы найти минимум этой функции, который достигается при $r = \mathbf{w}^T\mathbf{y}$; т. е. минимум соответствует безошибочной классификации. Если обозначить через $\mathbf{w}(k)$ весовой вектор на k-м шаге итерации, то в обобщенном виде можно записать следующий алгоритм градиентного спуска:

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \alpha\left[\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}}\right]_{\mathbf{w}=\mathbf{w}(k)}, \tag{12.2-38}$$

где $\mathbf{w}(k+1)$ — новое значение вектора $\mathbf{w}$, а параметр $\alpha > 0$ задает коэффициент коррекции. Согласно (12.2-37) имеем

$$\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = -(r - \mathbf{w}^T\mathbf{y})\mathbf{y}. \tag{12.2-39}$$

Подстановка этого результата в рекуррентное соотношение (12.2-38) дает

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \alpha\left[r(k) - \mathbf{w}^T(k)\mathbf{y}(k)\right]\mathbf{y}(k), \tag{12.2-40}$$

причем начальный весовой вектор $\mathbf{w}(1)$ выбирается произвольно.

По определению, изменение весового вектора («дельта») есть векторная величина

$$\Delta\mathbf{w} = \mathbf{w}(k+1) - \mathbf{w}(k), \tag{12.2-41}$$

и соотношение (12.2-40) можно переписать в форме *алгоритма дельта-коррекции:*

$$\Delta\mathbf{w} = \alpha e(k)\mathbf{y}(k), \tag{12.2-42}$$

где

$$e(k) = r(k) - \mathbf{w}^T(k)\mathbf{y}(k) \tag{12.2-43}$$

есть величина допущенной ошибки при использовании весового вектора $\mathbf{w}(k)$ для распознавания объекта $\mathbf{y}(k)$.

Равенство (12.2-43) указывает величину ошибки для весового вектора $\mathbf{w}(k)$. Если мы заменим его на $\mathbf{w}(k+1)$, оставляя объект тем же самым, ошибка станет равной

$$e(k) = r(k) - \mathbf{w}^T(k+1)\mathbf{y}(k). \tag{12.2-44}$$

Тогда величина изменения ошибки составит

$$\begin{aligned}\Delta e(k) &= \left[r(k) - \mathbf{w}^T(k+1)\mathbf{y}(k)\right] - \left[r(k) - \mathbf{w}^T(k)\mathbf{y}(k)\right] = \\ &= -\left[\mathbf{w}^T(k+1) - \mathbf{w}^T(k)\right]\mathbf{y}(k) = -\Delta\mathbf{w}^T\mathbf{y}(k).\end{aligned} \tag{12.2-45}$$

Но $\Delta\mathbf{w} = \alpha e(k)\mathbf{y}(k)$, поэтому

$$\Delta e = -\alpha e(k)\mathbf{y}^T(k)\mathbf{y}(k) = -\alpha e(k)\|\mathbf{y}(k)\|^2. \tag{12.2-46}$$

Следовательно, при изменении весового вектора $\mathbf{w}(k)$ происходит уменьшение ошибки с коэффициентом $\alpha\|\mathbf{y}(k)\|^2$. С предъявлением следующего образа начнется новый цикл адаптации, в котором следующая ошибка уменьшится с коэффициентом $\alpha\|\mathbf{y}(k+1)\|^2$ и т. д.

От выбора параметра α зависит устойчивость и скорость сходимости алгоритма [Widrow, Stearns, 1985]. Для устойчивости алгоритма необходимо, чтобы $0 < \alpha < 2$. На практике используется интервал значений $0{,}1 < \alpha < 1{,}0$. Хотя доказательство этого факта здесь не приводится, алгоритм, определяемый соотношениями (12.2-40) или (12.2-42) и (12.2-43), сходится к решению, минимизирующему средний квадрат ошибки на образах обучающей выборки. Если классы линейно разделимы, то вышеописанный алгоритм не обязательно строит решение в виде разделяющей гиперплоскости. Иначе говоря, решение, минимизирующее средний квадрат ошибки, необязательно является решением в смысле теоремы о сходимости персептрона. Такая неопределенность — цена за возможность применения алгоритма, который в данной конкретной постановке сходится и для разделимых, и для неразделимых классов.

Рассматривавшиеся до сих пор два алгоритма обучения персептрона могут быть распространены на случаи, когда имеется более двух классов, а также используются нелинейные дискриминантные функции. Но, учитывая сделанные выше замечания исторического характера, нет особого смысла излагать здесь алгоритмы обучения для случая многих классов. Вместо этого мы рассмотрим обучение для нескольких классов в контексте нейронных сетей.

Многослойные нейронные сети без обратной связи

В этом разделе мы сосредоточимся на построении дискриминантных функций в задачах распознавания образов из нескольких классов. Эти функции не зависят от того, разделимы классы или нет, и основаны на архитектурах, состоящих из слоев вычислительных элементов типа персептрона.

Базовая архитектура

Архитектура рассматриваемой модели нейронной сети представлена на рис. 12.16. Она состоит из слоев, в которых находятся идентичные по структуре вычислительные узлы (нейроны), организованные таким образом, что выход каждого нейрона одного слоя соединяется со входом каждого нейрона следующего слоя. Число нейронов в первом слое, называемом слоем A, равно N_A; оно часто выбирается равным размерности входных векторов-образов: $N_A = n$. Число нейронов выходного слоя, называемого слоем Q, обозначается N_Q. Это число N_Q равно W — числу классов, образы которых данная нейронная сеть обучена распознавать. Сеть распознает объект с вектором признаков $\mathbf{x}$ как принадлежащий классу ω_i, если на i-м выходе сети присутствует «высокий» уровень, а на остальных выходах — «низкий», что разъясняется в дальнейшем.

Как показано на увеличенном фрагменте рис. 12.16, каждый нейрон имеет тот же вид, что и рассмотренная ранее модель персептрона (см. рис. 12.14), с тем исключением, что вместо активирующей функции с разрывным пороговым преобразованием используется непрерывная S-образная функция со «сглаженным порогом», поскольку для разработки обучающего правила необходима дифференцируемость вдоль всех путей в нейронной сети. Требуемой дифференцируемостью обладает следующая S-образная функция активации:

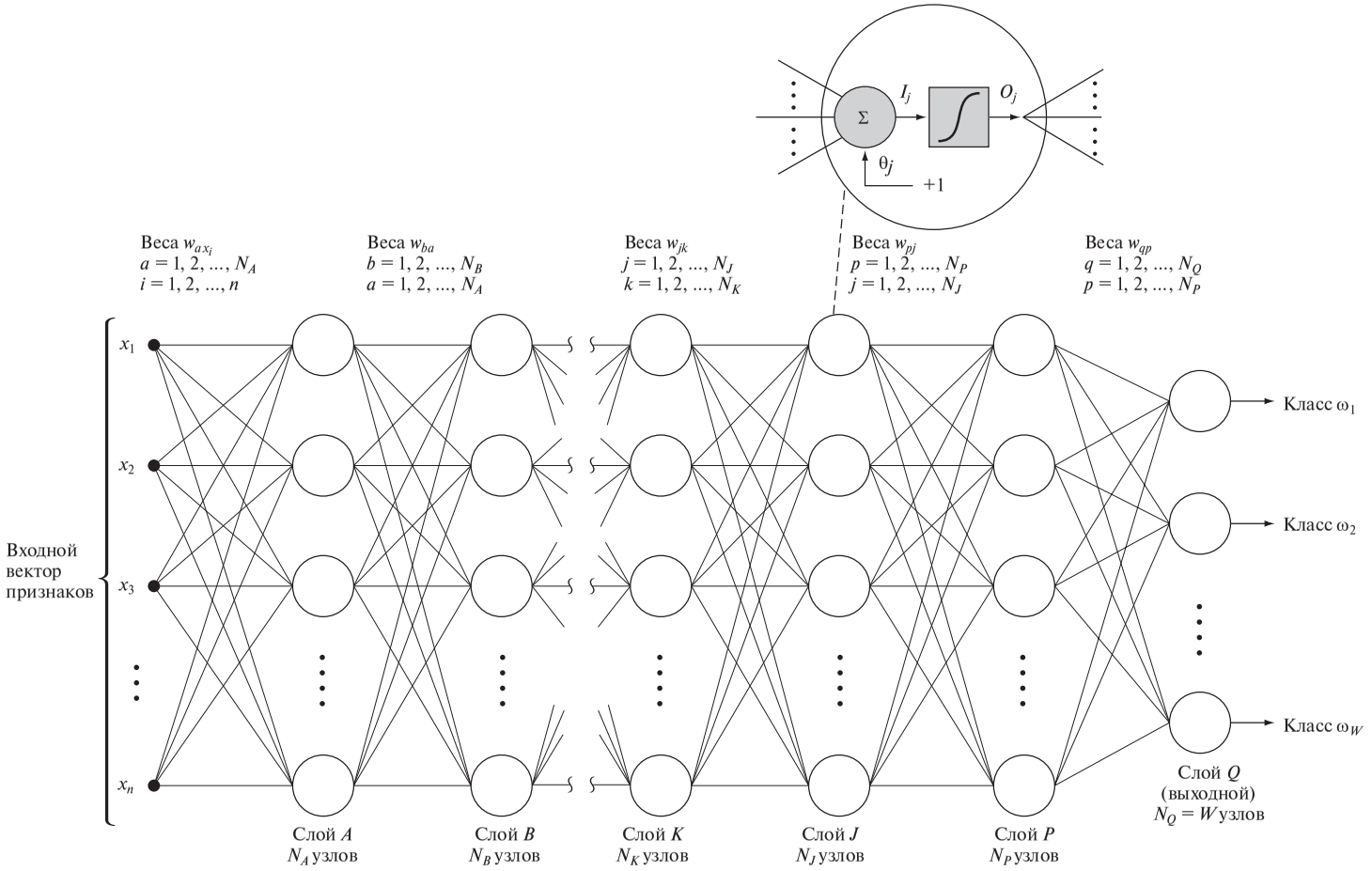


Рис. 12.16. Модель многослойной нейронной сети без обратной связи. В увеличенном фрагменте показана базовая структура, которую имеет каждый элементарный нейрон сети. Величина смещения θ_j рассматривается как еще одна весовая компонента

$$h_j(I_j) = \frac{1}{1 + e^{-(I_j - \theta_j)/\theta_0}}, \qquad (12.2\text{-}47)$$

где I_j, $j = 1, 2, ..., N_J$ — значение на входе активирующего элемента каждого узла слоя J нейронной сети, θ_j — величина смещения, а параметр θ_0 определяет крутизну S-образной функции.

На рис. 12.17 приведен график функции (12.2-47), а также (пунктирными линиями) показаны «высокий» и «низкий» уровни выходного сигнала каждого узла. Итак, при использовании данной функции значение уровня на выходе узла будет высоким при $I_j > \theta_j$ и низким при $I_j < \theta_j$.[6] Как видно из рис. 12.17, S-образная функция активации всюду положительна и достигает своих предельных значений 0 и 1, когда значение на входе активирующего элемента равно минус или плюс бесконечности. По этой причине в качестве порогов нижнего и верхнего уровней сигнала на выходе нейронов в модели на рис. 12.16 выбираются значения вблизи 0 и 1, скажем, 0,05 и 0,95. В принципе, для разных слоев нейронной сети или даже для разных узлов в одном слое могут применяться активирующие функции различного вида, однако на практике обычно во всей сети используют функции активации одинакового вида.

Показанное на рис. 12.17 смещение θ_j аналогично весовому коэффициенту w_{n+1} рассмотренного ранее персептрона (см. рис. 12.14(*а*)). Эта функция со смещенным порогом может быть реализована по схеме, аналогичной рис. 12.14(*а*), при этом смещение θ_j рассматривается как дополнительный коэффициент, на который умножается постоянный единичный входной сигнал, одинаковый для всех узлов сети. Следуя преобладающей в литературе системе обозначений, мы не показываем отдельный постоянный входной сигнал +1 для всех узлов сети на рис. 12.16, а вместо этого считаем этот входной сигнал и модифицирующий его вес θ_j составной частью каждого узла нейронной сети. Как видно на увеличенном фрагменте рис. 12.16, для каждого из N_j узлов в слое J имеется по одному такому коэффициенту.

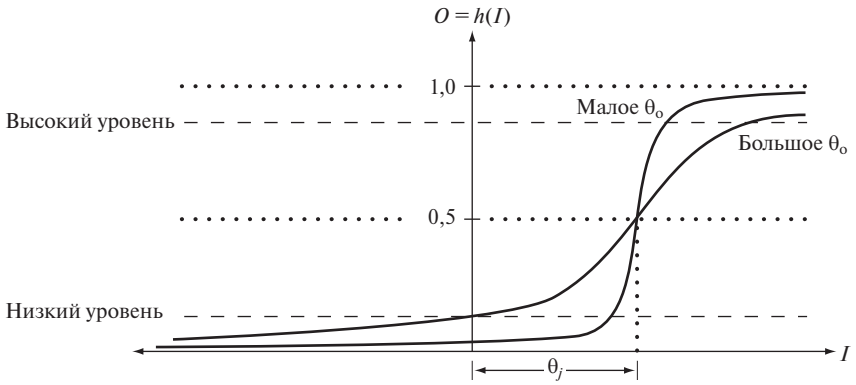


Рис. 12.17. *S*-образная функция активации, задаваемая выражением (12.2-47)

[6] Точнее, высоким при $I_j > \theta_j + \tau(\theta_0)$ и низким при $I_j < \theta_j - \tau(\theta_0)$, где $[\theta_j + \tau(\theta_0)]$ и $[\theta_j - \tau(\theta_0)]$ — те значения I, при которых график функции (12.2 47) пересекает соответственно линию «высокого» или «низкого» уровня выходного сигнала. — *Прим. перев.*

Входом для узла любого слоя сети на рис. 12.16 является взвешенная сумма выходных сигналов всех узлов предыдущего слоя. Пусть слой K, предшествующий слою J сети (заметим, что на рис. 12.16 не предполагается никакой алфавитной упорядоченности), создает на входе активирующего элемента каждого узла слоя J сигнал, обозначаемый I_j:

$$I_j = \sum_{k=1}^{N_K} w_{jk} O_k \qquad (12.2\text{-}48)$$

для $j = 1, 2,..., N_J$, где N_J — число узлов в слое J, N_K — число узлов в слое K, а w_{jk} — веса (весовые коэффициенты), модифицирующие выходные сигналы O_k узлов слоя K на входе узлов слоя J. Эти выходные сигналы слоя K имеют значения

$$O_k = h_k(I_k), \quad k = 1,2,\ldots,N_K. \qquad (12.2\text{-}49)$$

Важно четко понимать систему индексных обозначений, фигурирующих в выражении (12.2-48), поскольку мы будем пользоваться ей на протяжении всей оставшейся части данного раздела. Прежде всего отметим, что I_j, $j = 1, 2,..., N_J$, обозначает сигнал на входе *активирующего элемента* j-го узла слоя J, т. е. I_1 есть сигнал на входе активирующего элемента первого (верхнего) узла слоя J, I_2 — сигнал на входе активирующего элемента второго узла слоя J и т. д. У каждого узла в слое J имеется N_K входов, но *каждый* отдельный вход умножается на свой собственный весовой коэффициент. Так, N_K входов первого узла в слое J взвешиваются с коэффициентами w_{1k}, $k = 1, 2,..., N_K$; входы второго узла имеют веса w_{2k}, $k = 1, 2,..., N_K$ и т. д. Следовательно, для преобразования выходных сигналов слоя K на входе слоя J требуется в общей сложности $N_J \times N_K$ коэффициентов. Чтобы полностью описать узлы в слое J, необходимы еще дополнительные N_J коэффициентов — смещений θ_j.

Подстановка выражений (12.2-48) в уравнение (12.2-47) дает

$$h_j(I_j) = \frac{1}{1 + e^{-\left(\sum_{k=1}^{N_k} w_{jk} O_k - \theta_j\right) \Big/ \theta_0}}. \qquad (12.2\text{-}50)$$

Активирующую функцию данного вида мы будем использовать на протяжении оставшейся части этого раздела.

Адаптация нейронов выходного слоя в ходе обучения не представляет трудностей, поскольку желаемый выходной сигнал всех этих узлов известен. Основная проблема при обучении многослойной сети состоит в настройке весов так называемых *скрытых слоев*, т. е. всех, кроме выходного.

Обучение путем обратного распространения ошибки

Вначале сосредоточим внимание на выходном слое. Суммарный квадрат ошибки между желаемыми реакциями r_q узлов выходного слоя Q и соответствующими фактическими реакциями O_q равен

$$E_Q = \frac{1}{2} \sum_{q=1}^{N_Q} (r_q - O_q)^2, \qquad (12.2\text{-}51)$$

где N_Q — число узлов выходного слоя Q, а полусумма взята для удобства последующего дифференцирования. Наша цель состоит в разработке обучающего

правила, подобного дельта-правилу, которое бы позволяло корректировать веса в каждом слое таким образом, чтобы функция ошибки, заданная формулой (12.2-51), стремилась к минимальному значению. Для получения такого результата, как и ранее, будем корректировать веса пропорционально частной производной функции ошибки по этим весам. Другими словами,

$$\Delta w_{qp} = -\alpha \frac{\partial E_Q}{\partial w_{qp}}, \tag{12.2-52}$$

где слой P предшествует слою Q и α — положительный коэффициент коррекции. Таким образом, определение Δw_{qp} аналогично определению $\Delta \mathbf{w}$ в (12.2-42).

Функция ошибки E_Q зависит от выходных сигналов O_q, а они, в свою очередь, являются функциями входных сигналов I_q. Используя правило дифференцирования сложной функции, вычисляем частную производную E_Q следующим образом:

$$\frac{\partial E_Q}{\partial w_{qp}} = \frac{\partial E_Q}{\partial I_q} \frac{\partial I_q}{\partial w_{qp}}. \tag{12.2-53}$$

С учетом (12.2-48)

$$\frac{\partial I_q}{\partial w_{qp}} = \frac{\partial}{\partial w_{qp}} \sum_{p=1}^{N_P} w_{qp} O_p = O_p. \tag{12.2-54}$$

Подстановка выражений (12.2-53) и (12.2-54) в (12.2-52) дает

$$\Delta w_{qp} = -\alpha \frac{\partial E_Q}{\partial I_q} O_p = \alpha \delta_q O_p, \tag{12.2-55}$$

где

$$\delta_q = -\frac{\partial E_Q}{\partial I_q}. \tag{12.2-56}$$

Чтобы вычислить $\partial E_Q / \partial I_q$, воспользуемся формулой производной сложной функции, выражая частную производную через скорость изменения E_Q относительно O_q и скорость изменения O_q относительно I_q. Иначе говоря,

$$\delta_q = -\frac{\partial E_Q}{\partial I_q} = -\frac{\partial E_Q}{\partial O_q} \frac{\partial O_Q}{\partial I_q}. \tag{12.2-57}$$

Из (12.2-51) следует, что

$$\frac{\partial E_Q}{\partial O_q} = -(r_q - O_q), \tag{12.2-58}$$

а согласно (12.2-49) имеем

$$\frac{\partial O_q}{\partial I_q} = \frac{\partial}{\partial I_q} h_q(I_q) = h_q'(I_q). \tag{12.2-59}$$

Подставляя (12.2-58) и (12.2-59) в (12.2-57), получаем

$$\delta_q = (r_q - O_q) h_q'(I_q), \tag{12.2-60}$$

т. е. выражение, пропорциональное величине ошибки ($r_q - O_q$). Подставляя выражения (12.2-56)—(12.2-58) в (12.2-55), окончательно получаем

$$\Delta w_{qp} = \alpha(r_q - O_q)h_q'(I_q)O_p = \alpha\delta_q O_p. \qquad (12.2\text{-}61)$$

После того, как задана функция $h_q(I_q)$, все члены в (12.2-61) или становятся известными, или могут быть получены из наблюдения за сетью. Другими словами, предъявляя любой обучающий образ на вход сети, мы знаем, какой должна быть желаемая реакция r_q каждого выходного узла. Значения O_q на каждом выходном узле мы можем наблюдать, равно как и I_q (сигналы на входах активирующих элементов слоя Q) и O_p (выходные сигналы узлов слоя P). Таким образом, мы теперь знаем, как откорректировать веса на связях между предпоследним и последним слоями нейронной сети.

Продолжим рассмотрение, двигаясь назад от выходного слоя. Проанализируем теперь, что происходит в слое P. Действуя точно так же, как описывалось выше, получаем

$$\Delta w_{pj} = \alpha(r_p - O_p)h_p'(I_p)O_j = \alpha\delta_p O_j, \qquad (12.2\text{-}62)$$

где составляющая ошибки имеет вид

$$\delta_p = (r_p - O_p)h_p'(I_p). \qquad (12.2\text{-}63)$$

За исключением r_p, все остальные члены в выражениях (12.2-62) и (12.2-63) либо известны, либо могут быть получены из наблюдения за сетью. Член r_p бессодержателен во внутреннем слое сети, так как мы не знаем, какова должна быть реакция промежуточных узлов по отношению к принадлежности образов к тому или иному классу. Мы можем формулировать вид желаемой реакции r только на выходах сети, где происходит окончательная классификация образов. Если бы мы имели эту информацию для внутренних узлов, то дальнейшие слои стали бы ненужными. Таким образом, мы хотим найти способ переформулирования δ_p на основании величин, которые или известны, или могут быть получены из наблюдения за сетью.

Возвращаясь к равенству (12.2-57), запишем составляющую ошибки для слоя P в виде

$$\delta_p = -\frac{\partial E_p}{\partial I_p} = -\frac{\partial E_p}{\partial O_p}\frac{\partial O_p}{\partial I_p}. \qquad (12.2\text{-}64)$$

Член $\partial O_p/\partial I_p$ не доставляет трудностей. Как и ранее, он равен

$$\frac{\partial O_p}{\partial I_p} = \frac{\partial h_p(I_p)}{\partial I_p} = h_p'(I_p), \qquad (12.2\text{-}65)$$

т. е. известен при заданной функции h_p, поскольку I_p можно наблюдать. Членом, который порождает сигнал r_p, является производная $\partial E_p/\partial O_p$, поэтому ее необходимо выразить таким образом, чтобы она не содержала r_p. Применяя правило дифференцирования сложной функции, перепишем эту производную следующим образом:

$$-\frac{\partial E_p}{\partial O_p} = -\sum_{q=1}^{N_Q}\frac{\partial E_p}{\partial I_q}\frac{\partial I_q}{\partial O_p} = \sum_{q=1}^{N_Q}\left(-\frac{\partial E_p}{\partial I_q}\right)\frac{\partial}{\partial O_p}\sum_{p=1}^{N_P} w_{qp}O_p = \sum_{q=1}^{N_Q}\left(-\frac{\partial E_p}{\partial I_q}\right)w_{qp} = \sum_{q=1}^{N_Q}\delta_q w_{qp}, \qquad (12.2\text{-}66)$$

где последний шаг следует из (12.2-56). Подстановка (12.2-65) и (12.2-66) в (12.2-64) дает желаемое выражение для δ_p:

$$\delta_p = h_p'(I_p)\sum_{q=1}^{N_Q}\delta_q w_{qp}\,. \tag{12.2-67}$$

Теперь параметр δ_p можно вычислить, так как все его члены известны. Итак, формулы (12.2-62) и (12.2-67) образуют окончательное обучающее правило для слоя *P*. Выражение (12.2-67) важно тем, что в нем δ_p вычисляется через величины δ_q и w_{qp}, которые были найдены в слое, следующем за *P*. После вычисления составляющей ошибки и весов для слоя *P* эти величины можно будет использовать аналогичным образом для нахождения ошибки и весов для слоя, непосредственно предшествующего слою *P*. Другими словами, мы нашли способ распространять ошибку назад по сети, начиная с ошибки в выходном слое.

Можно подытожить и обобщить изложенную процедуру обучения следующим образом. Для любого слоя *J*, которому непосредственно предшествует слой *K*, вычисляются веса w_{jk}, модифицирующие связи между этими слоями; для этого используется выражение

$$\Delta w_{jk} = \alpha\delta_j O_k\,. \tag{12.2-68}$$

Если слой *J* является выходным, то δ_j вычисляется как

$$\delta_j = (r_j - O_j)h_j'(I_j)\,. \tag{12.2-69}$$

Если слой *J* является внутренним и следующим (в сторону выхода) является слой *P*, тогда δ_j задается формулой

$$\delta_j = h_j'(I_j)\sum_{p=1}^{N_P}\delta_p w_{jp} \tag{12.2-70}$$

для $j = 1, 2,..., N_J$. Применяя активирующую функцию вида (12.2-50) с параметром $\theta_0 = 1$, получаем

$$h_j'(I_j) = O_j(1-O_j)\,, \tag{12.2-71}$$

и тогда выражения (12.2-69) и (12.2-70) приобретают следующий достаточно элегантный вид:

$$\delta_j = (r_j - O_j)O_j(1-O_j) \tag{12.2-72}$$

для выходного слоя и

$$\delta_j = O_j(1-O_j)\sum_{p=1}^{N_P}\delta_p w_{jp} \tag{12.2-73}$$

для внутренних слоев. В обоих выражениях (12.2-72) и (12.2-73) $j = 1, 2,..., N_J$.

Формулы (12.2-68)—(12.2-70) вместе составляют обобщенное дельта-правило обучения многослойной нейронной сети без обратной связи, изображенной на рис. 12.16. Процесс начинается с произвольного набора весов в узлах сети (но не всех одинаковых). После этого применение обобщенного дельта-правила на любом шаге итерации складывается из двух основных этапов. На первом этапе на вход сети предъявляется обучающий вектор признаков, и сигналы распространяются по слоям вплоть до установления сигнала O_j на выходе каждо-

го узла. Затем реакции O_q узлов выходного уровня сравниваются с желаемыми выходными сигналами r_q, в результате чего строятся составляющие ошибок δ_q. На втором этапе осуществляется обратный проход по сети, при котором соответствующие сигналы ошибки передаются в каждый узел, что позволяет нужным образом откорректировать его веса. Эта процедура применяется и к смещениям θ_j, которые, как говорилось выше, рассматриваются в качестве дополнительных весовых коэффициентов, на которые умножается единичный сигнал, подаваемый на вход сумматора каждого узла нейронной сети.

Обычная практика состоит в том, чтобы прослеживать ошибки сети, в том числе ошибки, связанные с конкретными образами. При успешном сеансе обучения величина ошибки сети уменьшается по мере роста числа итераций, и процедура обучения сходится к устойчивому набору весовых векторов, в котором в случае дополнительного обучения наблюдаются лишь небольшие флуктуации. Для выяснения того, что в процессе обучения некоторый образ классифицируется правильно, необходимо проверить, что реакция узла выходного слоя, сопоставляемого тому классу, к которому принадлежит данный образ, имеет высокий уровень, а сигналы остальных узлов выходного слоя имеют низкий уровень, как было определено выше.

После того, как обучение системы закончено, она применяется для классификации неизвестных образов с использованием тех значений параметров, которые были установлены в процессе обучения. В обычном режиме работы все линии обратной связи разрываются. После этого сигналы любого поступающего образа свободно распространяются по всем слоям, и образ классифицируется как принадлежащий классу, который соответствует узлу с высоким уровнем на выходе, при условии, что на выходах остальных узлов уровень сигнала низкий. Если высокий уровень отмечается сразу на нескольких узлах выходного слоя сети или не обнаруживается ни на одном из узлов, то (по выбору разработчика) либо объявляется об отказе от классификации, либо принимается решение отнести объект к тому классу, на выходном узле которого присутствует сигнал максимальной амплитуды.

Пример 12.6. Классификация формы фигуры с помощью нейронной сети.
■ Проиллюстрируем теперь процесс обучения нейронной сети (общий вид которой представлен на рис. 12.16) распознаванию фигур четырех эталонных форм, изображенных на рис. 12.18(*а*), вместе с искаженными вариантами форм, примеры которых показаны на рис. 12.18(*б*).

Векторы признаков объектов строились путем вычисления нормированных сигнатур фигур (см. раздел 11.1.3), после чего от каждой сигнатуры брались 48 отсчетов через равные интервалы. Полученные 48-мерные вектора подавались на вход трехслойной нейронной сети без обратной связи, показанной на рис. 12.19. Число узлов в первом слое сети было выбрано равным 48 — размерности исходного пространства признаков. Четыре нейрона в третьем (выходном) слое соответствуют числу распознаваемых классов, а число нейронов в среднем слое было эвристически выбрано равным 26 (среднее арифметическое числа узлов входного и выходного слоев). Для выбора числа узлов во внутренних слоях нейронной сети не существует известных правил, поэтому выбор обычно производится на основании предшествующего опыта или делается произволь-

а
б

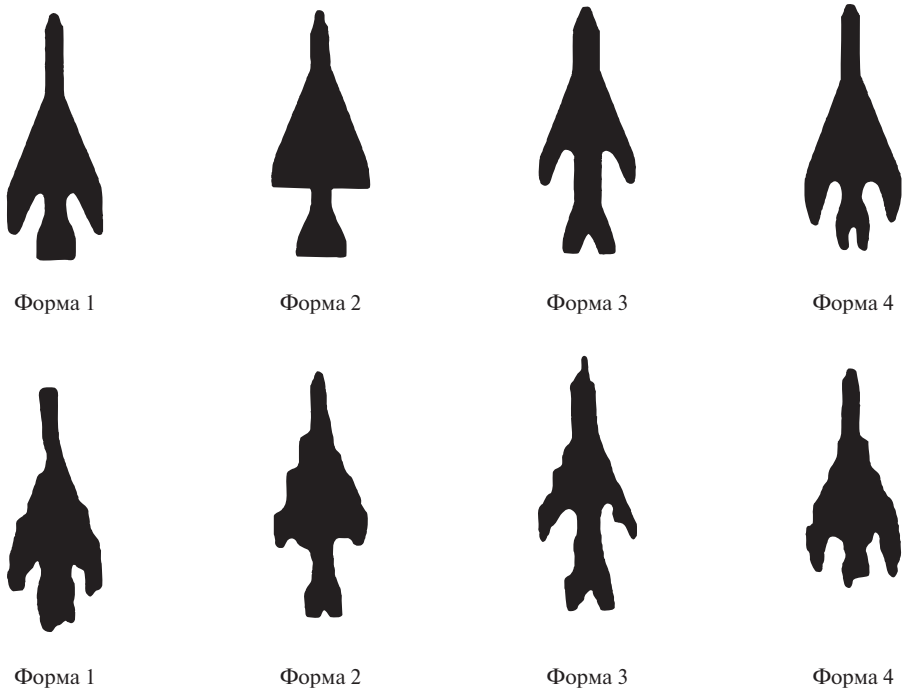


Рис. 12.18. (*а*) Эталонные формы и (*б*) типичные искаженные фигуры, использованные для обучения нейронной сети на рис. 12.19. (Изображения предоставлены д-ром Лалит Гупта, кафедра ECE, Южный университет шт. Иллинойс)

но с последующим уточнением по результатам эксперимента на контрольной выборке. Четыре узла выходного слоя в данном случае представляют четыре класса ω_j, $j = 1, 2, 3, 4$ (сверху вниз соответственно). После того, как зафиксирована структура сети, необходимо выбрать вид активирующих функций для всех узлов во всех слоях. Все эти функции активации были выбраны согласно выражению (12.2-50) при $\theta_0 = 1$, так что в соответствии с проведенным выше рассмотрением применимы формулы (12.2-72) и (12.2-73).

Процесс обучения состоял из двух стадий. На первой стадии весам были присвоены небольшие случайные начальные значения с нулевым средним, после чего проводилось обучение сети на обучающей выборке неискаженных образов, подобных приведенным на рис. 12.18(*а*). В ходе обучения контролировались выходные сигналы узлов последнего слоя. Обучение сети распознаванию фигур всех четырех классов считалось законченным, когда для любого обучающего образа из класса ω_i реакции узлов выходного уровня составляли $O_i \geq 0{,}95$ и $O_q \leq 0{,}05$ для $q = 1, 2, ..., N_Q$; $q \neq i$. Другими словами, для любого образа из класса ω_i соответствующий этому классу выходной узел должен был выдавать сигнал высокого ($\geq 0{,}95$) уровня, при этом выходные сигналы всех прочих узлов выходного слоя должны были иметь низкий ($\leq 0{,}05$) уровень.

На второй стадии обучения использовались искаженные образы, которые генерировались следующим образом. Каждый пиксель границы эталонной фигуры (без искажений) с вероятностью R случайным образом передвигался

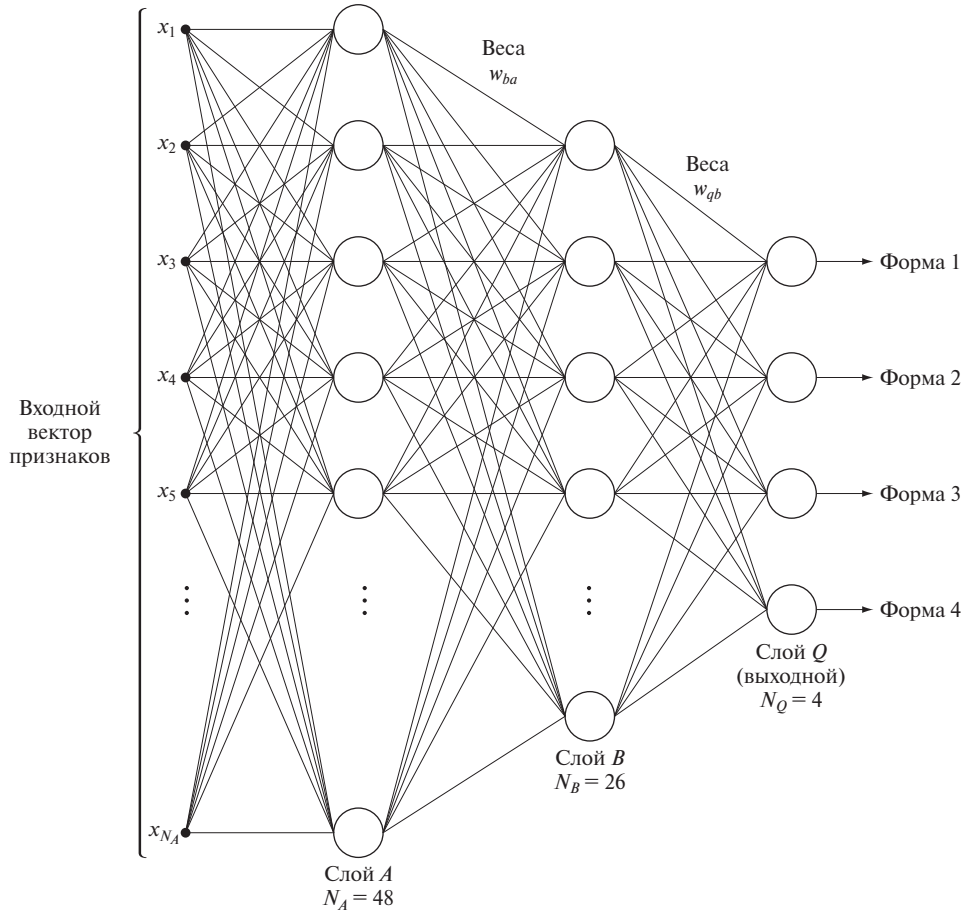


Рис. 12.19. Трехслойная нейронная сеть, использованная для распознавания фигур на рис. 12.18. (Изображение предоставлено д-ром Лалит Гупта, кафедра ECE, Южный университет шт. Иллинойс)

в положение одного из восьми соседних пикселей и с вероятностью $V = 1 - R$ сохранял свои первоначальные координаты. Тем самым уровень искажений возрастал с увеличением значения R, которое можно трактовать как уровень шума сигнатуры. Были построены две выборки зашумленных данных. В первой содержалось по 100 искаженных объектов из каждого класса, полученных при изменении R от 0,1 до 0,6, что составило в общей сложности 400 образов. Этот набор объектов, называемый *контрольной выборкой*, использовался для оценки характеристик системы по результатам обучения.

Для обучения системы на зашумленных данных было сформировано несколько выборок фигур с искажениями. Выборки обозначаются символом R_t, значение которого численно равно R — уровню искажений при генерации обучающих данных. Первая состояла из 10 образцов каждого класса, сгенерированных при $R_t = 0$. В качестве начальных весовых векторов на этой стадии обучения использовались те, которые были найдены на первой стадии обучения (без искажений), и системе давалась возможность провести их коррек-

цию на обучающей последовательности, состоящей из нового набора данных. Поскольку $R_t = 0$ означает отсутствие искажений, такое повторное обучение было попросту продолжением ранее проведенного обучения на эталонах. Затем с окончательными весами, полученными при дообучении, работа нейронной сети испытывалась на контрольной выборке; полученные результаты показаны на рис. 12.20 в форме кривой с пометкой $R_t = 0{,}0$. Вероятность ошибочной классификации оценивалась как отношение числа ошибочно распознанных объектов к общему числу образов в контрольной выборке, что является общепринятой характеристикой определения качества работы нейронной сети.

После этого, начиная с весовых векторов, которые были найдены в результате обучения на данных, построенных при $R_t = 0$, система снова обучалась, но уже на искаженных данных, сгенерированных при $R_t = 0{,}1$, причем качество распознавания (с новыми весами) опять определялось по полной контрольной выборке (кривая $R_t = 0{,}1$ на рис. 12.20). Как видно из графика, качество заметно повысилось. На этом же рисунке приведены результаты, полученные в дальнейшем при повторении такой процедуры дообучения и перепроверки с выборками $R_t = 0{,}2$; 0,3 и 0,4. Как и ожидалось, при правильном обучении системы вероятность ошибочной классификации объектов из контрольной выборки уменьшается по мере роста значения R_t, поскольку обучение системы при этом производится на данных с более высоким уровнем искажений. Единственным исключением на рис. 12.20 является результат, полученный при $R_t = 0{,}4$, причи-

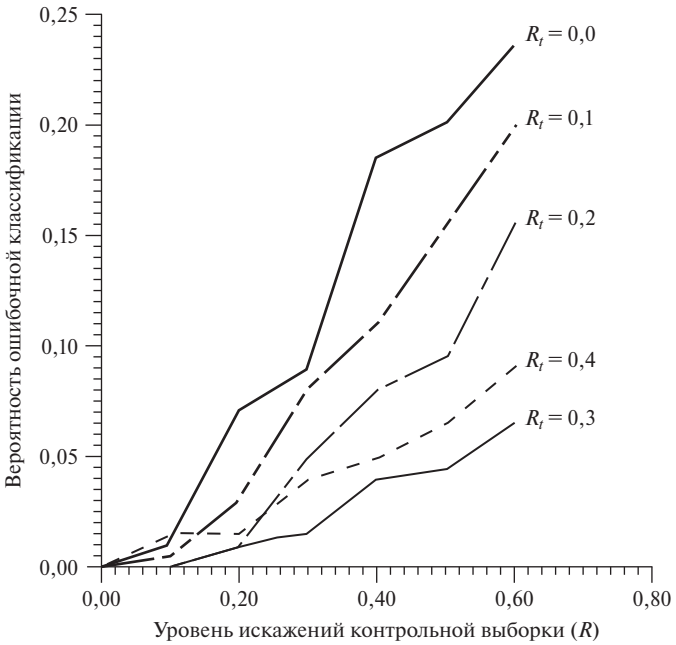


Рис. 12.20. Качество работы нейронной сети в зависимости от уровней искажений обучающей (R_t) и контрольной (R) выборок. (Изображение предоставлено д-ром Лалит Гупта, кафедра ECE, Южный университет шт. Иллинойс)

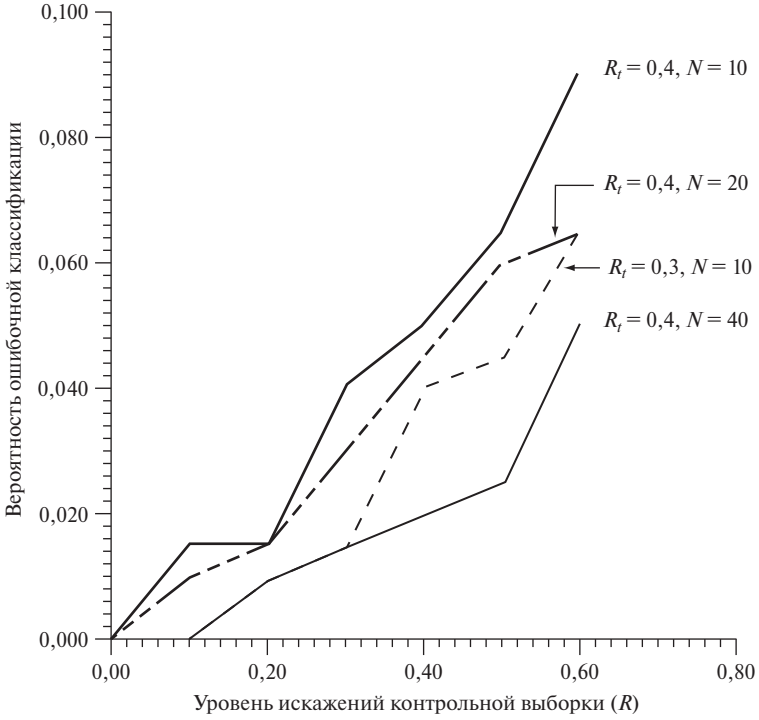


Рис. 12.21. Повышение качества распознавания для $R_t = 0{,}4$ при увеличении объема обучающей выборки (для сравнения показан прежний график при $R_t = 0{,}3$ из рис. 12.20). (Изображение предоставлено д-ром Лалит Гупта, кафедра ECE, Южный университет шт. Иллинойс)

ной чего является малый объем выборки, использованной для обучения системы: при таком числе эталонных образов она оказывается неспособной эффективно адаптироваться к изменениям формы, которые возникают при высоком уровне искажений. Эту гипотезу подтверждают результаты на рис. 12.21, где показано, что при увеличении числа образов в обучающих выборках происходит снижение вероятности ошибочной классификации. Для сравнения на рис. 12.21 также приведен график для $R_t = 0{,}3$ из рис. 12.20.

Приведенные результаты показывают, что трехслойная нейронная сеть способна обучаться распознаванию формы искаженных объектов после умеренного дообучения. Уже после обучения на неискаженных данных ($R_t = 0{,}0$ на рис. 12.20) система достигает уровня правильного распознавания, близкого к 77 %, при проверке на значительно искаженных контрольных данных ($R_t = 0{,}6$ на рис. 12.20). Вероятность правильного распознавания на тех же данных возрастает почти до 99 %, если система дообучается на искаженных данных ($R_t = 0{,}3$ и 0,4). Важно обратить внимание, что обучение системы проводилось путем постепенного повышения ее мощности распознавания с помощью систематического и плавного увеличения уровня искажений. Если природа искажений известна, такой метод идеально подходит для улучшения сходимости и повышения устойчивости нейронной сети в процессе ее обучения. ■

Сложность разделяющих поверхностей

Ранее отмечалось, что однослойный персептрон строит разделяющую поверхность в форме гиперплоскости. Возникает естественный вопрос: а какую форму имеет разделяющая поверхность, реализуемая многослойной нейронной сетью, подобной модели на рис. 12.16? В нижеследующем рассмотрении демонстрируется, что трехслойная сеть способна строить сколь угодно сложные разделяющие поверхности, образованные пересекающимися гиперплоскостями.

Для начала рассмотрим двухслойную сеть с двумя входами, как показано на рис. 12.22(*а*). В случае двух входов все объекты являются двумерными, и, следовательно, каждый узел первого слоя сети реализует прямую в двумерном пространстве. Обозначим 1 и 0 соответственно высокий и низкий уровни выходного сигнала для каждого из этих двух узлов. Будем считать, что выходной сигнал 1 показывает, что соответствующий входной вектор лежит с положительной стороны разделяющей прямой. На единственный узел второго слоя с узлов первого слоя могут поступать следующие комбинации выходных сигналов: (1, 1), (1, 0), (0, 1) и (0, 0). Если определить две области — одну для класса ω_1, лежащую с положительной стороны от обеих прямых, и другую для класса ω_2, расположенную где-то еще, — выходной узел способен относить любой входной объект к одной из этих областей простым выполнением логической операции И. Иначе говоря, реакция выходного узла будет 1 (что указывает на класс ω_1), только когда оба выходных сигнала первого уровня одновременно равны 1. Вышеописанный узел нейронной сети способен выполнять операцию И, если весовые коэффициенты на его входах равны 1, а значение смещения θ_j берется из полуоткрытого интервала (1, 2]. Таким образом, если предположить, что реакции узлов первого слоя могут принимать только значения 0 или 1, то сигнал выходного узла установится на высоком уровне (тем самым указывая на класс ω_1) лишь в том случае, если сумма сигналов от первого слоя будет больше 1. Рис. 12.22(*б*) и (*в*) демонстрируют, что показанная на рис. 12.22(*а*) сеть способна успешно разграничивать два класса образов, не разделяющихся одиночной линейной поверхностью.

Если число узлов первого слоя увеличить до трех, изображенная на рис. 12.22(*а*) сеть будет строить разделяющую поверхность, образованную

а б в

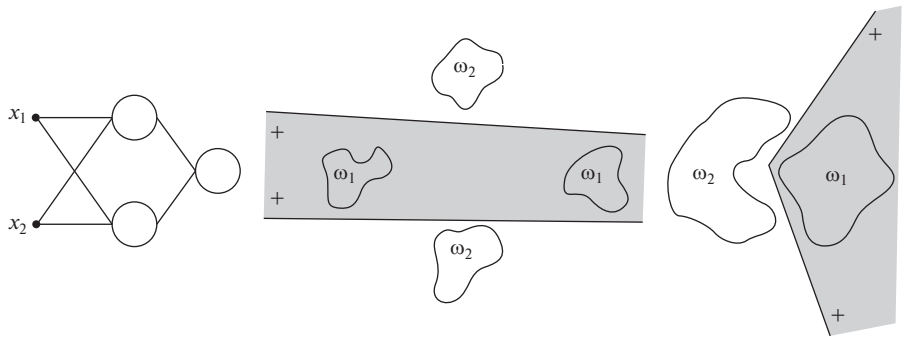


Рис. 12.22. (*а*) Двухвходовая двухслойная нейронная сеть без обратной связи. (*б*) и (*в*) Примеры разделяющих поверхностей, реализуемых такой сетью

пересечением трех прямых. Требование, чтобы класс ω_1 располагался с положительной стороны от всех трех линий, будет означать выпуклую область, ограниченную тремя этими линиями. На самом деле простым увеличением числа узлов первого слоя двухслойной нейронной сети можно построить любую, сколь угодно сложную, открытую или замкнутую выпуклую область.

Следующий логический шаг состоит в увеличении числа слоев до трех. В этом случае, как и прежде, узлы первого слоя реализуют прямые. Узлы второго слоя выполняют затем операции И, так чтобы построить области из различных полуплоскостей, образуемых этими прямыми. Узлы третьего слоя приписывают различные области к конкретным классам. Предположим, например, что класс ω_1 состоит из двух отдельных областей, каждая из которых ограничена своим набором прямых. Тогда два узла второго слоя отвечают за области, относящиеся к одному и тому же классу, а один из выходных узлов (в третьем слое) должен сигнализировать о появлении объекта из этого класса, когда высокий уровень имеется на выходе любого из упомянутых двух узлов второго слоя. При обозначении высоких и низких уровней на выходах узлов второго слоя 1 и 0 соответственно такая схема работы выходных узлов эквивалентна выполнению ими логической операции ИЛИ. Применительно к обсуждаемым узлам нейронной сети этого можно достичь, взяв значение смещения θ_j из полуоткрытого интервала [0, 1). Тогда всякий раз, когда установится высокий уровень на выходе по меньшей мере одного узла во втором слое, присоединенного с ненулевым весом к данному выходному узлу, на выходе последнего также появится высокий уровень, указывая на принадлежность предъявленного образа к тому классу, который соответствует этому узлу выходного слоя.

Рис. 12.23 суммирует итоги вышесказанного. Обратите внимание на нижнюю строку, иллюстрирующую, что сложность областей в пространстве решений, которые могут быть реализованы трехслойной нейронной сетью, в принципе является неограниченной. На практике серьезные трудности обычно вызывает построение второго слоя, который бы формировал набор правильных реакций для различных сочетаний конкретных классов. Причина этого в том, что разделяющие прямые не заканчиваются в местах пересечений, а неограниченно продолжаются, в результате чего образы одного класса могут оказаться по обе стороны разделяющих прямых в пространстве признаков. Практически на втором слое могут возникнуть трудности при выяснении того, какие прямые (полуплоскости) должны быть включены в операцию И для данного класса образов — или же это вообще невозможно сделать. Ссылка на проблему «исключающего ИЛИ» в третьей колонке рис. 12.23 относится к тому факту, что в случае двоичных признаков в двумерном пространстве возможно существование всего четырех различающихся объектов. При этом если в класс ω_1 входят образы {(0, 1), (1, 0)}, а в класс ω_2 — образы {(0, 0), (1, 1)}, то функция принадлежности объектов к классам задается логической операцией «исключающее ИЛИ», которая принимает единичное значение только тогда, когда любой один ее аргумент равен 1, и нулевое значение — в остальных случаях. Таким образом, значение 1 этой функции означает принадлежность объекта к классу ω_1, а значение 0 — к классу ω_2.

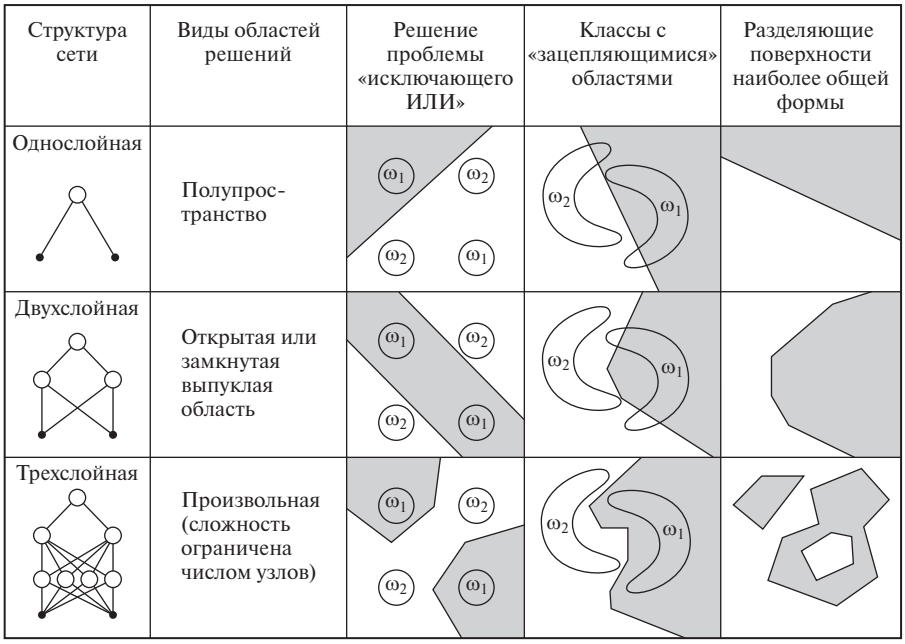


Рис. 12.23. Виды областей в пространстве решений, которые могут быть сформированы с помощью одно- или многослойных нейронных сетей без обратной связи при одном или двух скрытых слоях и двух входах. [Lippmann]

Проведенное выше рассмотрение легко обобщается на случай n измерений: вместо прямых мы имеем дело с гиперплоскостями. Однослойная сеть реализует разделяющую поверхность в форме гиперплоскости, которая делит на две части все n-мерное пространство признаков объектов. Двухслойная сеть строит произвольные выпуклые области, образованные пересекающимися гиперплоскостями. Трехслойная сеть реализует разделяющие поверхности произвольной сложности. В последних двух случаях достижимая сложность формы областей определяется числом узлов каждого слоя. В случае однослойной сети число классов ограничено двумя. В двух других случаях число классов произвольно, поскольку число выходных узлов можно выбрать в соответствии с решаемой задачей.

В этом месте возникает естественный вопрос: какой смысл имеет изучение нейронных сетей с числом слоев более трех? Ведь трехслойная сеть способна строить сколь угодно сложные разделяющие поверхности. Ответ кроется в том методе, который применяется для обучения сети с использованием лишь трех слоев. Обучающее правило для сети на рис. 12.16 минимизирует показатель ошибки, но ничего не говорит о том, как связывать группы гиперплоскостей с конкретными узлами второго слоя трехслойной сети рассматриваемого вида. По существу, проблема обмена числа слоев на число узлов в слое остается неизученной. На практике компромиссное решение обычно находится методом проб и ошибок либо по прошлому опыту решения задач в данной проблемной области.

12.3. Структурные методы распознавания

Методы, обсуждавшиеся в разделе 12.2, основывались на количественных характеристиках объектов и по большей части игнорировали структурные связи, присущие форме объектов. Рассматриваемые в этом разделе структурные методы, напротив, нацелены на распознавание образов именно за счет использования связей такого вида. В данном разделе будут представлены два основных подхода, применяемых для распознавания формы границы объекта, которая представляется в виде строки символов. На практике для использования структурных методов распознавания образов наиболее подходящим является именно строковое представление.

12.3.1. Сопоставление номеров фигур

Для сравнения формы границ областей, описанных в терминах номеров фигур (см. раздел 11.2.2), можно построить процедуру, аналогичную рассмотренной в разделе 12.1.1 идее минимума расстояния для векторов признаков. С учетом обсуждения в разделе 11.2.2 *степень сходства* k формы границ двух областей определяется как наибольшее значение порядка (т. е. числа цифр в записи номера фигуры), при котором их номера фигур еще совпадают. Например, пусть a и b — номера фигур для замкнутых границ, представленных 4-связными цепными кодами. Эти две формы имеют степень сходства k, если

$$\begin{aligned} s_j(a) &= s_j(b) \quad \text{при } j = 4,6,8,\ldots,k \\ s_j(a) &\neq s_j(b) \quad \text{при } j = k+2, k+4,\ldots, \end{aligned} \tag{12.3-1}$$

где функция s означает номер фигуры, а ее индекс равен значению порядка номера фигуры. *Расстояние* между двумя фигурами a и b определяется как величина, обратная их степени сходства:

$$D(a,b) = \frac{1}{k}. \tag{12.3-2}$$

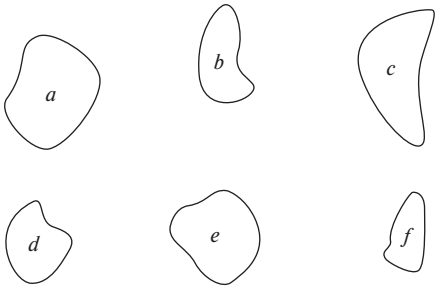
Определенная так мера расстояния действительно удовлетворяет следующим свойствам:

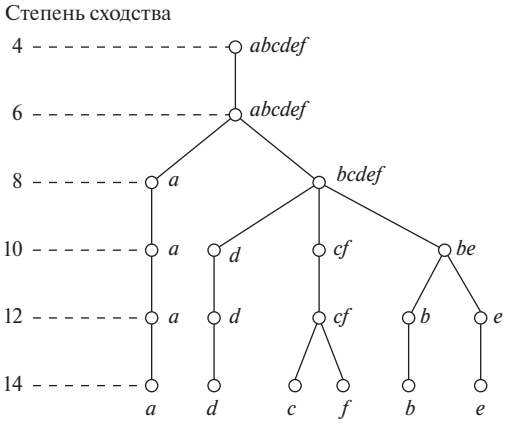
$$\begin{aligned} &D(a,b) \geq 0 \\ &D(a,b) = 0 \Leftrightarrow a = b \\ &D(a,c) \leq \max\left[D(a,b), D(b,c)\right]. \end{aligned} \tag{12.3-3}$$

Для сопоставления форм можно использовать как значение k, так и D. Если используется степень сходства, то чем больше значение k, тем более похожими по форме являются границы областей (заметим, что для идентичных фигур k равно бесконечности). При сравнении по мере расстояния (12.3-2) имеет место обратная зависимость.

Пример 12.7. Использование номеров фигур для сравнения форм областей.
■ Предположим, что имеется фигура f, и мы хотим найти наиболее похожую на нее среди набора других пяти фигур (a, b, c, d и e); все фигуры приведены

а
б *в*





	a	*b*	*c*	*d*	*e*	*f*
a	∞	6	6	6	6	6
b		∞	8	8	10	8
c			∞	8	8	12
d				∞	8	8
e					∞	8
f						∞

Рис. 12.24. (*а*) Фигуры. (*б*) Гипотетическое дерево сходства. (*в*) Матрица сходства. [Bribiesca, Guzman]

на рис. 12.24(*а*). Эта задача аналогична той, когда имеется пять образцов формы и требуется найти, к какому из них ближе всего данная неизвестная форма. Процедуру поиска можно изобразить с помощью дерева сходства, показанного на рис. 12.24(*б*). Корень дерева соответствует наименьшей возможной степени сходства, которая в данном примере равна 4. Предположим, что все формы идентичны вплоть до порядка 8, за исключением фигуры *а*, имеющей степень сходства с остальными фигурами, равную 6. Двигаясь по дереву ниже, находим, что фигура *d* имеет по отношению к остальным степень сходства 8 и так далее. Фигуры *f* и *c* однозначно согласуются друг с другом, имея более высокую степень сходства, чем любая другая пара фигур. Напротив, другую крайность образует фигура *а*; если бы она была неизвестной фигурой, то все, что можно было бы сказать с помощью такого метода, — что *а* похожа на все остальные фигуры со степенью сходства 6. Та же информация может быть компактно представлена в форме *матрицы сходства*, показанной на рис. 12.24(*в*). ■

12.3.2. Сопоставление строк символов

Предположим, что границы *a* и *b* двух областей закодированы в виде строк символов (см. раздел 11.5), которые обозначены $a_1a_2...a_n$ и $b_1b_2...b_m$ соответственно.

Будем говорить о совпадении строк на k-й позиции, если $a_k = b_k$. Пусть α — общее число совпадений этих строк, тогда число несовпадающих символов равно

$$\beta = \max(|a|,|b|) - \alpha, \tag{12.3-4}$$

где $|\arg|$ обозначает длину аргумента, представленного символьной строкой (т. е. число символов в ней). Можно показать, что $\beta = 0$ тогда и только тогда, когда представления a и b идентичны (см. задачу 12.21).

Простой мерой сходства a и b является отношение

$$R = \frac{\alpha}{\beta} = \frac{\alpha}{\max(|a|,|b|) - \alpha}. \tag{12.3-5}$$

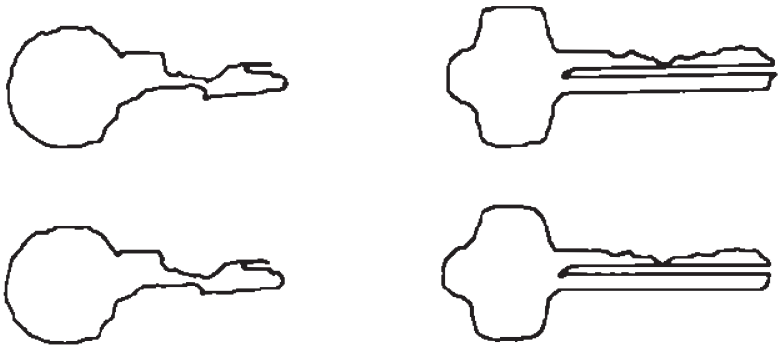
Отсюда следует, что величина R равна бесконечности при полном совпадении и нулю, когда соответственные символы a и b не совпадают ни на одной позиции (в этом случае $\alpha = 0$). Поскольку сравнение выполняется посимвольно, с точки зрения уменьшения вычислений важен выбор начальной точки на каждой из границ. В данной ситуации будет полезен любой способ, позволяющий хотя бы приблизительно стандартизовать выбор начальной точки, поскольку он даст преимущество перед полным перебором всех положений начальной точки в одной из строк символов, на каждом шаге которого требуется циклический сдвиг строки с вычислением величины (12.3-5). Наибольшее значение R, найденное при переборе, дает наилучший вариант совпадения строк.

Пример 12.8. Иллюстрация сопоставления строк символов.

■ На рис. 12.25(*а*) и (*б*) показаны примеры границы объектов из двух разных классов. Эти границы аппроксимировались ломаными линиями минимальной длины (см. раздел 11.1.3); их вид представлен на рис. 12.25(*в*) и (*г*). На основе этих ломаных были сформированы описания границ — строки символов, полученные вычислением внутреннего угла θ между соседними звеньями ломаной при обходе ее по часовой стрелке (значения угла кодировались одним из восьми символов для каждого интервала в 45°, т. е. α_1: $0° < \theta \le 45°$, α_2: $45° < \theta \le 90°$, ..., α_8: $315° < \theta \le 360°$).

В таблице на рис. 12.25(*д*) показаны результаты вычисления характеристики R для строк символов, соответствующих шести различным объектам первого класса, относительно друг друга. В ячейках таблицы указаны значения R, а обозначение, например, 1.c указывает на строку третьего объекта из класса 1. В таблице на рис. 12.25(*е*) показаны результаты аналогичного попарного сопоставления строк для объектов второго класса. Наконец, в таблице на рис. 12.25(*ж*) приводятся значения R, полученные при сравнении символьных строк, описывающих объекты из разных классов. Обратите внимание, что здесь значения R значительно меньше, чем любые значения в первых двух таблицах. Это указывает, что мера сходства R обладает хорошими дискриминантными свойствами для указанных двух классов объектов. Например, если бы было неизвестно, к какому классу относится строка символов, обозначенная 1.a, то при сравнении ее с образцами строк класса 1 *наименьшее* полученное значение характеристики R составило бы 4,7 (см. рис. 12.25(*д*)). Напротив, при сравнении с образцами класса 2 *наибольшее* значение R равнялось бы 1,24 (см. рис. 12.25(*ж*)). На этом основании можно сделать вывод, что строка 1.a является объектом класса 1. Такой

а *б*
в *г*
д *е*
ж



R	1.a	1.b	1.c	1.d	1.e	1.f
1.a	∞					
1.b	16.0	∞				
1.c	9.6	26.3	∞			
1.d	5.1	8.1	10.3	∞		
1.e	4.7	7.2	10.3	14.2	∞	
1.f	4.7	7.2	10.3	8.4	23.7	∞

R	2.a	2.b	2.c	2.d	2.e	2.f
2.a	∞					
2.b	33.5	∞				
2.c	4.8	5.8	∞			
2.d	3.6	4.2	19.3	∞		
2.e	2.8	3.3	9.2	18.3	∞	
2.f	2.6	3.0	7.7	13.5	27.0	∞

R	1.a	1.b	1.c	1.d	1.e	1.f
2.a	1.24	1.50	1.32	1.47	1.55	1.48
2.b	1.18	1.43	1.32	1.47	1.55	1.48
2.c	1.02	1.18	1.19	1.32	1.39	1.48
2.d	1.02	1.18	1.19	1.32	1.29	1.40
2.e	0.93	1.07	1.08	1.19	1.24	1.25
2.f	0.89	1.02	1.02	1.24	1.22	1.18

Рис. 12.25. (*а*) и (*б*) Примеры границ объектов из двух различных классов; (*в*) и (*г*) аппроксимация этих границ ломаными линиями; (*д*)—(*ж*) таблицы значений *R*. [Sze, Yang]

подход к классификации аналогичен классификатору по минимуму расстояния, описанному в разделе 12.2.1. ■

Заключение

Начиная с главы 9 наше изучение цифровой обработки изображений переместилось с процедур, выходом которых являлись изображения, к операциям, выходом которых являются признаки изображений, в том смысле, как это описывалось в разделе 1.1. Хотя представленный в настоящей главе материал носит вводный характер, затронутые темы имеют фундаментальное значение для понимания современного состояния проблемы распознавания объектов. Как от-

мечалось в начале главы, распознавание объектов является логическим завершением этой книги. Чтобы двигаться дальше, нам потребовались бы понятия, выходящие за те границы рассмотрения, которые мы наметили в разделе 1.4. В частности, следующим логическим шагом должно было бы стать изучение методов анализа изображений, для чего потребовались бы понятия из области искусственного интеллекта.

Как уже отмечалось в разделах 1.1 и 1.4, искусственный интеллект и некоторые смежные области, в частности анализ сцен и машинное зрение, все еще пребывают на относительно ранних стадиях практических разработок. Для решения задач анализа изображений сегодня характерны эвристические подходы. Хотя эти подходы и в самом деле разнообразны, однако большинство из них существенно опирается на методы, рассмотренные в данной книге.

Закончив освоение материала двенадцати глав, читатель теперь способен понимать основную проблематику сферы цифровой обработки изображений как с теоретической, так и с практической точки зрения. На протяжении всего рассмотрения мы старались заложить твердую основу для дальнейшего изучения как обсуждаемых, так и смежных областей. Учитывая специфический характер многих задач обработки изображений, ясное понимание базовых принципов существенно повышает шансы на успешное решение этих задач.

Ссылки и литература для дальнейшего изучения

Подготовительный материал для разделов 12.1—12.2.2 приводится в книгах [Theodoridis, Koutroumbas, 2006], [Duda, Hart, Stork, 2001] и [Tou, Gonzalez, 1974]. Также представляет интерес обзор [Jain et al., 2000]. В книге [Principe et al., 1999] дан хороший обзор нейронных сетей. Специальный выпуск журнала [*IEEE Trans. Image Processing*, 1998] стоит сравнить с аналогичным специальным выпуском, вышедшим десятью годами ранее ([*IEEE Computer*, 1988]). Материал раздела 12.2.3 носит вводный характер. Фактически рассматриваемая модель нейронной сети является лишь одной из многочисленных моделей, предложенных за прошедшее время. Тем не менее эта модель является представительной и широко используется в обработке изображений. Пример с распознаванием искаженных фигур является переделанным вариантом из статей [Gupta et al., 1990, 1994]. В работе [Gori, Scarselli, 1998] обсуждается вопрос о мощности классификации, обеспечиваемой многослойными нейронными сетями. Хорошим дополнительным источником по этой теме служит статья [Ueda, 2000], где сообщается о подходе, основанном на использовании линейных комбинаций нейронных сетей для минимизации ошибки классификации.

В качестве дополнительного материала по разделу 12.3.1 рекомендуется статья [Bribiesca, Guzman, 1980]. По поводу сопоставления строк символов см. работы [Sze, Yang, 1981], [Oommen, Loke, 1997] и [Gdalyahu, Weinshall, 1999]. По тематике структурных методов распознавания образов дополнительно рекомендуются работы [Gonzalez, Thomason, 1978], [Fu, 1982], [Bunke, Sanfeliu, 1990], [Tanaka, 1995], [Vailaya et al., 1998], [Aizaka, Nakamura, 1999] и [Jonk et al., 1999], а также книга [Huang, 2002].

Задачи

Подробные решения задач, отмеченных звездочкой, приводятся на интернет-сайте книги. Там же можно найти темы предлагаемых проектов, основанных на материале данной главы.

12.1 **(а)** Вычислите дискриминантные функции классификатора по минимуму расстояния для образов на рис. 12.1. Необходимые векторы математического ожидания получите самостоятельно путем измерений и анализа данных на рисунке.

(б) Нарисуйте разделяющие поверхности, реализуемые дискриминантными функциями в (а).

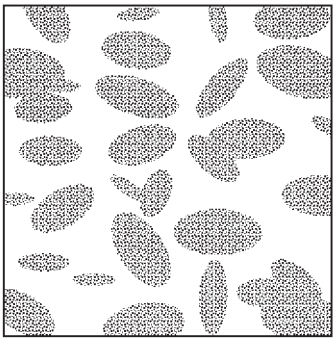
★12.2 Покажите, что формулы (12.2-4) и (12.2-5) реализуют одну и ту же функцию с точки зрения классификации объектов.

12.3 Покажите, что поверхность, заданная уравнением (12.2-6), есть гиперплоскость в n-мерном пространстве, перпендикулярная отрезку, соединяющему точки $\mathbf{m}_i$ и $\mathbf{m}_j$, и проходящая через его середину.

★12.4 Покажите, как обсуждаемый в связи с рис. 12.7 классификатор по минимуму расстояния можно реализовать с помощью W блоков резисторов (где W — число классов), сумматоров по числу блоков (для суммирования токов) и W-входовой схемы выбора максимума, способной выбирать вход, через который протекает максимальный ток.

12.5 Покажите, что коэффициент корреляции, определяемый соотношением (12.2-8), принимает значения в диапазоне [–1, 1]. (*Подсказка:* выразите функцию $\gamma(x, y)$ в векторной форме.)

★12.6 В результате эксперимента получаются двоичные изображения пятен эллиптической формы (см. рисунок ниже). Пятна бывают трех размеров со средними значениями главных осей эллипсов (1,3, 0,7), (1,0, 0,5) и (0,75, 0,25). Разброс размеров этих осей составляет ±15 % от указанных средних значений. Разработайте систему обработки изображений, которая бы отбрасывала неполные или перекрывающиеся эллипсы, а затем классифицировала бы оставшиеся раздельные эллипсы по трем указанным классам. Представьте решение в виде блок-схемы с указанием конкретных подробностей работы каждого блока. Для решения задачи классификации используйте классификатор по минимуму расстояния, четко указав, как предполагается получать обучающую выборку и как объекты из этой выборки будут использоваться для обучения классификатора.



12.7 Следующие два класса образов имеют гауссовы распределения: ω_1: $\{(0, 0)^T, (4, 0)^T, (4, 4)^T, (0, 4)^T\}$ и ω_2: $\{(5, 5)^T, (7, 5)^T, (7, 7)^T, (5, 7)^T\}$.

(а) Предполагая, что $P(\omega_1) = P(\omega_2) = 1/4$, получите уравнение байесовской разделяющей поверхности между этими двумя классами.

(б) Нарисуйте эту разделяющую поверхность.

★12.8 Решите задачу 12.7 для следующих классов образов: ω_1: $\{(-1, 0)^T, (0, -1)^T, (1, 0)^T, (0, 1)^T\}$ и ω_2: $\{(-4, 0)^T, (0, -4)^T, (4, 0)^T, (0, 4)^T\}$. Заметьте, что эти классы не являются линейно разделимыми.

12.9 Решите задачу 12.6 с использованием байесовского классификатора, предполагая, что распределения являются нормальными. Четко укажите, как предполагается получать обучающую выборку и как объекты из этой выборки будут использоваться для обучения классификатора.

★12.10 Байесовские дискриминантные функции $d_j(\mathbf{x}) = p(\mathbf{x}|\omega_j)P(\omega_j)$, $j = 1, 2,..., W$ были выведены с использованием нуль-единичной функции потерь. Докажите, что эти дискриминантные функции обеспечивают минимальную вероятность ошибки. (*Подсказка:* вероятность ошибки $p(e)$ равна $1 - p(c)$, где $p(c)$ — вероятность правильного распознавания. Если образ с вектором признаков $\mathbf{x}$ принадлежит классу ω_i, то $p(c|\mathbf{x}) = p(\omega_i|\mathbf{x})$. Найдите $p(c)$ и покажите, что она максимальна (т. е. $p(e)$ минимальна), когда $p(\mathbf{x}|\omega_i)P(\omega_i)$ принимает максимальное значение.)

12.11 **(а)** Примените алгоритм персептрона к следующим классам образов: ω_1: $\{(0, 0, 0)^T, (1, 0, 0)^T, (1, 0, 1)^T, (1, 1, 0)^T\}$ и ω_2: $\{(0, 0, 1)^T, (0, 1, 1)^T, (0, 1, 0)^T, (1, 1, 1)^T\}$. Положите $c = 1$, а $\mathbf{w}(1) = (-1, -2, -2, 0)^T$.

(б) Нарисуйте разделяющую поверхность, полученную в п. (а). Укажите положительную сторону поверхности и области, занимаемые классами.

★12.12 Алгоритм персептрона, задаваемый рекуррентными соотношениями (12.2-34)—(12.2-36), можно выразить более компактно, умножая образы класса ω_2 на -1. При этом корректирующие шаги алгоритма приобретают вид $\mathbf{w}(k + 1) = \mathbf{w}(k)$, если $\mathbf{w}^T(k)\mathbf{y}(k) > 0$, и $\mathbf{w}(k + 1) = \mathbf{w}(k) + c\mathbf{y}(k)$ в остальных случаях. Это одна из нескольких известных формулировок алгоритма обучения персептрона, которую можно вывести из общей формулы градиентного спуска

$$\mathbf{w}(k+1) = \mathbf{w}(k) - c\left[\frac{\partial J(\mathbf{w}, \mathbf{y})}{\partial \mathbf{w}}\right]_{\mathbf{w}=\mathbf{w}(k)},$$

где $c > 0$, $J(\mathbf{w}, \mathbf{y})$ — целевая функция и частная производная вычисляется в точке $\mathbf{w} = \mathbf{w}(k)$. Покажите, что эту формулировку алгоритма персептрона можно получить из общей процедуры градиентного спуска, используя целевую функцию вида $J(\mathbf{w}, \mathbf{y}) = (|\mathbf{w}^T\mathbf{y}| - \mathbf{w}^T\mathbf{y})/2$, где $|\arg|$ — абсолютное значение аргумента. (*Примечание:* частная производная $\mathbf{w}^T\mathbf{y}$ по $\mathbf{w}$ равна $\mathbf{y}$.)

12.13 Докажите, что алгоритм обучения персептрона, задаваемый соотношениями (12.2-34)—(12.2-36), сходится за конечное число шагов, если обучающие выборки классов являются линейно разделимыми. [*Подсказка:* умножьте образы из класса ω_2 на -1 и введите в рассмотрение такой неотрицательный порог T, чтобы алгоритм обучения персептрона (со значением $c = 1$) выражался условиями $\mathbf{w}(k + 1) = \mathbf{w}(k)$, если $\mathbf{w}^T(k)\mathbf{y}(k) > \mathrm{T}$, и $\mathbf{w}(k + 1) = \mathbf{w}(k) + \mathbf{y}(k)$

в остальных случаях. Вам может потребоваться неравенство Коши — Шварца: $\|\mathbf{a}\|^2 \cdot \|\mathbf{b}\|^2 \geq (\mathbf{a}^T\mathbf{b})^2$].

★12.14 Укажите структуру и весовые коэффициенты нейронной сети, которая бы действовала *в точности* так же, как классификатор по минимуму расстояния для двух классов образов в n-мерном пространстве.

12.15 Укажите структуру и весовые коэффициенты нейронной сети, которая бы действовала *в точности* так же, как байесовский классификатор для двух классов образов в n-мерном пространстве. Считайте, что классы подчиняются нормальным законам распределения с отличающимися математическими ожиданиями и одинаковыми ковариационными матрицами.

★12.16 **(а)** При каких условиях нейронные сети, построенные в задачах 12.14 и 12.15, будут идентичны?

(б) Можно ли получить конкретную нейронную сеть из п. (а), применяя описанное в разделе 12.2.3 обобщенное дельта-правило для обучения многослойной нейронной сети без обратной связи на достаточно большой выборке образов из каждого класса?

12.17 Два класса в двумерном пространстве имеют такие распределения, что образы класса ω_1 случайно распределены вдоль окружности радиуса r_1, а образы класса ω_2 — вдоль концентрической окружности радиуса r_2, где $r_2 = 4r_1$. Постройте структуру нейронной сети с минимальным количеством слоев и узлов, которая бы правильно классифицировала образы из этих двух классов.

★12.18 Решите задачу 12.6 с помощью нейронной сети. Четко укажите, как предполагается получать обучающую выборку и как объекты из этой выборки будут использоваться для обучения классификатора. Выберите наиболее простую конфигурацию нейронной сети, которая, по Вашему мнению, подходит для решения задачи.

12.19 Покажите, что приведенное в (12.2-71) выражение $h_j'(I_j) = O_j(1 - O_j)$, где $h_j'(I_j) = \partial h_j(I_j)/\partial I_j$, следует из выражения (12.2-50) при $\theta_0 = 1$.

***12.20** Покажите, что мера расстояния $D(A, B)$ из определения (12.3-2) удовлетворяет свойствам (12.3-3).

12.21 Покажите, что величина β в (12.3-4) равна 0 тогда и только тогда, когда строки символов a и b идентичны.

12.22 Фабрика осуществляет массовое производство маленьких американских флажков для спортивных мероприятий. Служба гарантии качества заметила, что в моменты пика производства на некоторых печатных машинах случайным образом пропадают от одной до трех звездочек и от одной до трех полосок. В остальном выпускаемые флажки не содержат дефектов. Хотя брак и составляет малую долю общего объема производства, директор фабрики хочет решить эту проблему и считает, что наиболее экономичным способом будет автоматизированный контроль с использованием методов обработки изображений. Основные технические данные производственного процесса следующие. Флажки имеют размеры приблизительно 7,5 × 12,5 см и двигаются по производственной линии со скоростью приблизительно 40 см/с в продольной ориентации (с допустимым отклонением ±10°) с расстоянием между флажками приблизительно 5 см. «Приблизительно» во всех случаях означает допустимое отклонение ±5 %. Директор фабрики нанимает Вас для разработ-

ки системы обработки изображений к каждой из производственных линий. Вам сказано, что оценка предлагаемого подхода будет определяться с позиций его простоты и стоимости. Разработайте полную систему, исходя из модели на рис. 1.23. Представьте Ваш проект (включая допущения и спецификации) в форме краткого (но ясного) письменного отчета, адресованного директору фабрики.

Приложения

Кодовые таблицы для сжатия изображений

В данном приложении содержатся таблицы кодов для использования со стандартами сжатия MKKTT и JPEG. Табл. A-1 и A-2 представляют собой модифицированные таблицы кодов Хаффмана для сжатия MKKTT руппы 3 и группы 4. Таблицы A-3, A-4 и A5 используются при JPEG-кодировании коэффициентов ДКП. По поводу использования этих таблиц см. разделы 8.2.5 и 8.2.8 главы 8.

Таблица A-1. Коды окончания MKKTT

Длина серии	Кодовое слово		Длина серии	Кодовое слово	
	Белая серия	Черная серия		Белая серия	Черная серия
0	00110101	0000110111	32	00011011	000001101010
1	000111	010	33	00010010	000001101011
2	0111	11	34	00010011	000011010010
3	1000	10	35	00010100	000011010011
4	1011	011	36	00010101	000011010100
5	1100	0011	37	00010110	000011010101
6	1110	0010	38	00010111	000011010110
7	1111	00011	39	00101000	000011010111
8	10011	000101	40	00101001	000001101100
9	10100	000100	41	00101010	000001101101
10	00111	0000100	42	00101011	000011011010
11	01000	0000101	43	00101100	000011011011
12	001000	0000111	44	00101101	000001010100
13	000011	00000100	45	00000100	000001010101
14	110100	00000111	46	00000101	000001010110
15	110101	000011000	47	00001010	000001010111
16	101010	0000010111	48	00001011	000001100100
17	101011	0000011000	49	01010010	000001100101
18	0100111	0000001000	50	01010011	000001010010
19	0001100	00001100111	51	01010100	000001010011
20	0001000	00001101000	52	01010101	000000100100
21	0010111	00001101100	53	00100100	000000110111
22	0000011	00000110111	54	00100101	000000111000
23	0000100	00000101000	55	01011000	000000100111
24	0101000	00000010111	56	01011001	000000101000
25	0101011	00000011000	57	01011010	000001011000
26	0010011	000011001010	58	01011011	000001011001
27	0100100	000011001011	59	01001010	000000101011
28	0011000	000011001100	60	01001011	000000101100
29	00000010	000011001101	61	00110010	000001011010
30	00000011	000001101000	62	00110011	000001100110
31	00011010	000001101001	63	00110100	000001100111

Таблица А-2. Коды продолжения MKKTT

Длина серии	Кодовое слово		Длина серии	Кодовое слово	
	Белая серия	Черная серия		Белая серия	Черная серия
64	11011	0000001111	960	011010100	0000001110011
128	10010	000011001000	1024	011010101	0000001110100
192	010111	000011001001	1088	011010110	0000001110101
256	0110111	000001011011	1152	011010111	0000001110110
320	00110110	000000110011	1216	011011000	0000001110111
384	00110111	000000110100	1280	011011001	0000001010010
448	01100100	000000110101	1344	011011010	0000001010011
512	01100101	0000001101100	1408	011011011	0000001010100
576	01101000	0000001101101	1472	010011000	0000001010101
640	01100111	0000001001010	1536	010011001	0000001011010
704	011001100	0000001001011	1600	010011010	0000001011011
768	011001101	0000001001100	1664	011000	0000001100100
832	011010010	0000001001101	1728	010011011	0000001100101
896	011010011	0000001110010			

	Кодовое слово		Кодовое слово
1792	00000001000	2240	000000010110
1856	00000001100	2304	000000010111
1920	00000001101	2368	000000011100
1984	000000010010	2432	000000011101
2048	000000010011	2496	000000011110
2112	000000010100	2560	000000011111
2176	000000010101	КС	000000000001

Таблица А-3. Категории кодирования JPEG-коэффициентов[1]

Диапазон	Категория DC-разностей	Категория AC
0	0	–
–1, 1	1	1
–3, –2, 2, 3	2	2
–7, ..., –4, 4, ..., 7	3	3
–15, ..., –8, 8, ..., 15	4	4
–31, ..., –16, 16, ..., 31	5	5
–63, ..., –32, 32, ..., 63	6	6
–127, ..., –64, 64, ..., 127	7	7
–255, ..., –128, 128, ..., 255	8	8
–511, ..., –256, 256, ..., 511	9	9
–1023, ..., –512, 512, ..., 1023	A	A
–2047, ..., –1024, 1024, ..., 2047	B	B
–4095, ..., –2048, 2048, ..., 4095	C	C
–8191, ..., –4096, 4096, ..., 8191	D	D
–16383, ..., –8192, 8192, ..., 16383	E	E
–32767, ..., –16384, 16384, ..., 32767	F	–

Таблица А-4. Стандартные JPEG-коды для DC-коэффициентов (яркость)

Категория	Основной код	Длина	Категория	Основной код	Длина
0	010	3	6	1110	10
1	011	4	7	11110	12
2	100	5	8	111110	14
3	00	5	9	1111110	16
4	101	7	A	11111110	18
5	110	8	B	111111110	20

[1] Для сокращения записи в таблицах А-3—А-5 применены шестнадцатеричные числа; в них первые 10 цифр совпадают с цифрами десятичных чисел, а последние 6 обозначаются латинскими буквами от 0 до F. Тем самым, $A_{16} = 10_{10}$, $B_{16} = 11_{10}$, $C_{16} = 12_{10}$, $D_{16} = 13_{10}$, $E_{16} = 14_{10}$, $F_{16} = 15_{10}$. — *Прим. перев.*

Таблица А-5. Стандартные JPEG-коды для AC-коэффициентов (яркость)

Длина серии/ категория	Основной код	Длина	Длина серии/ категория	Основной код	Длина
0/0	1010 (=EOB)	4	4/1	111011	7
0/1	00	3	4/2	1111111000	12
0/2	01	4	4/3	1111111110010111	19
0/3	100	6	4/4	1111111110011000	20
0/4	1011	8	4/5	1111111110011001	21
0/5	11010	10	4/6	1111111110011010	22
0/6	111000	12	4/7	1111111110011011	23
0/7	1111000	14	4/8	1111111110011100	24
0/8	1111110110	18	4/9	1111111110011101	25
0/9	1111111110000010	25	4/A	1111111110011110	26
0/A	1111111110000011	26	5/1	1111010	8
1/1	1100	5	5/2	1111111001	12
1/2	111001	8	5/3	1111111110011111	19
1/3	1111001	10	5/4	1111111110100000	20
1/4	111110110	13	5/5	1111111110100001	21
1/5	11111110110	16	5/6	1111111110100010	22
1/6	1111111110000100	22	5/7	1111111110100011	23
1/7	1111111110000101	23	5/8	1111111110100100	24
1/8	1111111110000110	24	5/9	1111111110100101	25
1/9	1111111110000111	25	5/A	1111111110100110	26
1/A	1111111110001000	26	6/1	1111011	8
2/1	11011	6	6/2	11111111000	13
2/2	11111000	10	6/3	1111111110100111	19
2/3	1111110111	13	6/4	1111111110101000	20
2/4	1111111110001001	20	6/5	1111111110101001	21
2/5	1111111110001010	21	6/6	1111111110101010	22
2/6	1111111110001011	22	6/7	1111111110101011	23
2/7	1111111110001100	23	6/8	1111111110101100	24
2/8	1111111110001101	24	6/9	1111111110101101	25
2/9	1111111110001110	25	6/A	1111111110101110	26
2/A	1111111110001111	26	7/1	11111001	9
3/1	111010	7	7/2	11111111001	13
3/2	111110111	11	7/3	1111111110101111	19
3/3	11111110111	14	7/4	1111111110110000	20
3/4	1111111110010000	20	7/5	1111111110110001	21
3/5	1111111110010001	21	7/6	1111111110110010	22
3/6	1111111110010010	22	7/7	1111111110110011	23
3/7	1111111110010011	23	7/8	1111111110110100	24
3/8	1111111110010100	24	7/9	1111111110110101	25
3/9	1111111110010101	25	7/A	1111111110110110	26
3/A	1111111110010110	26	8/1	11111010	9

Таблица А-5. (*Окончание*)

Длина серии/ категория	Основной код	Длина	Длина серии/ категория	Основной код	Длина
8/2	111111111000000	17	C/2	1111111111011010	18
8/3	1111111110110111	19	C/3	1111111111011011	19
8/4	1111111110111000	20	C/4	1111111111011100	20
8/5	1111111110111001	21	C/5	1111111111011101	21
8/6	1111111110111010	22	C/6	1111111111011110	22
8/7	1111111110111011	23	C/7	1111111111011111	23
8/8	1111111110111100	24	C/8	1111111111100000	24
8/9	1111111110111101	25	C/9	1111111111100001	25
8/A	1111111110111110	26	C/A	1111111111100010	26
9/1	111111000	10	D/1	11111111010	12
9/2	1111111110111111	18	D/2	1111111111100011	18
9/3	1111111111000000	19	D/3	1111111111100100	19
9/4	1111111111000001	20	D/4	1111111111100101	20
9/5	1111111111000010	21	D/5	1111111111100110	21
9/6	1111111111000011	22	D/6	1111111111100111	22
9/7	1111111111000100	23	D/7	1111111111101000	23
9/8	1111111111000101	24	D/8	1111111111101001	24
9/9	1111111111000110	25	D/9	1111111111101010	25
9/A	1111111111000111	26	D/A	1111111111101011	26
A/1	111111001	10	E/1	111111110110	13
A/2	1111111111001000	18	E/2	1111111111101100	18
A/3	1111111111001001	19	E/3	1111111111101101	19
A/4	1111111111001010	20	E/4	1111111111101110	20
A/5	1111111111001011	21	E/5	1111111111101111	21
A/6	1111111111001100	22	E/6	1111111111110000	22
A/7	1111111111001101	23	E/7	1111111111110001	23
A/8	1111111111001110	24	E/8	1111111111110010	24
A/9	1111111111001111	25	E/9	1111111111110011	25
A/A	1111111111010000	26	E/A	1111111111110100	26
B/1	111111010	10	**F/0**	111111110111	12
B/2	1111111111010001	18	F/1	1111111111110101	17
B/3	1111111111010010	19	F/2	1111111111110110	18
B/4	1111111111010011	20	F/3	1111111111110111	19
B/5	1111111111010100	21	F/4	1111111111111000	20
B/6	1111111111010101	22	F/5	1111111111111001	21
B/7	1111111111010110	23	F/6	1111111111111010	22
B/8	1111111111010111	24	F/7	1111111111111011	23
B/9	1111111111011000	25	F/8	1111111111111100	24
B/A	1111111111011001	26	F/9	1111111111111101	25
C/1	1111111010	11	F/A	1111111111111110	26

Литература

Abidi M.A., Gonzalez R.C. (eds.) [1992]. *Data Fusion in Robotics and Machine Intelligence*, Academic Press, New York.

Abidi M.A., Eason R.O., Gonzalez R.C. [1991]. *Autonomous Robotics Inspection and Manipulation Using Multisensor Feedback*. IEEE Computer, vol. 24, no. 4, pp. 17–31.

Abramson N. [1963]. *Information Theory and Coding*, McGraw-Hill, New York.

Adiv G. [1985]. *Determining Three-Dimensional Motion and Structure from Optical Flow Generated by Several Moving Objects.* IEEE Trans. Pattern Analysis Machine Intelligence, vol. PAMI-7, no. 4, pp. 384–401.

Aggarwal J.K., Badler N.I. (eds.) [1980]. *Motion and Time-Varying Imagery.* IEEE Trans. Pattern Analysis Machine Intelligence, Special Issue, vol. PAMI-2, no. 6, pp. 493–588.

Aguado A.S., Nixon M.S., Montiel M.M. [1998]. *Parameterizing Arbitrary Shapes via Fourier Descriptors for Evidence-Gathering Extraction*. Computer Vision and Image Understanding, vol. 69, no. 2, pp. 202–221.

Ahmed N., Natarajan T., Rao K.R. [1974]. *Discrete Cosine Transforms.* IEEE Trans. Computers, vol. C-23, pp. 90–93.

Ahmed N., Rao K.R. [1975]. *Orthogonal Transforms for Digital Signal Processing*, Springer-Verlag, New York. [Имеется перевод: Ахмед Н., Рао К.Р. *Ортогональные преобразования при обработке цифровых сигналов.* — М.: Связь, 1980.]

Aizaka K., Nakamura A. [1999]. *Parsing of Two-Dimensional Images Represented by Quadtree Adjoining Grammars.* Pattern Recognition, vol. 32, no. 2, pp. 277–294.

Alexiadis D.S., Sergiadis G.D. [2007]. *Estimation of Multiple Accelerated Motions Using Chirp-Fourier Transforms and Clustering.* IEEE Trans. Image Proc., vol. 16, no. 1, pp. 142–152.

Alliney S. [1993]. *Digital Analysis of Rotated Images*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 15, no. 5, pp. 499–504.

Ando S. [2000]. *Consistent Gradient Operators*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 22, no. 3, pp. 252–265.

Andrews H.C. [1970]. *Computer Techniques in Image Processing*, Academic Press, New York. [Имеется перевод: Эндрюс Г. *Применение вычислительных машин для обработки изображений.* — М.: Энергия, 1977.]

Andrews H.C., Hunt B.R. [1977]. *Digital Image Restoration*, Prentice Hall, Englewood Cliffs, N.J.

Anelli G., Broggi A., Destri G. [1998]. *Decomposition of Arbitrarily-Shaped Morphological Structuring Elements Using Genetic Algorithms.* IEEE Trans. Pattern Analysis Machine Intelligence, vol. 20, no. 2, pp. 217–224.

Ang P.H., Ruetz P.A., Auld D. [1991]. *Video Compression Makes Big Gains.* IEEE Spectrum, vol. 28, no. 10, pp. 16–19.

Antonini M., Barlaud M., Mathieu P., Daubechies I. [1992]. *Image Coding Using Wavelet Transform. IEEE Trans.* Image Processing, vol. 1, no. 2, pp. 205–220.

Ascher R.N., Nagy G. [1974]. *A Means for Achieving a High Degree of Compaction on Scan-Digitized Printed Text.* IEEE Transactions on Comp., C-23, pp. 1174–1179.

Atchison D.A., Smith G. [2000]. *Optics of the Human Eye,* Butterworth-Heinemann, Boston, Mass.

Baccar M., Gee L.A., Abidi M.A., Gonzalez, R.C. [1996]. *Segmentation of Range Images via Data Fusion and Morphological Watersheds*. Pattern Recognition, vol. 29, no. 10, pp. 1671–1685.

Bajcsy R., Lieberman L. [1976]. *Texture Gradient as a Depth Cue*. Computer Graphics Image Processing, vol. 5, no. 1, pp. 52–67.

Bakir T., Reeves J.S. [2000]. *A Filter Design Method for Minimizing Ringing in a Region of Interest in MR Spectroscopic Images*. IEEE Trans. Medical Imaging, vol. 19, no. 6, pp. 585–600.

Ballard D.H. [1981]. *Generalizing the Hough Transform to Detect Arbitrary Shapes*. Pattern Recognition, vol. 13, no. 2, pp. 111–122.

Ballard D.H., Brown C.M. [1982]. *Computer Vision*, Prentice Hall, Englewood Cliffs, N.J.

Banham M.R., Galatsanos H.L., Gonzalez H.L., Katsaggelos A.K. [1994]. *Multichannel Restoration of Single Channel Images Using a Wavelet-Based Subband Decomposition*. IEEE Trans. Image Processing, vol. 3, no. 6, pp. 821–833.

Banham M.R., Katsaggelos A.K. [1996]. *Spatially Adaptive Wavelet-Based Multiscale Image Restoration*. IEEE Trans. Image Processing, vol. 5, no. 5, pp. 619–634.

Basart J.P., Gonzalez R.C. [1992]. *Binary Morphology*, in Advances in Image Analysis, Mahdavieh Y., Gonzalez R.C. (eds.), SPIE Press, Bellingham, Wash., pp. 277–305.

Basart J.P., Chacklackal M.S., Gonzalez R.C. [1992]. *Introduction to Gray-Scale Morphology*, in Advances in Image Analysis, Mahdavieh Y., Gonzalez R.C. (eds.), SPIE Press, Bellingham, Wash., pp. 306–354.

Bates R.H.T., McDonnell M.J. [1986]. *Image Restoration and Reconstruction*. Oxford University Press, New York. [Имеется перевод: Бейтс Р., Мак-Доннелл М. *Восстановление и реконструкция изображений*. — М.: Мир, 1989.]

Battle G. [1987]. *A Block Spin Construction of Ondelettes. Part I: Lemarié Functions*. Commun. Math. Phys., vol. 110, pp. 601–615.

Battle G. [1988]. *A Block Spin Construction of Ondelettes. Part II: the QFT Connection*. Commun. Math. Phys., vol. 114, pp. 93–102.

Baumert L.D., Golomb S.W., Hall M., Jr. [1962]. *Discovery of a Hadamard Matrix of Order 92*. Bull. Am. Math. Soc., vol. 68, pp. 237–238.

Baxes G.A. [1994]. *Digital Image Processing: Principles and Applications*, John Wiley & Sons, New York.

Baylon D.M., Lim J.S. [1990]. *Transform/Subband Analysis and Synthesis of Signals*. Tech. Report, MIT Research Laboratory of Electronics. Cambridge. Mass.

Bell E.T. [1965]. *Men of Mathematics*, Simon & Schuster, New York.

Bengtsson A., Eklundh J.O. [1991]. *Shape Representation by Multiscale Contour Approximation*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 13, no. 1, pp. 85–93.

Benson K.B. [1985]. *Television Engineering Handbook*, McGraw-Hill, New York.

Berger T. [1971]. *Rate Distortion Theory*, Prentice Hall, Englewood Cliffs, N.J.

Beucher S. [1990]. *Doctoral Thesis*, Centre de Morphologie Mathématique, École des Mines de Paris, France. (The core of this material is contained in the following paper.)

Beucher S., Meyer F. [1992]. *The Morphological Approach of Segmentation: The Watershed Transformation*, in Mathematical Morphology in Image Processing, Dougherty E. (ed.), Marcel Dekker, New York.

Bezdek J. C., et al. [2005]. *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*. Springer, New York.

Bhaskaran V., Konstantinos K. [1997]. *Image and Video Compression Standards: Algorithms and Architectures*. Kluwer, Boston, Mass.

Bhatt B., Birks D., Hermreck D. [1997]. *Digital Television: Making It Work*. IEEE Spectrum, vol. 34, no. 10, pp. 19–28.

Biberman L. M. [1973]. *Image Quality*. In Perception of Displayed Information. Biberman L. M. (ed.), Plenum Press, New York.

Bichsel M. [1998]. *Analyzing a Scene's Picture Set under Varying Lightning*. Computer Vision and Image Understanding, vol. 71, no. 3, pp. 271–280.

Bieniek A., Moga A. [2000]. *An Efficient Watershed Algorithm Based on Connected Components*. Pattern Recognition, vol. 33, no. 6, pp. 907–916.

Bisignani W. T., Richards G. P., Whelan J. W. [1966]. *The Improved Grey Scale and Coarse-Fine PCM Systems: Two New Digital TV Bandwidth Reduction Techniques*. Proc. IEEE, vol. 54, no. 3, pp. 376–390. [Имеется перевод: Бизиньяни, Ричардс, Уилан. *Система КИМ с улучшенным воспроизведением серых тонов и система КИМ на основе грубой и точной шкалы — два новых метода сокращения полосы дискретного телевизионного сигнала*. — ТИИЭР, 1973, т. 54, № 3, с. 45–65.]

Blahut R. E. [1987]. *Principles and Practice of Information Theory*, Addison-Wesley, Reading, Mass.

Bleau A., Leon L. J. [2000]. *Watershed-Based Segmentation and Region Merging*. Computer Vision and Image Understanding, vol. 77, no. 3, pp. 317–370.

Blouke M. M., Sampat N., Canosa J. [2001]. *Sensors and Camera Systems for Scientific, Industrial, and Digital Photography Applications-II*, SPIE Press, Bellingham, Wash.

Blum H. [1967]. *A Transformation for Extracting New Descriptors of Shape*. In Models for the Perception of Speech and Visual Form, Wathen-Dunn W. (ed.), MIT Press, Cambridge, Mass.

Blume H., Fand A. [1989]. *Reversible and Irreversible Image Data Compression Using the S-Transform and Lempel-Ziv Coding*. Proc. SPIE Medical Imaging III: Image Capture and Display, vol. 1091, pp. 2–18.

Boie R. A., Cox I. J. [1992]. *An Analysis of Camera Noise*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 14, no. 6, pp. 671–674.

Born M., Wolf E. [1999]. *Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light*, 7th ed., Cambridge University Press, Cambride, UK. [Имеется перевод четвертого издания: Борн М., Вольф Э. *Основы оптики*. — М.: Наука,1973.]

Boulgouris N. V., Tzovaras D., Strintzis M. G. [2001]. *Lossless Image Compression Based on Optimal Prediction, Adaptive Lifting, and Conditional Arithmetic Coding*. IEEE Trans. Image Processing, vol. 10, no. 1, pp. 1–14.

Bouman C., Liu B. [1991]. *Multiple Resolution Segmentation of Textured Images*. IEEE Trans. Pattern. Analysis Machine Intelligence, vol. 13, no. 2, pp. 99–113.

Boyd J. E. Meloche J. [1998]. *Binary Restoration of Thin Objects in Multidimensional Imagery*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 20, no. 6, pp. 647–651.

Bracewell R. N. [1995]. *Two-Dimensional Imaging*, Prentice Hall, Upper Saddle River, N.J.

Bracewell R. N. [2000]. *The Fourier Transform and its Applications*, 3rd ed. McGraw-Hill, New York.

Brechet L., Lucas M., Doncarli C., Farnia D. [2007]. *Compression of Biomedical Signals with Mother Wavelet Optimization and Best-Basis Wavelet Packet Selection*. IEEE Trans. on Biomedical Engineering, in press.

Bribiesca E. [1981]. *Arithmetic Operations Among Shapes Using Shape Numbers*. Pattern Recognition, vol. 13, no. 2, pp. 123–138.

Bribiesca E. [1999]. *A New Chain Code*. Pattern Recognition, vol. 32, no. 2, pp. 235–251.

Bribiesca E. [2000]. *A Chain Code for Representing 3–D Curves*. Pattern Recognition, vol. 33, no. 5, pp. 755–765.

Bribiesca E., Guzman A. [1980]. *How to Describe Pure Form and How to Measure Differences in Shape Using Shape Numbers*. Pattern Recognition, vol. 12, no. 2, pp. 101–112.

Brigham E. O. [1988]. *The Fast Fourier Transform and its Applications*, Prentice Hall, Upper Saddle River, N.J.

Brinkman B. H., Manduca A., Robb R. A. [1998]. *Optimized Homomorphic Unsharp Masking for MR Grayscale Inhomogeneity Correction*. IEEE Trans. Medical Imaging, vol. 17, no. 2, pp. 161–171.

Brummer M. E. [1991]. *Hough Transform Detection of the Longitudinal Fissure in Tomographic Head Images*. IEEE Trans. Biomed. Images, vol. 10, no. 1, pp. 74–83.

Brzakovic D., Patton R., Wang R. [1991]. *Rule-Based Multi-template Edge Detection*. Computer Vision, Graphics, Image Processing: Graphical Models and Image Processing, vol. 53, no. 3, pp. 258–268.

Bunke H. Sanfeliu A. (eds.) [1990]. *Syntactic and Structural Pattern Recognition: Theory and Applications*, World Scientific, Teaneck, N.J.

Burrus C. S., Gopinath R. A., Guo H. [1998]. *Introduction to Wavelets and Wavelet Transforms*, Prentice Hall, Upper Saddle River, N J., pp. 250–251.

Burt P. J., Adelson E. H. [1983]. *The Laplacian Pyramid as a Compact Image Code*. IEEE Trans. Communications, vol. COM-31, no. 4, pp. 532–540.

Cameron J. P. [2005]. *Sets, Logic, and Categories*. Springer, New York.

Campbell J. D. [1969]. *Edge Structure and the Representation of Pictures*. Ph. D. dissertation, Dept. of Elec. Eng., University of Missouri, Columbia.

Candy J. C., Franke M. A., Haskell B. G., Mounts F. W. [1971]. *Transmitting Television as Clusters of Frame-to-Frame Differences*. Bell Sys. Tech. J., vol. 50, pp. 1889–1919.

Cannon T. M. [1974]. *Digital Image Deblurring by Non-Linear Homomorphic Filtering*. Ph. D. thesis, University of Utah.

Canny J. [1986]. *A Computational Approach for Edge Detection*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 8, no. 6, pp. 679–698.

Carey W. K., Chuang D. B., Hamami S. S. [1999]. *Regularity-Preserving Image Interpolation*. IEEE Trans. Image Processing, vol. 8, no. 9, pp. 1293–1299.

Caselles V., Lisani J.-L., Morel J.-M., Sapiro G. [1999]. *Shape Preserving Local Histogram Modification*. IEEE Trans. Image Processing, vol. 8, no. 2, pp. 220–230.

Castleman K. R. [1996]. *Digital Image Processing*, 2nd ed. Prentice Hall, Upper Saddle River, N.J.

Centeno J. A. S., Haertel V. [1997]. *An Adaptive Image Enhancement Algorithm*. *P*attern Recognition, vol. 30, no. 7 pp. 1183–1189.

Chan R. C., Karl W. C., Lees R. S. [2000]. *A New Model-Based Technique for Enhanced Small-Vessell Measurements in X-Ray Cine-Angiograms*. IEEE Trans. Medical Imaging, vol. 19, no. 3, pp. 243–255.

Chandler D., Hemami S. [2005]. *Dynamic Contrast-Based Quantization for Lossy Wavelet Image Compression*. IEEE Trans. Image Proc., vol. 14, no. 4, pp. 397–410.

Chang S. G., Yu B., Vetterli M. [2000]. *Spatially Adaptive Wavelet Thresholding with Context Modeling for Image Denoising*. IEEE Trans. Image Processing, vol. 9, no. 9, pp. 1522–1531.

Chang S. K. [1989]. *Principles of Pictorial Information Systems Design*, Prentice Hall, Englewood Cliffs, N.J.

Chang T., Kuo C.-C. J. [1993]. *Texture Analysis and Classification with Tree-Structures Wavelet Transforms*. IEEE Trans. Image Processing, vol. 2, no. 4, pp. 429–441.

Champeney D. C. [1987]. *A Handbook of Fourier Theorems*. Cambridge University Press, New York.

Chaudhuri B. B. [1983]. *A Note on Fast Algorithms for Spatial Domain Techniques in Image Processing*. IEEE Trans. Systems Man Cybernetic, vol. SMC-13, no. 6, pp. 1166–1169.

Chen M. C., Wilson A. N. [2000]. *Motion-Vector Optimization of Control Grid Interpolation and Overlapped Block Motion Compensation Using Iterated Dynamic Programming*. IEEE Trans. Image Processing., vol. 9, no. 7, pp. 1145–1157.

Chen Y.-S., Yu Y.-T. [1996]. *Thinning Approach for Noisy Digital Patterns*. Pattern Recognition, vol. 29, no. 11, pp. 1847–1862.

Cheng H. D., Huijuan Xu H. [2000]. *A Novel Fuzzy Logic Approach to Contrast Enhancement*. Pattern Recognition, vol. 33, no. 5, pp. 809–819.

Cheriet M., Said J. N., Suen C. Y. [1998]. *A Recursive Thresholding Technique for Image Segmentation*. IEEE Trans. Image Processing, vol. 7, no. 6, pp. 918–921.

Cheung J., Ferris D., Kurz L. [1997]. *On Classification of Multispectral Infrared Image Data*. IEEE Trans. Image Processing, vol. 6, no. 10, pp. 1456–1464.

Cheung K. K. T., Teoh E. K. [1999]. *Symmetry Detection by Generalized Complex (GC) Moments: A Closed-Form Solution*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 21, no. 5, pp. 466–476.

Chow C. K., Kaneko T. [1972]. *Automatic Boundary Detection of the Left Ventricle from Cineangiograms*. Comp., and Biomed. Res., vol. 5, pp. 388–410.

Chu C.-C., Aggarwal J.K. [1993]. *The Integration of Image Segmentation Maps Using Regions and Edge Information*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 15, no. 12, pp. 1241–1252.

CIE [1978]. *Uniform Color Spaces—Color Difference Equations—Psychometric Color Terms*, Commission Internationale de L'Eclairage, Publication No. 15, Supplement No. 2, Paris.

Clark J. J. [1989]. *Authenticating Edges Produced by Zero-Crossing Algorithms*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 12, no. 8, pp. 830–831.

Clarke R. J. [1985]. *Transform Coding of Images*, Academic Press, New York.

Cochran W. T., Cooley J. W., et al. [1967]. *What Is the Fast Fourier Transform?* IEEE Trans. Audio and Electroacoustics, vol. AU-15, no. 2, pp. 45–55. [Имеется перевод: Кокрэн и др. *Что такое быстрое преобразование Фурье?*. — ТИИЭР, 1967, т. 55, № 10.]

Coeurjolly D., Klette R. [2004]. *A Comparative Evaluation of Length Estimators of Digital Curves. IEEE Trans. Pattern. Analysis Machine Int.*, vol. 26, no. XX, pp. 252–258.

Cohen A., Daubechies I. [1992]. *A Stability Criterion for Biorthogonal Wavelet Bases and Their Related Subband Coding Schemes*, Technical report, AT&T Bell Laboratories.

Cohen A., Daubechies I., Feauveau J.-C. [1992]. *Biorthogonal Bases of Compactly Supported Wavelets*. Commun. Pure and Applied Mathematics, vol. 45, pp. 485–560.

Coifman R. R., Wickerhauser M. V. [1992]. *Entropy-Based Algorithms for Best Basis Selection*. IEEE Trans. Information Theory, vol. 38, no. 2, pp. 713–718.

Coltuc D., Bolon P., Chassery J. M. [2006]. *Exact Histogram Specification*. IEEE Trans. Image Processing, vol. 15, no. 5, pp. 1143–1152.

Cooley J. W., Lewis P. A. W., Welch P. D. [1967a]. *Historical Notes on the Fast Fourier Transform*. IEEE Trans. Audio and Electroacoustics, vol. AU-15, no. 2, pp. 76–79. [Имеется перевод той же статьи в другом журнале: Кули, Льюис, Уэлч. *Исторические замечания относительно быстрого преобразования Фурье.* — ТИИЭР, 1967, т. 55, № 10.]

Cooley J. W., Lewis P. A. W., Welch P. D. [1967b]. *Application of the Fast Fourier Transform to Computation of Fourier Integrals*. IEEE Trans. Audio and Electroacoustics, vol. AU-15, no. 2, pp. 79–84.

Cooley J. W., Lewis P. A. W., Welch P. D. [1969]. *The Fast Fourier Transform and Its Applications*. IEEE Trans. Educ., vol. E-12, no. 1. pp. 27–34.

Cooley J. W., Tukey J. W. [1965]. *An Algorithm for the Machine Calculation of Complex Fourier Series*. Math. of Comput., vol. 19, pp. 297–301.

Cornsweet T. N. [1970]. *Visual Perception*, Academic Press, New York.

Cortelazzo G. M., Lucchese L., Monti C. M. [1999]. *Frequency Domain Analysis of General Planar Rigid Motion with Finite Duration*. Journal Optical Society of America-A. Optics, Image Science, and Vision, vol. 16, no. 6, pp. 1238–1253.

Cowart A. E., Snyder W. E., Ruedger W. H. [1983]. *The Detection of Unresolved Targets Using the Hough Transform*. Computer Vision Graph Image Processing, vol. 21, pp. 222–238.

Cox I., Kilian J., Leighton F., Shamoon, T. [1997]. *Secure Spread Spectrum Watermarking for Multimedia*. IEEE Trans. Image Proc., vol. 6, no. 12, pp. 1673–1687.

Cox I., Miller M., Bloom J. [2001]. *Digital Watermarking*. Morgan Kaufmann (Elsevier), New York.

Creath K., Wyant J. C. [1992]. *Moire and Fringe Patterns*. In Optical Shop Testing, 2nd ed., (D. Malacara, ed.), John Wiley & Sons, New York, pp. 653–685.

Croisier A., Esteban D., Galand C. [1976]. *Perfect Channel Splitting by Use of Interpolation/Decimation/Tree Decomposition Techniques*. Int. Conf. On Inform. Sciences and Systems, Patras, Greece, pp. 443–446.

Cumani A., Guiducci A., Grattoni P. [1991]. *Image Description of Dynamic Scenes*. Pattern Recognition, vol. 24, no. 7, pp. 661–674.

Cutrona L. J., Hall W. D. [1968]. *Some Considerations in Post-Facto Blur Removal*. In Evaluation of Motion-Degraded Images, NASA Publ. SP-193, pp. 139–148.

Danielson G. C., Lanczos C. [1942]. *Some Improvements in Practical Fourier Analysis and Their Application to X-Ray Scattering from Liquids*. J. Franklin Institute, vol. 233, pp. 365–380, 435–452.

Daubechies I. [1988]. *Orthonormal Bases of Compactly Supported Wavelets. Commun*. On Pure and Applied Mathematics, vol. 41, pp. 909–996.

Daubechies I. [1990]. *The Wavelet Transform, Time-Frequency Localization and Signal Analysis*. IEEE Transactions on Information Theory, vol. 36, no. 5, pp. 961–1005.

Daubechies I. [1992].*Ten Lectures on Wavelets*, Society for Industrial and Applied Mathematics, Philadelphia, Pa. [Имеется перевод: Добеши И. *Десять лекций по вейвлетам*. — Ижевск: НИЦ «Регулярная и хаотическая динамика», 2001.]

Daubechies I. [1993]. *Orthonormal Bases of Compactly Supported Wavelets II, Variations on a Theme*. SIAM J. Mathematical Analysis, vol. 24, no. 2, pp. 499–519.

Daubechies I. [1996]. *Where Do We Go from Here? — A Personal Point of View*. Proc. IEEE, vol. 84, no. 4, pp. 510–513.

Daul C., Graebling P., Hirsch E. [1998]. *From the Hough Transform to a New Approach for the Detection of and Approximation of Elliptical Arcs*. Computer Vision and Image Understanding, vol. 72, no. 3, pp. 215–236.

Davies E. R. [2005]. *Machine Vision: Theory, Algorithms, Practicalities*. Morgan Kaufmann, San Francisco.

Davis L. S. [1982]. *Hierarchical Generalized Hough Transforms and Line-Segment Based Generalized Hough Transforms*. Pattern Recognition, vol. 15, no. 4, pp. 277–285.

Davis T. J. [1999]. *Fast Decomposition of Digital Curves into Polygons Using the Haar Transform*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 21, no. 8, pp. 786–790.

Davisson L. D. [1972]. *Rate-Distortion Theory and Application*. Proc. IEEE, vol. 60, pp. 800–808. [Имеется перевод: Дэвиссон. *Скорость создания сообщений. Теория и применение*. — ТИИЭР, 1972, т. 60, № 7.]

Delaney A. H., Bresler Y. [1995]. *Multiresolution Tomographic Reconstruction Using Wavelets*. IEEE Trans. Image Processing, vol. 4, no. 6, pp. 799–813.

Delon J., Desolneux A., Lisani J. L., Petro A. B. [2007]. *A Nonparametric Approach for Histogram Segmentation*. *I*EEE Trans. Image Proc., vol. 16, no. 1, pp. 253–261.

Delp E. J., Mitchell O. R. [1979]. *Image Truncation using Block Truncation Coding*. IEEE Trans. Communications, vol. COM-27, pp. 1335–1342.

Di Zenzo S. [1986]. *A Note on the Gradient of a Multi-Image*. Computer Vision, Graphics and Image Processing, vol. 33, pp. 116–125.

Dijkstra E. [1959]. *Note on Two Problems in Connection with Graphs*. Numerische Mathematik, vol. 1, pp. 269–271.

Djeziri S., Nouboud F., Plamondon R. [1998]. *Extraction of Signatures from Check Background Based on a Filiformity Criterion*. IEEE Trans. Image Processing, vol. 7, no. 102, pp. 1425–1438.

Dougherty E. R. [1992]. *An Introduction to Morphological Image Processing*, SPIE Press, Bellingham, Wash.

Dougherty E. R. (ed.) [2000]. *Random Processes for Image and Signal Processing*, IEEE Press, New York.

Dougherty E. R., Lotufo R. A. [2003]. *Hands-on Morphological Image Processing*. SPIE Press, Bellingham,WA.

Drew M. S., Wei J., Li Z.-N. [1999]. *Illumination Invariant Image Retrieval and Video Segmentation*. Pattern Recognition, vol. 32, no. 8, pp. 1369–1388.

Duda R. O., Hart P. E. [1972]. *Use of the Hough Transformation to Detect Lines and Curves in Pictures*. Comm. ACM, vol. 15, no. 1, pp. 11–15.

Duda R. O, Hart P. E., Stork D. G. [2001]. *Pattern Classification*, John Wiley & Sons, New York. [Имеется перевод более ранней книги: Дуда Р., Харт П. *Распознавание образов и анализ сцен*. — М.: Мир, 1976.]

Dugelay J., Roche S., Rey C., Doerr G. [2006]. *Still-Image Watermarking Robust to Local Geometric Distortions*. IEEE Trans. Image Proc., vol. 15, no. 9, pp. 2831–2842.

Edelman S. [1999]. *Representation and Recognition in Vision*, The MIT Press, Cambridge, Mass.

Elias P. [1952]. *Fourier Treatment of Optical Processes*. Journal Optical Society of America, vol. 42, no. 2, pp. 127–134.

Elliott D. F., Rao K. R. [1983]. *Fast Transforms: Algorithms and Applications*. Academic Press, New York.

Eng H.-L., Ma K.-K. [2001]. *Noise Adaptive Soft-Switching Median Filter*. IEEE Trans. Image Processing, vol. 10, no. 2, pp. 242–251.

Eng H.-L., Ma K.-K. [2006]. *A Switching Median Filter With Boundary Discriminitative Noise Detection for Extremely Corrupted Images*. IEEE Trans. Image Proc., vol. 15, no. 6, pp. 1506–1516.

Equitz W. H. [1989]. *A New Vector Quantization Clustering Algorithm*. IEEE Trans. Acoustic Speech Signal Processing, vol. ASSP-37, no. 10, pp. 1568–1575.

Etienne E. K., Nachtegael M. (eds.) [2000]. *Fuzzy Techniques in Image Processing*, Springer-Verlag, New York.

Evans A. N., Liu X. U. [2006]. *A Morphological Gradient Approach to Color Edge Detection*. IEEE Trans. Image Proc., vol. 15, no. 6, pp. 1454–1463.

Falconer D. G. [1970]. *Image Enhancement and Film Grain Noise*. Opt. Acta, vol. 17, pp. 693–705.

Fairchild M. D. [1998]. *Color Appearance Models*, Addison-Wesley, Reading, Mass.

Federal Bureau of Investigation [1993]. *WSQ Gray-Scale Fingerprint Image Compression Specification*, IAFIS-IC-0110v2, Washington, D. C.

Felsen L. B., Marcuvitz N. [1994]. *Radiation and Scattering of Waves*, IEEE Press, New York.

Ferreira A., Ubéda S. [1999]. *Computing the Medial Axis Transform in Parallel with Eight Scan Operators*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 21, no. 3, pp. 277–282.

Fischler M. A. [1980]. *Fast Algorithms for Two Maximal Distance Problems with Applications to Image Analysis*. Pattern Recognition, vol. 12, pp. 35–40.

Fisher R. A. [1936]. *The Use of Multiple Measurements in Taxonomic Problems*. Ann. Eugenics, vol. 7, Part 2, pp. 179–188. (Also in *Contributions to Mathematical Statistics*, John Wiley & Sons, New York, 1950.)

Flusser J. [2000]. *On the Independence of Rotation Moment Invariants*. Pattern Recognition, vol. 33, pp. 1405–1410.

Forsyth D. F., Ponce J. [2002]. *Computer Vision — A Modern Approach*. Prentice Hall, Upper Saddle River, NJ.

Fortner B., Meyer T. E. [1997]. *Number by Colors*, Springer-Verlag, New York. Fox, E. A. [1991]. *Advances in Interactive Digital Multimedia Systems*. Computer, vol. 24, no. 10, pp. 9–21.

Fram J. R., Deutsch E. S. [1975]. *On the Quantitative Evaluation of Edge Detection Schemes and Their Comparison with Human Performance*. IEEE Trans. Computers, vol. C-24, no. 6, pp. 616–628.

Freeman A. (translator) [1878]. J. Fourier, *The Analytical Theory of Heat*, Cambridge: University Press, London.

Freeman C. [1987]. *Imaging Sensors and Displays*, ISBN 0–89252–800–1, SPIE Press, Bellingham, Wash.

Freeman H. [1961]. *On the Encoding of Arbitrary Geometric Configurations*. IEEE Trans. Elec. Computers, vol. EC-10, pp. 260–268.

Freeman H. [1974]. *Computer Processing of Line Drawings*. Comput. Surveys, vol. 6, pp. 57–97.

Freeman H., Shapira R. [1975]. *Determining the Minimum-Area Encasing Rectangle for an Arbitrary Closed Curve*. Comm. ACM, vol. 18, no. 7, pp. 409–413.

Freeman J. A., Skapura D. M. [1991]. *Neural Networks: Algorithms, Applications, and Programming Techniques*, Addison-Wesley, Reading, Mass.

Frei W., Chen C. C. [1977]. *Fast Boundary Detection: A Generalization and a New Algorithm*. IEEE Trans. Computers, vol. C-26, no. 10, pp. 988–998.

Frendendall G. L., Behrend W. L. [1960]. *Picture Quality—Procedures for Evaluating Subjective Effects of Interference*. Proc. IRE, vol. 48, pp. 1030–1034.

Fu K. S. [1982]. *Syntactic Pattern Recognition and Applications*, Prentice Hall, Englewood Cliffs, N.J. [Имеется перевод более ранней книги: Фу К. *Структурные методы в распознавании образов*. — М.: Мир, 1977.]

Fu K. S., Bhargava B. K. [1973]. *Tree Systems for Syntactic Pattern Recognition*. IEEE Trans. Computers, vol. C-22, no. 12, pp. 1087–1099.

Fu K. S., Gonzalez R. C., Lee C. S. G. [1987]. *Robotics: Control, Sensing, Vision, and Intelligence*, McGraw-Hill, New York.

Fu K. S., Mui J. K. [1981]. *A Survey of Image Segmentation*. Pattern Recognition, vol. 13, no. 1, pp. 3–16.

Fukunaga K. [1972]. *Introduction to Statistical Pattern Recognition*. Academic Press, New York.

Furht B., Greenberg J., Westwater R. [1997]. *Motion Estimation Algorithms for Video Compression*. Kluwer Academic Publishers, Boston.

Gallager R., Voorhis D. V. [1975]. *Optimal Source Codes for Geometrically Distributed Integer Alphabets*. IEEE Trans. Inform. Theory, vol. IT-21, pp. 228–230.

Gao X., Sattar F., Vekateswarlu R. [2007]. *Multiscale Corner Detection of Gray Level Images Based on Log-Gabor Wavelet Transform*. IEEE Trans. Circuits and Systems for Video Technology, in press.

Garcia P. [1999]. *The Use of Boolean Model for Texture Analysis of Grey Images*. Computer Vision and Image Understanding, vol. 74, no. 3, pp. 227–235.

Gdalyahu Y., Weinshall D. [1999]. *Flexible Syntactic Matching of Curves and Its Application to Automated Hierarchical Classification of Silhouettes*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 21, no. 12, pp. 1312–1328.

Gegenfurtner K. R., Sharpe L. T. (eds.) [1999]. *Color Vision: From Genes to Perception*, Cambridge University Press, New York.

Geladi P., Grahn H. [1996]. *Multivariate Image Analysis*, John Wiley & Sons, New York.

Geman D., Reynolds G. [1992]. *Constrained Restoration and the Recovery of Discontinuities*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 14, no. 3, pp. 367–383.

Gentleman W. M. [1968]. *Matrix Multiplication and Fast Fourier Transformations*. Bell System Tech. J., vol. 47, pp. 1099–1103.

Gentleman W. M., Sande G. [1966]. *Fast Fourier Transform for Fun and Profit*. Fall Joint Computer Conf., vol. 29, pp. 563–578, Spartan, Washington, D. C.

Gharavi H., Tabatabai A. [1988]. *Sub-Band Coding of Monochrome and Color Images*. IEEE Trans. Circuits Sys., vol. 35, no. 2, pp. 207–214.

Giannakis G. B., Heath R. W., Jr. [2000]. *Blind Identification of Multichannel FIR Blurs and Perfect Image Restoration*. IEEE Trans. Image Processing, vol. 9, no. 11, pp. 1877–1896.

Giardina C. R., Dougherty E. R. [1988]. *Morphological Methods in Image and Signal Processing*, Prentice Hall, Upper Saddle River, N.J.

Golomb S. W. [1966]. *Run-Length Encodings*. IEEE Trans. Inform. Theory, vol. IT-12, pp. 399–401.

Gonzalez R. C. [1985]. *Computer Vision*. Yearbook of Science and Technology, McGraw-Hill, New York, pp. 128–132.

Gonzalez R. C. [1985]. *Industrial Computer Vision*. In Advances in Information Systems Science, Tou J. T. (ed.), Plenum, New York, pp. 345–385.

Gonzalez R. C. [1986]. *Image Enhancement and Restoration*. In Handbook of Pattern Recognition and Image Processing, Young T. Y., Fu K. S. (eds.), Academic Press, New York, pp. 191–213.

Gonzalez R. C., Edwards J. J., Thomason M. G. [1976]. *An Algorithm for the Inference of Tree Grammars*. Int. J. Comput. Info. Sci., vol. 5, no. 2, pp. 145–163.

Gonzalez R. C., Fittes B. A. [1977]. *Gray-Level Transformations for Interactive Image Enhancement*. Mechanism and Machine Theory, vol. 12, pp. 111–122.

Gonzalez R. C., Safabakhsh R. [1982]. *Computer Vision Techniques for Industrial Applications*. Computer, vol. 15, no. 12, pp. 17–32.

Gonzalez R. C., Thomason M. G. [1978]. *Syntactic Pattern Recognition: An Introduction*. Addison-Wesley, Reading, Mass.

Gonzalez R. C., Woods R. E. [1992]. *Digital Image Processing*. Addison-Wesley, Reading, Mass.

Gonzalez R. C., Woods R. E. [2002]. *Digital Image Processing*, 2nd ed., Prentice Hall, Upper Saddle River, NJ.

Gonzalez R. C., Woods R. E., Eddins S. L. [2004]. *Digital Image Processing Using MATLAB*, Prentice Hall, Upper Saddle River, NJ.

Good I. J. [1958]. *The Interaction Algorithm and Practical Fourier Analysis*. J. R. Stat. Soc. (Lond.), vol. B20, pp. 361–367; *Addendum*, vol. 22, 1960, pp. 372–375.

Goodson K. J., Lewis P. H. [1990]. *A Knowledge-Based Line Recognition System*. Pattern Recognition Letters, vol. 11, no. 4, pp. 295–304.

Gordon I. E. [1997]. *Theories of Visual Perception*, 2nd ed., John Wiley & Sons, New York.

Gori M., Scarselli F. [1998]. *Are Multilayer Perceptrons Adequate for Pattern Recognition and Verification?* IEEE Trans. Pattern Analysis Machine Intelligence, vol. 20, no. 11, pp. 1121–1132.

Goutsias J., Vincent L., Bloomberg D. S. (eds) [2000]. *Mathematical Morphology and Its Applications to Image and Signal Processing*, Kluwer Academic Publishers, Boston, Mass.

Graham R. E. [1958]. *Predictive Quantizing of Television Signals*. IRE Wescon Conv. Rec., vol. 2, pt. 2, pp. 147–157.

Graham R. L., Yao F. F. [1983]. *Finding the Convex Hull of a Simple Polygon*. J. Algorithms, vol. 4, pp. 324–331.

Gray R. M. [1984]. *Vector Quantization*. IEEE Trans. Acoustic Speech Signal Processing, vol. ASSP-1, no. 2, pp. 4–29.

Gröchenig K., Madych W. R. [1992]. *Multiresolution Analysis, Haar Bases and Self-Similar Tilings of R^n*. IEEE Trans. Information Theory, vol. 38, no. 2, pp. 556–568.

Grossman A., Morlet J. [1984]. *Decomposition of Hardy Functions into Square Integrable Wavelets of Constant Shape*. SIAM J. Applied Mathematics vol. 15, pp. 723–736.

Guil N., Villalba J., Zapata E. L. [1995]. *A Fast Hough Transform for Segment Detection. IEEE Trans. Image Processing*, vol. 4, no. 11, pp. 1541–1548.

Guil N., Zapata E. L. [1997]. *Lower Order Circle and Ellipse Hough Transform*. Pattern Recognition, vol. 30, no. 10, pp. 1729–1744.

Gunn S. R. [1998]. *Edge Detection Error in the Discrete Laplacian of a Gaussian*. Proc. 1998 Int'l Conference on Image Processing, vol. II, pp. 515–519.

Gunn S. R. [1999]. *On the Discrete Representation of the Laplacian of a Gaussian*. Pattern Recognition, vol. 32, no. 8, pp. 1463–1472.

Gupta L., Mohammad R. S., Tammana R. [1990]. *A Neural Network Approach to Robust Shape Classification*. Pattern Recognition, vol. 23, no. 6, pp. 563–568.

Gupta L., Srinath M. D. [1988]. *Invariant Planar Shape Recognition Using Dynamic Alignment*. Pattern Recognition, vol. 21, pp. 235–239.

Gupta L., Wang J., Charles A., Kisatsky P. [1994]. *Three-Layer Perceptron Based Classifiers for the Partial Shape Classification Problem*. Pattern Recognition, vol. 27, no. 1, pp. 91–97.

Haar A. [1910]. *Zur Theorie der Orthogonalen Funktionensysteme*. Math. Annal., vol. 69, pp. 331–371.

Habibi A. [1971]. *Comparison of Nth Order DPCM Encoder with Linear Transformations and Block Quantization Techniques*. IEEE Trans. Comm. Tech., vol. COM-19, no. 6, pp. 948–956.

Habibi A. [1974]. *Hybrid Coding of Pictorial Data*. IEEE Trans. Communications, vol. COM-22, no. 5, pp. 614–624.

Haddon J. F., Boyce J. F. [1990]. *Image Segmentation by Unifying Region and Boundary Information*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 12, no. 10, pp. 929–948.

Hall E. L. [1979]. *Computer Image Processing and Recognition*, Academic Press, New York.

Hamming R. W. [1950]. *Error Detecting and Error Correcting Codes*. Bell Sys. Tech. J., vol. 29, pp. 147–160. [Имеется перевод: Хэмминг Р. В. *Коды с обнаружением и исправлением ошибок*. — В кн.: Коды с обнаружением и исправлением ошибок. — М.: ИЛ, 1956.]

Hannah I., Patel D., Davies R. [1995]. *The Use of Variance and Entropy Thresholding Methods for Image Segmentation*. Pattern Recognition, vol. 28, no. 8, pp. 1135–1143.

Haralick R. M., Lee J. S. J. [1990]. *Context Dependent Edge Detection and Evaluation*. *Pattern Recognition*, vol. 23, no. 1–2, pp. 1–20.

Haralick R. M., Shapiro L. G. [1985]. *Survey: Image Segmentation*. Computer Vision, Graphics, Image Processing, vol. 29, pp. 100–132.

Haralick R. M., Shapiro L. G. [1992]. *Computer and Robot Vision*, vols. 1 & 2, Addison-Wesley, Reading, Mass.

Haralick R. M., Sternberg S. R., Zhuang X. [1987]. *Image Analysis Using Mathematical Morphology*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. PAMI-9, no. 4, pp. 532–550.

Haralick R. M., Shanmugan R., Dinstein I. [1973]. *Textural Features for Image Classification*. IEEE Trans Syst. Man Cyb., vol. SMC-3, no. 6, pp. 610–621.

Harikumar G., Bresler Y. [1999]. *Perfect Blind Restoration of Images Blurred by Multiple Filters: Theory and Efficient Algorithms*. IEEE Trans. Image Processing, vol. 8, no. 2, pp. 202–219.

Harmuth H. F. [1970]. *Transmission of Information by Orthogonal Signals*, Springer-Verlag, New York. [Имеется перевод: Хармут Х. Ф. *Передача информации ортогональными функциями*. — М.: Связь, 1975.]

Haris K., Efstratiadis S.N., Maglaveras N., Katsaggelos A. K. [1998]. *Hybrid Image Segmentation Using Watersheds and Fast Region Merging*. IEEE Trans. Image Processing, vol. 7, no. 12, pp. 1684–1699.

Hart P. E., Nilsson N. J., Raphael B. [1968]. *A Formal Basis for the Heuristic Determination of Minimum-Cost Paths*. IEEE Trans. Syst. Man Cyb, vol. SMC-4, pp. 100–107.

Hartenstein H., Ruhl M., Saupe D. [2000]. *Region-Based Fractal Image Compression*. IEEE Trans. Image Processing, vol. 9, no. 7, pp. 1171–1184.

Haskell B. G., Netravali A. N. [1997]. *Digital Pictures: Representation, Compression, and Standards*, Perseus Publishing, New York.

Haykin S. [1996]. *Adaptive Filter Theory*, Prentice Hall, Upper Saddle River, N.J.

Healy D. J., Mitchell O. R. [1981]. *Digital Video Bandwidth Compression Using Block Truncation Coding*. IEEE Trans. Communications, vol. COM-29, no. 12, pp. 1809–1817.

Heath M. D., Sarkar S., Sanocki T., Bowyer K. W. [1997]. *A Robust Visual Method for Assessing the Relative Performance of Edge-Detection Algorithms*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 19, no. 12, pp. 1338–1359.

Heath M., Sarkar S., Sanoki T., Bowyer K. [1998]. *Comparison of Edge Detectors: A Methodology and Initial Study*. Computer Vision and Image Understanding, vol. 69, no. 1, pp. 38–54.

Hebb D. O. [1949]. *The Organization of Behavior: A Neuropsychological Theory*, John Wiley & Sons, New York.

Heijmans H. J. A. M., Goutsias J. [2000]. *Nonlinear Multiresolution Signal Decomposition Schemes — Part II: Morphological Wavelets*. IEEE Trans. Image Processing, vol. 9, no. 11, pp. 1897–1913.

Highnam R., Brady M. [1997]. *Model-Based Image Enhancement of Far Infrared Images*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 19, no. 4, pp. 410–415.

Hojjatoleslami S. A., Kittler J. [1998]. *Region Growing: A New Approach*. IEEE Trans. Image Processing, vol. 7, no. 7, pp. 1079–1084.

Hong Pi, Hung Li, Hua Li [2006]. *A Novel Fractal Image Watermarking*. IEEE Trans. Multimedia, vol. 8, no. 3, pp. 488–499.

Hoover R. B., Doty F. [1996]. *Hard X-Ray/Gamma-Ray and Neutron Optics, Sensors, and Applications*, ISBN 0-8194-2247-9, SPIE Press, Bellingham, Wash.

Horn B. K. P. [1986]. *Robot Vision*, McGraw-Hill, New York. [Имеется перевод: Хорн Б. К. П. *Зрение роботов*. — М.: Мир, 1989.]

Hotelling H. [1933]. *Analysis of a Complex of Statistical Variables into Principal Components*. J. Educ. Psychol., vol. 24, pp. 417–441, 498–520.

Hough P. V. C. [1962]. *Methods and Means for Recognizing Complex Patterns*. U.S. Patent 3,069,654.

Hsu C. C., Huang, J. S. [1990]. *Partitioned Hough Transform for Ellipsoid Detection*. Pattern Recognition, vol. 23, no. 3–4, pp. 275–282.

Hu J., Yan H. [1997]. *Polygonal Approximation of Digital Curves Based on the Principles of Perceptual Organization*. Pattern Recognition, vol. 30, no. 5, pp. 701–718.

Hu M.K. [1962]. *Visual Pattern Recognition by Moment Invariants*. IRE Trans. Info. Theory, vol. IT-8, pp. 179–187.

Hu Y., Kwong S., Huang J. [2006]. *An Algorithm for Removable Visible Watermarking*. IEEE Trans. Circuits and Systems for Video Technology, vol. 16, no. 1, pp. 129–133.

Huang K.-Y. [2002]. *Syntactic Pattern Recognition for Seismic Oil Exploration*, World Scientific, Hackensack, NJ.

Huang S.-C., Sun Y.-N. [1999]. *Polygonal Approximation Using Generic Algorithms*. Pattern Recognition, vol. 32, no. 8, pp. 1409–1420.

Huang T. S. [1965]. *PCM Picture Transmission*. IEEE Spectrum, vol. 2, no. 12, pp. 57–63.

Huang T. S. [1966]. *Digital Picture Coding*. Proc. Natl. Electron. Conf., pp. 793–797.

Huang T. S., ed. [1975]. *Picture Processing and Digital Filtering*, Springer-Verlag, New York. [Имеется перевод: *Обработка изображений и цифровая фильтрация*. ред. Т. Хуанг. — М.: Мир, 1979.]

Huang T. S. [1981]. *Image Sequence Analysis*, Springer-Verlag, New York.

Huang T. S., Hussian A. B. S. [1972]. *Facsimile Coding by Skipping White*. IEEE Trans. Communications, vol. COM-23, no. 12, pp. 1452–1466.

Huang T. S., Tretiak O. J. (eds.). [1972]. *Picture Bandwidth Compression*, Gordon and Breech, New York.

Huang T. S., Yang G. T., Tang G. Y. [1979]. *A Fast Two-Dimensional Median Filtering Algorithm*. IEEE Trans. Acoustics, Speech, Signal Processing, vol. ASSP-27, pp. 13–18.

Huang Y., Schultheiss P. M. [1963]. *Block Quantization of Correlated Gaussian Random Variables*. IEEE Trans. Communication Systems, vol. CS-11, pp. 289–296.

Hubbard B. B. [1998]. *The World According to Wavelets-The Story of a Mathematical Technique in the Making*, 2nd ed., A. K. Peters, Ltd., Wellesley, Mass.

Hubel D. H. [1988]. *Eye, Brain, and Vision*, Scientific Amer. Library, W. H. Freeman, New York.

Huertas A., Medione G. [1986]. *Detection of Intensity Changes with Subpixel Accuracy using Laplacian-Gaussian Masks*. IEEE Trans. Pattern. Analysis Machine Intelligence, vol. PAMI-8, no. 5, pp. 651–664.

Huffman D. A. [1952]. *A Method for the Construction of Minimum Redundancy Codes*. Proc. IRE, vol. 40, no. 10, pp. 1098–1101. [Имеется перевод: Метод построения кодов с минимальной избыточностью. — В кн.: *Кибернетический сборник*. — М.: ИЛ, 1961, № 3, с. 37–54.]

Hufnagel R. E., Stanley N. R. [1964]. *Modulation Transfer Function Associated with Image Transmission Through Turbulent Media*. Journal Optical Society of America, vol. 54, pp. 52–61.

Hummel R. A. [1974]. *Histogram Modification Techniques*. Technical Report TR-329. F-44620–72C-0062, Computer Science Center, University of Maryland, College Park, Md.

Hunt B. R. [1971]. *A Matrix Theory Proof of the Discrete Convolution Theorem*. IEEE Trans. Audio and Electroacoustics, vol. AU-19, no. 4, pp. 285–288.

Hunt B. R. [1973]. *The Application of Constrained Least Squares Estimation to Image Restoration by Digital Computer*. IEEE Trans. Computers, vol. C-22, no. 9, pp. 805–812.

Hunter R., Robinson A. H. [1980]. *International Digital Facsimile Coding Standards*. Proc. IEEE, vol. 68, no. 7, pp. 854–867. [Имеется перевод: *Международные стандарты кодирования для цифровой факсимильной связи*. — ТИИЭР, 1980, т. 68, № 7, с. 112–129.]

Hurn M., Jennison C. [1996]. *An Extension of Geman and Reynolds' Approach to Constrained Restoration and the Recovery of Discontinuities*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 18, no. 6, pp. 657–662.

Hwang H., Haddad R.A. [1995]. *Adaptive Median Filters: New Algorithms and Results*. IEEE Trans. Image Processing, vol. 4, no. 4, pp. 499–502.
IEEE Computer [1974]. *Special issue on digital image processing*. vol. 7, no. 5.
IEEE Computer [1988]. *Special issue on artificial neural systems*. vol. 21, no. 3.
IEEE Trans. Circuits and Systems [1975]. *Special issue on digital filtering and image processing*, vol. CAS-2, pp. 161–304.
IEEE Trans. Computers [1972]. *Spec. issue on two-dimensional signal processing*, vol. C-21, no. 7.
IEEE Trans. Communication [1981]. *Special issue on picture communication systems*, vol. COM-29, no. 12.
IEEE Trans. Image Processing [1994]. *Special issue on image sequence compression*, vol. 3, no. 5.
IEEE Trans. Image Processing [1996]. *Special issue on vector quantization*, vol. 5, no. 2.
IEEE Trans. Image Processing [1996]. *Special issue on nonlinear image processing*, vol. 5, no. 6.
IEEE Trans. Image Processing [1997]. *Special issue on automatic target detection*, vol. 6, no. 1.
IEEE Trans. Image Processing [1997]. *Special issue on color imaging*, vol. 6, no. 7.
IEEE Trans. Image Processing [1998]. *Special issue on applications of neural networks to image processing*, vol. 7, no. 8.
IEEE Trans. Information Theory [1992]. *Special issue on wavelet transforms and mulitresolution signal analysis*, vol. 11, no. 2, Part II.
IEEE Trans. Pattern Analysis and Machine Intelligence [1989]. *Special issue on multiresolution processing*, vol. 11, no. 7.
IEEE Trans. Signal Processing [1993]. *Special issue on wavelets and signal processing*, vol. 41, no. 12.
IES Lighting Handbook, 9th ed. [2000]. *Illuminating Engineering Society Press*, New York.
ISO/IEC [1999]. *ISO/IEC 14495-1:1999: Information technology — Lossless and near-lossless compression of continuous-tone still images: Baseline*.
ISO/IEC JTC 1/SC 29/WG 1 [2000]. *ISO/IEC FCD 15444-1: Information technology — JPEG 2000 image coding system: Core coding system*.
Jähne B. [1997]. *Digital Image Processing: Concepts, Algorithms, and Scientific Applications*, Springer-Verlag, New York.
Jähne, B. [2002]. *Digital Image Processing*, 5th ed., Springer, New York.
Jain A.K. [1981]. *Image Data Compression: A Review*. Proc. IEEE, vol. 69, pp. 349–389. [Имеется перевод: Джайн А.К. *Сжатие видеоинформации*. — ТИИЭР, 1981, т. 69, № 3.]
Jain A.K. [1989]. *Fundamentals of Digital Image Processing*, Prentice Hall, Englewood Cliffs, N.J.
Jain A.K., Duin R.P.W., Mao J. [2000]. *Statistical Pattern Recognition: A Review*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 22, no. 1, pp. 4–37.
Jain J.R., Jain A.K. [1981]. *Displacement Measurement and Its Application in Interframe Image Coding*. IEEE Trans. Communications, vol. COM-29, pp. 1799–1808.
Jain R. [1981]. *Dynamic Scene Analysis Using Pixel-Based Processes*. Computer, vol. 14, no. 8, pp. 12–18.
Jain R., Kasturi R., Schunk B. [1995]. *Computer Vision*, McGraw-Hill, New York.

Jang B. K., Chin, R.T. [1990]. *Analysis of Thinning Algorithms Using Mathematical Morphology*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 12, no. 6, pp. 541–551.

Jayant N. S. (ed.) [1976]. *Waveform Quantization and Coding*, IEEE Press, New York.

Jones R., Svalbe I. [1994]. *Algorithms for the Decomposition of Gray-Scale Morphological Operations*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 16, no. 6, pp. 581–588.

Jonk A., van den Boomgaard S., Smeulders A. [1999]. *Grammatical Inference of Dashed Lines*. Computer Vision and Image Understanding, vol. 74, no. 3, pp. 212–226.

Kahaner D. K. [1970]. *Matrix Description of the Fast Fourier Transform*. IEEE Trans. Audio Electroacoustics, vol. AU-18, no. 4, pp. 442–450.

Kak A. C., Slaney M. [2001]. *Principles of Computerized Tomographic Imaging*, Published by the Society for Industrial and Applied Mathematics, Philadelphia, Pa.

Kamstra L., Heijmans H. J. A. M. [2005]. *Reversible Data Embedding Into Images Using Wavelet Techniques and Sorting*. IEEE Trans. Image Processing, vol. 14, no. 12, pp. 2082–2090.

Karhunen K. [1947]. *Über Lineare Methoden in der Wahrscheinlichkeitsrechnung*. Ann. Acad. Sci. Fennicae, Ser. A137. (Translated by I. Selin in «On Linear Methods in Probability Theory». T-131, 1960, The RAND Corp., Santa Monica, Calif.)

Kasson J., Plouffe W. [1992]. *An Analysis of Selected Computer Interchange Color Spaces*." ACM Trans. on Graphics, vol. 11, no. 4, pp. 373–405.

Katzir N., Lindenbaum M., Porat M. [1994]. *Curve Segmentation Under Partial Occlusion*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 16, no. 5, pp. 513–519.

Kerre E. E., Nachtegael M., eds. [2000]. *Fuzzy Techniques in Image Processing*. Springer-Verlag, New York.

Khanna T. [1990]. *Foundations of Neural Networks*, Addison-Wesley, Reading, Mass.

Kim C. [2005]. *Segmenting a Low-Depth-of-Filed Image Using Morphological Filters and Region Merging*. IEEE Trans. Image Proc., vol. 14, no. 10, pp. 1503–1511.

Kim J. K., Park J. M., Song K. S., Park H. W. [1997]. *Adaptive Mammographic Image Enhancement Using First Derivative and Local Statistics*. IEEE Trans. Medical Imaging, vol. 16, no. 5, pp. 495–502.

Kimme C., Ballard D. H., Sklansky J. [1975]. *Finding Circles by an Array of Accumulators*. Comm. ACM, vol. 18, no. 2, pp. 120–122.

Kirsch R. [1971]. *Computer Determination of the Constituent Structure of Biological Images*. Comput. Biomed. Res., vol. 4, pp. 315–328.

Kiver M. S. [1965]. *Color Television Fundamentals*, McGraw-Hill, New York.

Klette R., Rosenfeld A. [2004]. *Digital Geometry—Geometric Methods for Digital Picture Analysis*. Morgan Kaufmann, San Francisco.

Klinger A. [1976]. *Experiments in Picture Representation Using Regular Decomposition*. Computer Graphics Image Processing, vol. 5, pp. 68–105.

Knowlton K. [1980]. *Progressive Transmission of Gray-Scale and Binary Pictures by Simple, Efficient, and Lossless Encoding Schemes*. Proc. IEEE, vol. 68, no. 7, pp. 885–896. [Имеется перевод: Ноултон К. *Простые, эффективные методы кодирования без потерь для передачи многоуровневых и двухуровневых изображений с постепенным воспроизведением*. — ТИИЭР, 1980, т. 68, № 7, с. 149–162.]

Kohler R. J., Howell H. K. [1963]. *Photographic Image Enhancement by Superposition of Multiple Images*. Photogr. Sci. Eng., vol. 7, no. 4, pp. 241–245.

Kokaram A. [1998]. *Motion Picture Restoration*, Springer-Verlag, New York.

Kokare M., Biswas P., Chatterji B. [2005]. *Texture Image Retrieval Using New Rotated Complex Wavelet Filters*. IEEE Trans. Systems, Man, Cybernetics, Part B, vol. 35, no. 6, pp. 1168–1178.

Kramer H. P., Mathews M. V. [1956]. *A Linear Coding for Transmitting a Set of Correlated Signals*. IRE Trans. Info. Theory, vol. IT-2, pp. 41–46.

Langdon G. C., Rissanen J. J. [1981]. *Compression of Black–White Images with Arithmetic Coding*. IEEE Trans. Communications, vol. COM-29, no. 6, pp. 858–867.

Lantuéjoul C. [1980]. *Skeletonization in Quantitative Metallography*. In Issues of Digital Image Processing, Haralick R. M., Simon J. C. (eds.), Sijthoff and Noordhoff, Groningen, The Netherlands.

Latecki L. J., Lakämper R. [1999]. *Convexity Rule for Shape Decomposition Based on Discrete Contour Evolution*. Computer Vision and Image Understanding, vol. 73, no. 3, pp. 441–454.

Le Gall D., Tabatabai A. [1988]. *Sub-band Coding of Digital Images Using Symmetric Short Kernel Filters and Arithmetic Coding Techniques*. IEEE International Conference on Acoustics, Speech, and Signal Processing, New York, NY, pp. 761–765.

Ledley R. S. [1964]. *High-Speed Automatic Analysis of Biomedical Pictures*. Science, vol. 146, no. 3461, pp. 216–223.

Lee J.-S., Sun Y.-N., Chen C.-H. [1995]. *Multiscale Corner Detection by Using Wavelet Transforms*. IEEE Trans. Image Processing, vol. 4, no. 1, pp. 100–104.

Lee S. U., Chung S. Y., Park R. H. [1990]. *A Comparative Performance Study of Several Global Thresholding Techniques for Segmentation*. Computer Vision, Graphics, Image Processing, vol. 52, no. 2, pp. 171–190.

Lehmann T. M., Gönner C., Spitzer K. [1999]. *Survey: Interpolation Methods in Medical Image Processing*. IEEE Trans. Medical Imaging, vol. 18, no. 11, pp. 1049–1076.

Lema M. D., Mitchell O. R. [1984]. *Absolute Moment Block Truncation Coding and Its Application to Color Images*. IEEE Trans. Communications, vol. COM-32, no. 10. pp. 1148–1157.

Levine M. D. [1985]. *Vision in Man and Machine*, McGraw-Hill, New York.

Liang K.-C., Kuo C.-C. J. [1991]. *Waveguide: A Joint Wavelet-Based Image Representation and Description System*. IEEE Trans. Image Processing, vol. 8, no. 11, pp. 1619–1629.

Liang Q., Wendelhag J. W., Gustavsson T. [2000]. *A Multiscale Dynamic Programming Procedure for Boundary Detection in Ultrasonic Artery Images*. IEEE Trans. Medical Imaging, vol. 19, no. 2, pp. 127–142.

Liao P., Chen T., Chung P. [2001]. *A Fast Algorithm for Multilevel Thresholding*. J. Inform. Sc. and Eng., vol. 17, pp. 713–727.

Lillesand T. M., Kiefer R. W. [1999]. *Remote Sensing and Image Interpretation*, John Wiley & Sons, New York.

Lim J. S. [1990]. *Two-Dimensional Signal and Image Processing*, Prentice Hall, Upper Saddle River, N.J.

Limb J. O., Rubinstein C. B. [1978]. *On the Design of Quantizers for DPCM Coders: A Functional Relationship Between Visibility, Probability, and Masking*. IEEE Trans. Communications, vol. COM-26, pp. 573–578.

Lindblad T., Kinser J. M. [1998]. *Image Processing Using Pulse-Coupled Neural Networks*, Springer-Verlag, New York.

Linde Y., Buzo A., Gray R. M. [1980]. *An Algorithm for Vector Quantizer Design*. IEEE Trans. Communications, vol. COM-28, no. 1, pp. 84–95.

Lippmann R. P. [1987]. *An Introduction to Computing with Neural Nets*. IEEE ASSP Magazine, vol. 4, pp. 4–22.

Liu J., Yang Y.-H. [1994]. *Multiresolution Color Image Segmentation*. IEEE Trans Pattern Analysis Machine Intelligence, vol. 16, no. 7, pp. 689–700.

Liu-Yu S., Antipolis M. [1993]. *Description of Object Shapes by Apparent Boundary and Convex Hull*. Pattern Recognition, vol. 26, no. 1, pp. 95–107.

Lo R.-C., Tsai W.-H. [1995]. *Gray-Scale Hough Transform for Thick Line Detection in Gray-Scale Images*. Pattern Recognition, vol. 28, no. 5, pp. 647–661.

Loncaric S. [1998]. *A Survey of Shape Analysis Techniques*. Pattern Recognition, vol. 31, no. 8, pp. 983–1010.

Lu H. E., Wang P. S. P. [1986]. *A Comment on A Fast Parallel Algorithm for Thinning Digital Patterns*. Comm. ACM, vol. 29, no. 3, pp. 239–242.

Lu N. [1997]. *Fractal Imaging*, Academic Press, New York.

Lu W.-S and Antoniou A. [1992]. *Two-Dimensional Digital Filters*. Marcel Dekker, New York.

MacAdam D. L. [1942]. *Visual Sensitivities to Color Differences in Daylight*. Journal Optical Society of America, vol. 32, pp. 247–274.

MacAdam D. P. [1970]. *Digital Image Restoration by Constrained Deconvolution*. Journal Optical Society of America, vol. 60, pp. 1617–1627.

Maki A., Nordlund P., Eklundh J.-O. [2000]. *Attentional Scene Segmentation: Integrating Depth and Motion*. Computer Vision and Image Understanding, vol. 78, no. 3, pp. 351–373.

Malacara D. [2001]. *Color Vision and Colorimetry: Theory and Applications*, SPIE Optical Engineering Press, Bellingham, Wash.

Mallat S. [1987]. *A Compact Multiresolution Representation: The Wavelet Model*. Proc. IEEE Computer Society Workshop on Computer Vision, IEEE Computer Society Press, Washington, D. C., pp. 2–7.

Mallat S. [1989a]. *A Theory for Multiresolution Signal Decomposition: The Wavelet Representation*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. PAMI-11, pp. 674–693.

Mallat S. [1989b]. *Multiresolution Approximation and Wavelet Orthonormal Bases of L2*. Trans. American Mathematical Society, vol. 315, pp. 69–87.

Mallat S. [1989c]. *Multifrequency Channel Decomposition of Images and Wavelet models*. IEEE Trans. Acoustics, Speech, and Signal Processing, vol. 37, pp. 2091–2110.

Mallat S. [1998]. *A Wavelet Tour of Signal Processing*, Academic Press, Boston, Mass.

Mallat S. [1999]. *A Wavelet Tour of Signal Processing*. 2nd ed., Academic Press, San Diego, Calif.

Mallot A. H. [2000]. *Computational Vision*, The MIT Press, Cambridge, Mass.

Mamistvalov A. [1998]. *n-Dimensional Moment Invariants and Conceptual Mathematical Theory of Recognition [of] n-Dimensional Solids*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 20, no. 8, pp. 819–831.

Manjunath B., Salembier P., Sikora T. [2001]. *Introduction to MPEG-7*, John Wiley & Sons, West Sussex, UK.

Maragos P. [1987]. *Tutorial on Advances in Morphological Image Processing and Analysis*. Optical Engineering, vol. 26, no. 7, pp. 623–632.

Marchand-Maillet S., Sharaiha Y. M. [2000]. *Binary Digital Image Processing: A Discrete Approach*, Academic Press, New York.

Maren A. J., Harston C. T., Pap R. M. [1990]. *Handbook of Neural Computing Applications*, Academic Press, New York.

Marr D. [1982]. *Vision*, Freeman, San Francisco. [Имеется перевод: Марр Д. *Зрение. Информационный подход к изучению представления и обработки зрительных образов.* — М.: Радио и связь, 1987.]

Marr D., Hildreth E. [1980]. *Theory of Edge Detection*. Proc. Royal Society London, vol. B207, pp. 187–217.

Martelli A. [1972]. *Edge Detection Using Heuristic Search Methods*. Computer Graphics Image Processing, vol. 1, pp. 169–182.

Martelli A. [1976]. *An Application of Heuristic Search Methods to Edge and Contour Detection*. Comm. ACM, vol. 19, no. 2, pp. 73–83.

Martin M.B., Bell, A.E. [2001]. *New image compression techniques using multiwavelets and multiwavelet packets*. IEEE Trans. on Image Processing, vol. 10, no. 4, pp. 500–510.

Mather P. M. [1999]. *Computer Processing of Remotely Sensed Images: An Introduction*, John Wiley & Sons, New York.

Max J. [1960]. *Quantizing for Minimum Distortion*. IRE Trans. Info. Theory, vol. IT-6, pp. 7–12.

McClelland J. L., Rumelhart D. E. (eds.) [1986]. *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, vol. 2: *Psychological and Biological Models*, The MIT Press, Cambridge, Mass.

McCulloch W. S., Pitts W. H. [1943]. *A Logical Calculus of the Ideas Imminent in Nervous Activity.* Bulletin of Mathematical Biophysics, vol. 5, pp. 115–133. [Имеется перевод: Мак-Каллок У. С., Питтс В. *Логическое исчисление идей, относящихся к нервной активности./Автоматы.* ред. Шеннон К. Э., Маккарти Дж. — М.: ИЛ, 1956, с. 362–384.]

McFarlane M. D. [1972]. *Digital Pictures Fifty Years Ago.* Proc. IEEE, vol. 60, no. 7, pp. 768–770. [Имеется перевод: Макфарлэйн. *Дискретная передача изображений пятьдесят лет назад.* — ТИИЭР, 1972, т, 60, № 7, с. 15–30]

McGlamery B. L. [1967]. *Restoration of Turbulence-Degraded Images.* Journal Optical Society of America, vol. 57, no. 3, pp. 293–297.

Meijering H. W., Zuiderveld K. J., Viergever M. A. [1999]. *Image Registration for Digital Subtraction Angiography.* Int. J. Comput.Vision, vol. 31, pp. 227–246.

Meijering E. H. W., Niessen W. J., Viergever M. A. [1999]. *Retrospective Motion Correction in Digital Subtraction Angiography: A Review.* IEEE Trans. Medical Imaging, vol. 18, no. 1, pp. 2–21. [chap. 3]

Meijering E. H. W., et al. [2001]. *Reduction of Patient Motion Artifacts in Digital Subtraction Angiography: Evaluation of a Fast and Fully Automatic Technique*. Radiology, vol. 219, pp. 288–293.

Memon N., Neuhoff D. L., Shende S. [2000]. *An Analysis of Some Common Scanning Techniques for Lossless Image Coding.* IEEE Trans. Image Processing, vol. 9, no. 11, pp. 1837–1848.

Mesarović V. Z. [2000]. *Iterative Linear Minimum Mean-Square-Error Image Restoration from Partially Known Blur.* Journal Optical Society of America-A. Optics, Image Science, and Vision, vol. 17, no. 4, pp. 711–723.

Meyer Y. [1987]. *L'analyses par Ondelettes.* Pour la Science.

Meyer Y. [1990]. *Ondelettes et ope'rateurs*, Hermann, Paris.
Meyer Y. (ed.) [1992a]. *Wavelets and Applications: Proceedings of the International Conference, Marseille, France*, Mason, Paris, and Springer-Verlag, Berlin.
Meyer Y. (translated by D. H. Salinger) [1992b]. *Wavelets and Operators*, Cambridge University Press, Cambridge, UK.
Meyer Y. (translated by Ryan R. D.) [1993]. *Wavelets: Algorithms and Applications*, Society for Industrial and Applied Mathematics, Philadelphia.
Meyer F. G., Averbuch A. Z., Strömberg J.-O. [2000]. *Fast Adaptive Wavelet Packet Image Compression*. IEEE Trans. Image Processing, vol. 9, no. 7, pp. 792–800.
Meyer F., Beucher S. [1990]. *Morphological Segmentation*. J. Visual Communications and Image Representation, vol. 1, no. 1, pp. 21–46.
Meyer H., Rosdolsky H. G., Huang T. S. [1973]. *Optimum Run Length Codes*. IEEE Trans. Communications, vol. COM-22, no. 6, pp. 826–835.
Minsky M., Papert S. [1969]. *Perceptrons: An Introduction to Computational Geometry*, the MIT Press, Cambridge, Mass. [Имеется перевод: Минский М., Пейперт С. *Персептроны*. — М.: Мир, 1971.]
Mirmehdi M., Petrou M. [2000]. *Segmentation of Color Textures*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 22, no. 2, pp. 142–159.
Misiti M., Misiti Y., Oppenheim G., Poggi J.-M. [1996]. *Wavelet Toolbox User's Guide*, The MathWorks, Inc., Natick, Mass.
Mitchell D. P., Netravali A. N. [1988]. *Reconstruction Filters in Computer Graphics*. Comp. Graphics, vol. 22, no. 4, pp. 221–228.
Mitchell J., Pennebaker W., Fogg C., LeGall D. [1997]. *MPEG Video Compression Standard*. Chapman & Hall, New York.
Mitiche A. [1994]. *Computational Analysis of Visual Motion*, Perseus Publishing, New York.
Mitra S., Murthy C., Kundu M. [1998]. *Technique for Fractal Image Compression Using Genetic Algorithm*. IEEE Transactions on Image Processing, vol. 7, no. 4, pp. 586–593.
Mitra S. K., Sicuranza G. L. (eds.) [2000]. *Nonlinear Image Processing*, Academic Press, New York.
Mohanty S., et al. [1999]. *A Dual Watermarking Technique for Images*. Proc. 7th ACM International Multimedia Conference, ACM-MM'99, Part 2, pp. 49–51.
Moore G. A. [1968]. *Automatic Scanning and Computer Processes for the Quantitative Analysis of Micrographs and Equivalent Subjects*. In Pictorial Pattern Recognition, (G.C. Cheng et al. eds), pp. 275–326, Thomson, Washington, D.C.
Mukherjee D., Mitra, S. [2003]. *Vector SPIHT for Embedded Wavelet Video and Image Coding*. IEEE Trans. Circuits and Systems for Video Technology, vol. 13, no. 3, pp. 231–246.
Murase H., Nayar S. K. [1994]. *Illumination Planning for Object Recognition Using Parametric Eigen Spaces*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 16, no. 12, pp. 1219–1227.
Murino V., Ottonello C., Pagnan S. [1998]. *Noisy Texture Classfication: A Higher-Order Statistical Approach*. Pattern Recognition, vol. 31, no. 4, pp. 383–393.
Nagao M., Matsuyama T. [1980]. *A Structural Analysis of Complex Aerial Photographs*, Plenum Press, New York.
Najman L., Schmitt M. [1996]. *Geodesic Saliency of Watershed Contours and Hierarchical Segmentation*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 18, no. 12, pp. 1163–1173.

Narendra P. M., Fitch R. C. [1981]. *Real-Time Adaptive Contrast Enhancement.* IEEE Trans. Pattern Analysis Machine Intelligence, vol. PAMI-3, no. 6, pp. 655–661.

Netravali A. N. [1977]. *On Quantizers for DPCM Coding of Picture Signals.* IEEE Trans. Info. Theory, vol. IT-23, no. 3, pp. 360–370.

Netravali A. N., Limb, J.O. [1980]. *Picture Coding: A Review.* Proc. IEEE, vol. 68, no. 3, pp. 366–406. [Имеется перевод: Нетравали А. Н., Лимб Дж. О. *Кодирование изображений: Обзор.* — ТИИЭР, 1980, т. 68, № 3, с. 76–124.]

Nevatia R. [1982]. *Machine Perception*, Prentice Hall, Upper Saddle River, N.J.

Ngan K. N., Meier T., Chai D. [1999]. *Advanced Video Coding: Principles and Techniques.* Elsevier, Boston.

Nie Y., Barner, K. E. [2006]. *The Fuzzy Transformation and Its Applications in Image Processing.* IEEE Trans. Image Proc., vol. 15, no. 4, pp. 910–927.

Nilsson, N.J. [1965]. *Learning Machines: Foundations of Trainable Pattern-Classifying Systems*, McGraw-Hill, New York. [Имеется перевод: Нильсон Н. *Обучающие машины.* — М.: Мир, 1967.]

Nilsson N.J. [1971]. *Problem Solving Methods in Artificial Intelligence*, McGraw-Hill, New York. [Имеется перевод: Нильсон Н. *Искусственный интеллект. Методы поиска решений.* — М.: Мир, 1973.]

Nilsson N.J. [1980]. *Principles of Artificial Intelligence*, Tioga, Palo Alto, Calif.

Nixon M., Aguado, A. [2002]. *Feature Extraction and Image Processing*, Newnes, Boston, MA.

Noble B., Daniel J. W. [1988]. *Applied Linear Algebra*, 3rd ed., Prentice Hall, Upper Saddle River, N.J.

O'Connor Y. Z., Fessler J. A. [2006]. *Fourier-Based Forward and Back-Projections in Iterative Fan-Beam Tomographic Image Reconstruction*. IEEE Trans. Med. Imag., vol. 25, no. 5, pp. 582–589.

Odegard J. E., Gopinath R. A., Burrus C. S. [1992]. *Optimal Wavelets for Signal Decomposition and the Existence of Scale-Limited Signals.* Proceedings of IEEE Int. Conf. On Signal Processing, ICASSP-92, San Francisco, CA, vol. IV, 597–600.

Olkkonen J., Olkkonen H. [2007]. *Discrete Lattice Wavelet Transform*, IEEE Trans. Circuits and Systems II: Express Briefs, vol. 54, no. 1, pp. 71–75.

Olson C. F. [1999]. *Constrained Hough Transforms for Curve Detection*. Computer Vision and Image Understanding, vol. 73, no. 3, pp. 329–345.

O'Neil J. B. [1971]. *Entropy Coding in Speech and Television Differential PCM Systems.* IEEE Trans. Info. Theory, vol. IT-17, pp. 758–761.

Oommen R. J., Loke R. K. S. [1997]. *Pattern Recognition of Strings with Substitutions, Insertions, Deletions, and Generalized Transpositions.* Pattern Recognition, vol. 30, no. 5, pp. 789–800.

Oppenheim A. V., Schafer R. W. [1975]. *Digital Signal Processing*, Prentice Hall, Englewood Cliffs, N.J. [Имеется перевод: Оппенгейм А. В., Шафер Р. В. *Цифровая обработка сигналов.* — М.: Связь, 1979.]

Oppenheim A. V., Schafer R. W., Stockham T. G., Jr. [1968]. *Nonlinear Filtering of Multiplied and Convolved Signals.* Proc. IEEE, vol. 56, no. 8, pp. 1264–1291. [Имеется перевод: Оппенгейм А., Шафер Р., Стокхэм Т. *Нелинейная фильтрация перемноженных и свернутых сигналов.* — ТИИЭР, 1968, т, 56, № 8, с. 5–33]

Oster G., Nishijima Y. [1963]. *Moiré Patterns.* Scientific American, vol. 208, no. 5, pp. 54–63.

Otsu N. [1979]. *A Threshold Selection Method from Gray-Level Histograms*. IEEE Trans. Systems, Man, and Cybernetics, vol. 9, no. 1, pp. 62–66.

Oyster C.W. [1999]. *The Human Eye: Structure and Function*, Sinauer Associates, Sunderland, Mass.

Paez M.D., Glisson T.H. [1972]. *Minimum Mean-Square-Error Quantization in Speech PCM and DPCM Systems*. IEEE Trans. Communications, vol. COM-20, pp. 225–230.

Pao Y.H. [1989]. *Adaptive Pattern Recognition and Neural Networks*, Addison-Wesley, Reading, Mass.

Papamarkos N., Atsalakis A. [2000]. *Gray-Level Reduction Using Local Spatial Features*. Computer Vision and Image Understanding, vol. 78, no. 3, pp. 336–350.

Papoulis A. [1991]. *Probability, Random Variables, and Stochastic Processes*, 3rd ed., McGraw-Hill, New York.

Parhi K., Nishitani T. [1999]. *Digital Signal Processing in Multimedia Systems*. Chapter 18: A Review of Watermarking Principles and Practices, M. Miller, et al., pp. 461–485, Marcel Dekker Inc., New York.

Park H., Chin R.T. [1995]. *Decomposition of Arbitrarily-Shaped Morphological Structuring*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 17, no. 1, pp. 2–15.

Parker J.R. [1991]. *Gray Level Thresholding in Badly Illuminated Images*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 13, no. 8, pp. 813–819.

Parker J.R. [1997]. *Algorithms for Image Processing and Computer Vision*. John Wiley & Sons, New York.

Patrascu V. [2004]. *Fuzzy Enhancement Method Using Logarithmic Model*. IEEEFuzz' 04, vol. 3, pp. 1431–1436.

Pattern Recognition [2000]. *Special issue on Mathematical Morphology and Nonlinear Image Processing*, vol. 33, no. 6, pp. 875–1117.

Pavlidis T. [1977]. *Structural Pattern Recognition*, Springer-Verlag, New York.

Pavlidis T. [1982]. *Algorithms for Graphics and Image Processing*, Computer Science Press, Rockville, Md. [Имеется перевод: Павлидис Т. *Алгоритмы машинной графики и обработки изображений*. — М.: Радио и связь, 1986.]

Pavlidis T., Liow Y.T. [1990]. *Integrating Region Growing and Edge Detection*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 12, no. 3, pp. 225–233.

Peebles P.Z. [1993]. *Probability, Random Variables, and Random Signal Principles*, 3rd ed., McGraw-Hill, New York.

Pennebaker W.B., Mitchell J. L. [1992]. *JPEG: Still Image Data Compression Standard*. Van Nostrand Reinhold, New York.

Pennebaker W.B., Mitchell J. L., Langdon G.G., Jr., Arps R.B. [1988]. *An Overview of the Basic Principles of the Q-coder Adaptive Binary Arithmetic Coder*. IBM J. Res. Dev., vol. 32, no. 6, pp. 717–726.

Perez A., Gonzalez R.C. [1987]. *An Iterative Thresholding Algorithm for Image Segmentation*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. PAMI-9, no. 6, pp. 742–751.

Perona P., Malik J. [1990]. *Scale-Space and Edge Detection Using Anisotropic Diffusion*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 12, no. 7, pp. 629–639.

Persoon E., Fu K.S. [1977]. *Shape Discrimination Using Fourier Descriptors*. IEEE Trans. Systems Man Cybernetics, vol. SMC-7, no. 2, pp. 170–179.

Petrou M., Bosdogianni P. [1999]. *Image Processing: The Fundamentals*, John Wiley & Sons, UK.

Petrou M., Kittler J. [1991]. *Optimal Edge Detector for Ramp Edges*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 13, no. 5, pp. 483–491.

Piech M.A. [1990]. *Decomposing the Laplacian*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 12, no. 8, pp. 830–831.

Pitas I., Vanetsanopoulos A.N. [1990]. *Nonlinear Digital Filters: Principles and Applications*, Kluwer Academic Publishers, Boston, Mass.

Plataniotis K.N, and Venetsanopoulos A.N. [2000]. *Color Image Processing and Applications*, Springer-Verlag, New York.

Pokorny C.K., Gerald C.F. [1989]. *Computer Graphics: The Principles Behind the Art and Science*, Franklin, Beedle & Associates, Irvine, Calif.

Porco C.C., West R.A., et al. [2004]. *Cassini Imaging Science: Instrument Characteristics and Anticipated Scientific Investigations at Saturn*. Space Science Reviews, vol. 115, pp. 363–497.

Poynton C.A. [1996]. *A Technical Introduction to Digital Video*, John Wiley & Sons, New York.

Prasad L., Iyengar S.S. [1997]. *Wavelet Analysis with Applications to Image Processing*, CRC Press, Boca Raton, Fla.

Pratt W.K. [1991]. *Digital Image Processing*, 2nd ed., John Wiley & Sons, New York. Pratt W.K. [2001]. *Digital Image Processing*, 3rd ed., John Wiley & Sons, New York. [Имеется перевод 1-го издания: Прэтт У. *Цифровая обработка изображений*. — М.: Мир, 1982. Кн. 1 и 2]

Preparata F.P., Shamos M.I. [1985]. *Computational Geometry: An Introduction*, Springer-Verlag, New York.

Preston K. [1983]. *Cellular Logic Computers for Pattern Recognition*. Computer, vol. 16, no. 1, pp. 36–47.

Prewitt J.M.S. [1970]. *Object Enhancement and Extraction*. in Picture Processing and Psychopictorics, Lipkin B.S., Rosenfeld A. (eds.), Academic Press, New York.

Prince J.L., Links J.M. [2006]. *Medical Imaging Signals and Systems, Prentice Hall*. Upper Saddle River, NJ.

Principe J.C., Euliano N.R., Lefebre W.C. [1999]. *Neural and Adaptive Systems: Fundamentals through Simulations*, John Wiley & Sons, New York.

Pritchard D.H. [1977]. *U. S. Color Television Fundamentals — A Review*. IEEE Trans. Consumer Electronics, vol. CE-23, no. 4, pp. 467–478.

Proc. IEEE [1967]. *Special issue on redundancy reduction*, vol. 55, no. 3. [Имеется перевод: *Сокращение избыточности. Тематический выпуск*. — ТИИЭР, 1967, т. 55, № 3.]

Proc. IEEE [1972]. *Special issue on digital picture processing*, vol. 60, no. 7. [Имеются переводы: ТИИЭР, 1972, т. 60, № 7; а также: *Обработка изображений при помощи цифровых вычислительных машин*. — М.: Мир, 1973.]

Proc. IEEE [1980]. *Special issue on the encoding of graphics*, vol. 68, no. 7. [Имеется перевод: *Цифровое кодирование. Тематический выпуск*. — ТИИЭР, 1980, т. 68, № 7.]

Proc. IEEE [1985]. *Special issue on visual communication systems*, vol. 73, no. 2. [Имеется перевод: *Системы видеосвязи. Тематический выпуск*. — ТИИЭР, 1985, т. 73, № 4 и № 5.]

Qian R.J., Huang T.S. [1996]. *Optimal Edge Detection in Two-Dimensional Images*. IEEE Trans. Image Processing, vol. 5, no. 7, pp. 1215–1220.

Rabbani M., Jones P.W. [1991]. *Digital Image Compression Techniques*, SPIE Press, Bellingham, Wash.

Rajala S.A., Riddle A.N., Snyder W.E. [1983]. *Application of One-Dimensional Fourier Transform for Tracking Moving Objects in Noisy environments*. Computer Vision Image Processing, vol. 21, pp. 280–293.

Ramachandran G.N., Lakshminarayanan A.V. [1971]. *Three Dimensional Reconstructions from Radiographs and Electron Micrographs: Application of Convolution Instead of Fourier Transforms*. Proc. Nat. Acad. Sci., vol. 68, pp. 2236–2240.

Rane S., Sapiro G. [2001]. *Evaluation of JPEG-LS, the New Lossless and Controlled-Lossy Still Image Compression Standard, for Compression of High-Resolution Elevation Data*. IEEE Trans. Geoscience and Remote Sensing, vol. 39, no. 10, pp. 2298–2306.

Rangayyan R.M. [2005]. *Biomedical Image Analysis*. CRC Press, Boca Raton, FL.

Reddy B.S., Chatterji B.N. [1996]. *An FFT-Based Technique for Translation, Rotation, and Scale Invariant Image Registration*. IEEE Trans. Image Processing, vol. 5, no. 8, pp. 1266–1271.

Regan D.D. [2000]. *Human Perception of Objects: Early Visual Processing of Spatial Form Defined by Luminance, Color, Texture, Motion, and Binocular Disparity*, Sinauer Associates, Sunderland, Mass.

Rice R.F. [1979]. *Some Practical Universal Noiseless Coding Techniques*. Tech. Rep. JPL-79-22, Jet Propulsion Lab., Pasadena, CA.

Ritter G.X., Wilson J.N. [2001]. *Handbook of Computer Vision Algorithms in Image Algebra*, CRC Press, Boca Raton, Fla.

Roberts L.G. [1965]. *Machine Perception of Three-Dimensional Solids*. In Optical and Electro-Optical Information Processing, Tippet J.T. (ed.), MIT Press, Cambridge, Mass. [Имеется перевод: Робертс Л. *Автоматическое восприятие трехмерных объектов.* // Интегральные роботы. — М.: Мир, 1973, с. 162–208.]

Robertson A.R. [1977]. *The CIE 1976 Color Difference Formulae*. Color Res. Appl., vol. 2, pp. 7–11.

Robinson G.S. [1976]. *Detection and Coding of Edges Using Directional Masks*. University of Southern California, Image Processing Institute, Report no. 660.

Robinson J.A. [1965]. *A Machine-Oriented Logic Based on the Resolution Principle*. J. ACM, vol. 12, no. 1, pp. 23–41.

Robinson J. [2006]. *Adaptive Prediction Trees for Image Compression*. IEEE Trans. Image Proc., vol. 15, no. 8, pp. 2131–2145.

Rock I. [1984]. *Perception*, W.H. Freeman, New York, New York.

Roese J.A., Pratt W.K., Robinson G.S. [1977]. *Interframe Cosine Transform Image Coding*. IEEE Trans. Communications, vol. COM-25, pp. 1329–1339.

Rosenblatt F. [1959]. *Two Theorems of Statistical Separability in the Perceptron*. In Mechanisation of Thought Processes: Proc. of Symposium No. 10, held at the National Physical Laboratory, November 1958, H.M. Stationery Office, London, vol. 1, pp. 421–456.

Rosenblatt F. [1962]. *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*, Spartan, Washington, D.C. [Имеется перевод: Розенблатт Ф. *Принципы нейродинамики. Персептрон и теория механизмов мозга.* — М.: Мир, 1966.]

Rosenfeld A. (ed.) [1984]. *Multiresolution Image Processing and Analysis*, Springer-Verlag, New York.

Rosenfeld A. [1999]. *Image Analysis and Computer Vision: 1998*. Computer Vision and Image Understanding, vol. 74, no. 1, pp. 36–95.

Rosenfeld A. [2000]. *Image Analysis and Computer Vision: 1999*. Computer Vision and Image Understanding, vol. 78, no. 2, pp. 222–302.

Rosenfeld A., Kak A. C. [1982]. *Digital Picture Processing*, vols. 1 and 2, 2nd ed., Academic Press, New York.

Rosin P. L. [1997]. *Techniques for Assessing Polygonal Approximations of Curves*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 19, no. 6, pp. 659–666.

Rudnick P. [1966]. *Note on the Calculation of Fourier Series*. Math. Comput., vol. 20, pp. 429–430.

Rumelhart D. E., Hinton G. E., Williams R. J. [1986]. *Learning Internal Representations by Error Propagation*. In Parallel Distributed Processing: Explorations in the Microstructures of Cognition, vol. 1: Foundations, Rumelhart D. E., et al. (eds.), MIT Press, Cambridge, Mass., pp. 318–362.

Rumelhart D. E., McClelland J. L. (eds.) [1986]. *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, *vol. 1: Foundations*, MIT Press, Cambridge, Mass.

Runge C. [1903]. *Zeit. für Math., and Physik*, vol. 48, p. 433.

Runge C. [1905]. *Zeit. für Math., and Physik*, vol. 53, p. 117.

Runge C., König H. [1924]. *Die Grundlehren der Mathematischen Wissenschaften*. Vorlesungen iiber Numerisches Rechnen, vol. 11, Julius Springer, Berlin.

Russ J. C. [1999]. *The Image Processing Handbook*, 3rd ed., CRC Press, Boca Raton, Fla.

Russo F., Ramponi G. [1994]. *Edge Extraction by FIRE Operators*. Fuzz-IEEE '94, vol. 1, pp. 249–243, IEEE Press, New York.

Sahni S., Jenq J.-F. [1992]. *Serial and Parallel Algorithms for the Medial Axis Transform*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 14, no. 12, pp. 1218–1224.

Sahoo S. S. P. K., Wong A. K. C., Chen Y. C. [1988]. *Survey of Thresholding Techniques*. Computer Vision, Graphics and Image Processing, vol. 41, pp. 233–260.

Saito N., Cunningham, M. A. [1990]. *Generalized E-Filter and its Application to Edge Detection*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 12, no. 8, pp. 814–817.

Sakrison D. J., Algazi V. R. [1971]. *Comparison of Line-by-Line and Two-Dimensional Encoding of Random Images*. IEEE Trans. Information Theory, vol. IT-17, no. 4, pp. 386–398.

Salari E., Siy P. [1984]. *The Ridge-Seeking Method for Obtaining the Skeleton of Digital Images*. IEEE Trans. System Man Cybernetics, vol. SMC-14, no. 3, pp. 524–528.

Salinas R. A., Abidi M. A., Gonzalez R. C. [1996]. *Data Fusion: Color Edge Detection and Surface Reconstruction Through Regularization*. IEEE Trans. Industrial Electronics, vol. 43, no. 3, pp. 355–363, 1996.

Sato Y. [1992]. *Piecewise Linear Approximation of Plane Curves by Perimeter Optimization*. Pattern Recognition, vol. 25, no. 12, pp. 1535–1543.

Sauvola J., Pietikainen M. [2000]. *Adaptive Document Image Binarization*. Pattern Recognition, vol. 33, no. 2, pp. 225–236.

Schalkoff R. J. [1989]. *Digital Image Processing and Computer Vision*, John Wiley & Sons, New York.

Schonfeld D., Goutsias J. [1991]. *Optimal Morphological Pattern Restoration from Noisy Binary Images*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 13, no. 1, pp. 14–29.

Schowengerdt R.A. [1983]. *Techniques for Image Processing and Classification in Remote Sensing*, Academic Press, New York.

Schreiber W.F. [1956]. *The Measurement of Third Order Probability Distributions of Television Signals*. IRE Trans. Info. Theory, vol. IT-2, pp. 94–105.

Schreiber W.F. [1967]. *Picture Coding*. Proc. IEEE (Special issue on Redundancy Reduction), vol. 55, pp. 320–330. [Имеется перевод: Шрейбер В. *Кодирование изображений*. — ТИИЭР, 1967, т. 55, № 3, с. 84–96.]

Schreiber W.F., Knapp C.F. [1958]. *TV Bandwidth Reduction by Digital Coding*. Proc. IRE National Convention, pt. 4, pp. 88–99.

Schwartz J.W., Barker R.C. [1966]. *Bit-Plane Encoding: A Technique for Source Encoding*. IEEE Trans. Aerosp. Elec. Systems, vol. AES-2, no. 4, pp. 385–392.

Selesnick I., Baraniuk R., Kingsbury N. [2005]. *The Dual-Tree Complex Wavelet Transform*. IEEE Signal Processing Magazine, vol. 22, no. 6, pp. 123–151.

Serra J. [1982]. *Image Analysis and Mathematical Morphology*, Academic Press, New York.

Serra J. (ed.) [1988]. *Image Analysis and Mathematical Morphology*, vol. 2, Academic Press, New York.

Sezan M.I., Rabbani M., Jones P.W. [1989]. *Progressive Transmission of Images Using a Prediction/Residual Encoding Approach*. Opt. Eng., vol. 28, no. 5, pp. 556–564.

Shack R.V. [1964]. *The Influence of Image Motion and Shutter Operation on the Photographic Transfer Function*. Appl. Opt., vol. 3, pp. 1171–1181.

Shafarenko L., Petrou M., Kittler J. [1998]. *Histogram-Based Segmentation in a Perceptually Uniform Color Space*. IEEE Trans. Image Processing, vol. 7, no. 9, pp. 1354–1358.

Shaked D., Bruckstein A.M. [1998]. *Pruning Medial Axes*. Computer Vision and Image Understanding, vol. 69, no. 2, pp. 156–169.

Shannon C.E. [1948]. *A Mathematical Theory of Communication*. The Bell Sys. Tech. J., vol. XXVII, no. 3, pp. 379–423. [Имеется перевод: *Математическая теория связи*. // Шеннон. Работы по теории информации и кибернетике. М.: ИЛ, 1963, с. 243–332.]

Shapiro L.G., Stockman G.C. [2001]. *Computer Vision*, Prentice Hall, Upper Saddle River, N.J.

Shapiro V.A. [1996]. *On the Hough Transform of Multi-Level Pictures*. Pattern Recognition, vol. 29, no. 4, pp. 589–602.

Shariat H., Price K.E. [1990]. *Motion Estimation with More Than Two Frames*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 12, no. 5, pp. 417–434.

Shepp L.A., Logan B.F. [1974]. *The Fourier Reconstruction of a Head Section*. IEEE Trans. Nucl. Sci., vol. NS-21, pp. 21–43.

Sheppard J.J., Jr., Stratton R.H., Gazley C., Jr. [1969]. *Pseudocolor as a Means of Image Enhancement*. Am. J. Optom. Arch. Am. Acad. Optom., vol. 46, pp. 735–754.

Shih F.Y., Wong W.-T. [1994]. *Fully Parallel Thinning with Tolerance to Boundary Noise*. Pattern Recognition, vol. 27, no. 12, pp. 1677–1695.

Shih F.Y.C., Mitchell O.R. [1989]. *Threshold Decomposition of Gray-Scale Morphology into Binary Morphology*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 11, no. 1, pp. 31–42.

Shirley P. [2002]. *Fundamentals of Computer Graphics*. A.K. Peters, Natick, MA.
Sid-Ahmed M.A. [1995]. *Image Processing: Theory, Algorithms, and Architectures*, McGraw-Hill, New York.
Sikora T. [1997]. *MPEG Digital Video-Coding Standards*. IEEE Signal Processing, vol. 14, no. 5, pp. 82–99.
Simon J.C. [1986]. *Patterns and Operators: The Foundations of Data Representations*, McGraw-Hill, New York.
Sklansky J., Chazin R.L., Hansen B.J. [1972]. *Minimum-Perimeter Polygons of Digitized Silhouettes*. IEEE Trans. Computers, vol. C-21, no. 3, pp. 260–268.
Sloboda F., Zatko B., Stoer J. [1998]. *On Approximation of Planar One-Dimensional Continua*. in Advances in Digital and Computational Geometry, R. Klette, A. Rosenfeld, F. Sloboda (eds.), Springer, Singapore, pp. 113–160.
Smirnov A. [1999]. *Processing of Multidimensional Signals*, Springer-Verlag, New York.
Smith A.R. [1978]. *Color Gamut Transform Pairs*. Proc. SIGGRAPH '78, published as Computer Graphics, vol. 12, no. 3, pp. 12–19.
Smith J.O. III [2003]. *Mathematics of the Discrete Fourier Transform*, W3K Publishing, CCRMA, Stanford, CA. (Also available online at http://ccrma.stanford.edu/~jos/mdft.)
Smith M.J.T., Barnwell T.P. III [1984]. *A Procedure for Building Exact Reconstruction Filter Banks for Subband Coders*. Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing, San Diego, Calif.
Smith M.J.T., Barnwell T.P. III [1986]. *Exact Reconstruction Techniques for Tree-Structured Subband Coders*. IEEE Trans. On Acoustics, Speech, and Signal Processing, vol. 34, no. 3, pp. 434–441.
Sobel I.E. [1970]. *Camera Models and Machine Perception*. Ph.D. dissertation, Stanford University, Palo Alto, Calif.
Sonka M., Hlavac V., Boyle R. [1999]. *Image Processing, Analysis, and Machine Vision*, 2nd ed., PWS Publishing, New York.
Snyder W.E., Qi Hairong [2004]. *Machine Vision*. Cambridge University Press, New York.
Soille P. [2003]. *Morphological Image Analysis: Principles and Applications*. 2nd ed., Springer-Verlag, New York.
Solari S. [1997]. *Digital Video and Audio Compression*. McGraw-Hill, New York.
Stark H. (ed.) [1987]. *Image Recovery: Theory and Application*, Academic Press, New York. [Имеется перевод: Старк Г. *Реконструкция изображений* — М.: Мир, 1992.]
Stark J.A. [2000]. *Adaptive Image Contrast Enhancement Using Generalizations of Histogram Equalization*. IEEE Trans. Image Processing, vol. 9, no. 5, pp. 889–896.
Stockham T.G., Jr. [1972]. *Image Processing in the Context of a Visual Model*. Proc. IEEE, vol. 60, no. 7, pp. 828–842. [Имеется перевод: Стокхэм. *Обработка изображений в контексте моделей зрения*. — ТИИЭР, 1972, т. 60, № 7.]
Storer J.A., Reif J.H., eds. [1991]. *Proceedings of DDC '91*, IEEE Computer Society Press, Los Alamitos, Calif.
Strang G., Nguyen T. [1996]. *Wavelets and Filter Banks*, Wellesley-Cambridge Press Wellesley, Mass.
Stumpff K. [1939]. *Tafeln und Aufgaben zur Harmonischen Analyse und Periodogrammrechnung*, Julius Springer, Berlin.

Sussner P., Ritter G.X. [1997]. *Decomposition of Gray-Scale Morphological Templates Using the Rank Method*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 19, no. 6, pp. 649–658.

Swets D.L., Weng J. [1996]. *Using Discriminant Eigenfeatures for Image Retrieval*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 18, no. 8, pp. 1831–1836.

Symes P.D. [2001]. *Video Compression Demystified*. McGraw-Hill, New York.

Sze T.W., Yang Y.H. [1981]. *A Simple Contour Matching Algorithm*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. PAMI-3, no. 6, pp. 676–678.

Tanaka E. [1995]. *Theoretical Aspects of Syntactic Pattern Recognition*. Pattern Recognition, vol. 28, no. 7 pp. 1053–1061.

Tanimoto S.L. [1979]. *Image Transmission with Gross Information First*. Computer Graphics Image Processing, vol. 9, pp. 72–76.

Tasto M., Wintz P.A. [1971]. *Image Coding by Adaptive Block Quantization*. IEEE Trans. Communications *Tech*., vol. COM-19, pp. 957–972.

Tasto M., Wintz P.A. [1972]. *A Bound on the Rate-Distortion Function and Application to Images*. IEEE Trans. Info. Theory, vol. IT-18, pp. 150–159.

Teh C.H., Chin R.T. [1989]. *On the Detection of Dominant Points on Digital Curves*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 11, no. 8, pp. 859–872.

Theoridis S., Konstantinos K. [2006]. *Pattern Recognition*. 3rd ed., Academic Press, New York.

Thévenaz P., Unser M. [2000]. *Optimization of Mutual Information for Multiresolution Image Registration*. IEEE Trans. Image Processing, vol. 9, no. 12, pp. 2083–2099.

Thomas L.H. [1963]. *Using a Computer to Solve Problems in Physics*. Application of Digital Computers, Ginn, Boston.

Thomason M.G., Gonzalez R.C. [1975]. *Syntactic Recognition of Imperfectly Specified Patterns*. IEEE Trans. Computers, vol. C-24, no. 1, pp. 93–96.

Thompson W.B. (ed.) [1989]. *Special Issue on Visual Motion*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 11, no. 5, pp. 449–541.

Thompson W.B., Barnard, S.T. [1981]. *Lower-Level Estimation and Interpretation of Visual Motion*. Computer, vol. 14, no. 8, pp. 20–28.

Thorell L.G., Smith W.J. [1990]. *Using Computer Color Effectively*, Prentice Hall, Upper Saddle River, N.J.

Tian J., Wells R.O., Jr. [1995]. *Vanishing Moments and Wavelet Approximation*, Technical Report CML TR-9501, Computational Mathematics Lab., Rice University, Houston, Texas.

Tizhoosh H.R. [2000]. *Fuzzy Image Enhancement: An Overview*. in Fuzzy Techniques in Image Processing, E. Kerre and M. Nachtegael, eds., Springer-Verlag, New York.

Tomita F., Shirai Y., Tsuji S. [1982]. *Description of Texture by a Structural Analysis*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. PAMI-4, no. 2, pp. 183–191.

Topiwala P.N. (ed.) [1998]. *Wavelet Image and Video Compression*, Kluwer Academic Publishers, Boston, Mass.

Toro J., Funt B. [2007]. *A Multilinear Constraint on Dichromatic Planes for Illumination Estimation*. IEEE Trans. Image Proc., vol. 16, no. 1, pp. 92–97.

Tou J.T., Gonzalez R.C. [1974]. *Pattern Recognition Principles*, Addison-Wesley, Reading, Mass. [Имеется перевод: Ту Дж., Гонсалес Р. *Принципы распознавания образов*. — М.: Мир, 1978.]

Tourlakis G.J. [2003]. *Lectures in Logic and Set Theory*. Cambridge University Press, Cambridge, UK.

Toussaint G.T. [1982]. *Computational Geometric Problems in Pattern Recognition*. In Pattern Recognition Theory and Applications, Kittler, J., Fu, K. S., Pau, L.F. (eds.), Reidel, New York, pp. 73–91.

Tsai J.-C., Hsieh C.-H, and Hsu T.-C. [2000]. *A New Dynamic Finite-State Vector Quantization Algorithm for Image Compression*. IEEE Trans. Image Processing, vol. 9, no. 11, pp. 1825–1836.

Tsujii O., Freedman M.T., Mun K.S. [1998]. *Anatomic Region-Based Dynamic Range Compression for Chest Radiographs Using Warping Transformation of Correlated Distribution*. IEEE Trans. Medical Imaging, vol. 17, no. 3, pp. 407–418.

Udpikar V.R., Raina J.P. [1987]. *BTC Image Coding Using Vector Quantization*. IEEE Trans. Communications, vol. COM-35, no. 3, pp. 352–356.

Ueda N. [2000]. *Optimal Linear Combination of Neural Networks for Improving Classification Performance*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 22, no. 2, pp. 207–215.

Ullman S. [1981]. *Analysis of Visual Motion by Biological and Computer Systems*. IEEE Computer, vol. 14, no. 8, pp. 57–69. [Имеется перевод: Ульман Ш. *Принципы восприятия подвижных объектов*. — М.: Радио и связь, 1983.]

Umbaugh S.E. [2005]. *Computer Imaging: Digital Image Analysis and Processing*. CRC Press, Boca Raton, FL.

Umeyama S. [1988]. *An Eigendecomposition Approach to Weighted Graph Matching Problems*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 10, no. 5, pp. 695–703.

Unser M. [1995]. *Texture Classification and Segmentation Using Wavelet Frames*. IEEE Trans. on Image Processing, vol. 4, no. 11, pp. 1549–1560.

Unser M., Aldroubi A., Eden M. [1993]. *A Family of Polynomial Spline Wavelet Transforms*. Signal Processing, vol. 30, no. 2, pp. 141–162.

Unser M., Aldroubi A., Eden M. [1993]. *B-Spline Signal Processing, Parts I and II*. IEEE Trans. Signal Processing, vol. 41, no. 2, pp. 821–848.

Unser M., Aldroubi A., Eden, M. [1995]. *Enlargement or Reduction of Digital Images with Minimum Loss of Information*. IEEE Trans. Image Processing, vol. 4, no. 5, pp. 247–257.

Vaidyanathan P.P., Hoang P.-Q. [1988]. *Lattice Structures for Optimal Design and Robust Implementation of Two-Channel Perfect Reconstruction Filter Banks*. IEEE Trans. Acoustics, Speech, and Signal Processing, vol. 36, no. 1, pp. 81–94.

Vailaya A., Jain A., Zhang H.J. [1998]. *On Image Classification: City Images vs. Landscapes*. Pattern Recognition, vol. 31, no. 12, pp. 1921–1935.

Vetterli M. [1986]. *Filter Banks Allowing Perfect Reconstruction Signal Processing*, vol. 10, no. 3, pp. 219–244.

Vetterli M., Kovacevic J. [1995]. *Wavelets and Subband Coding*, Prentice Hall, Englewood Cliffs, N.J.

Vincent L. [1993]. *Morphological Grayscale Reconstruction in Image Analysis:Applications and Efficient Algorithms*. IEEE Trans. Image Proc., vol. 2. no. 2, pp. 176–201.

Voss K., Suesse H. [1997]. *Invariant Fitting of Planar Objects by Primitives*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 19, no. 1, pp. 80–84.

Vuylsteke P., Kittler J. [1990]. *Edge-Labeling Using Dictionary-Based Relaxation*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 12, no. 2, pp. 165–181.

Walsh J.W.T. [1958]. *Photometry*, Dover, New York.

Wang D., Zhang L., Vincent A., Speranza F. [2006]. *Curved Wavelet Transform for Image Coding*. IEEE Trans. Image Proc., vol. 15, no. 8, pp. 2413–2421.

Wang G., Zhang J., Pan G.-W. [1995]. *Solution of Inverse Problems in Image Processing by Wavelet Expansion*. IEEE Trans. Image Processing, vol. 4, no. 5, pp. 579–593.

Wang Y.-P., Lee S. L., Toraichi K. [1999]. *Multiscale Curvature-Based Shape Representation Using* β*-Spline Wavelets*. IEEE Trans. Image Processing, vol. 8, no. 11, pp. 1586–1592.

Wang Z., Rao K. R., Ben-Arie J. [1996]. *Optimal Ramp Edge Detection Using Expansion Matching*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 18, no. 11, pp. 1092–1097.

Watt A. [1993]. *3D Computer Graphics*, 2nd ed., Addison-Wesley, Reading, Mass.

Wechsler [1980]. *Texture Analysis—A Survey*. Signal Process, vol. 2, pp. 271–280.

Wei D., Tian J., Wells R. O., Jr., Burrus C. S. [1998]. *A New Class of Biorthogonal Wavelet Systems for Image Transform Coding*. IEEE Trans. Image Processing, vol. 7, no. 7, pp. 1000–1013.

Weinberger M. J., Seroussi G., Sapiro G. [2000]. *The LOCO-I Lossless Image Compression Algorithm: Principles and Standardization into JPEG-LS*. IEEE Trans. Image Processing, vol. 9, no. 8, pp. 1309–1324.

Westenberg M. A., Roerdink J. B. T. M. [2000]. *Frequency Domain Volume Rendering by the Wavelet X-ray Transform*. IEEE Trans. Image Processing, vol. 9, no. 7, pp. 1249–1261.

Weszka J. S. [1978]. *A Survey of Threshold Selection Techniques*. Computer Graphics Image Processing, vol. 7, pp. 259–265.

White J. M., Rohrer G. D. [1983]. *Image Thresholding for Optical Character Recognition and Other Applications Requiring Character Image Extraction*. IBM J. Res. Devel., vol. 27, no. 4, pp. 400–411.

Widrow B. [1962]. *Generalization and Information Storage in Networks of 'Adaline' Neurons*. In Self-Organizing Systems 1962, Yovitz, M. C., et al. (eds.), Spartan, Washington, D. C., pp. 435–461.

Widrow B., Hoff M. E. [1960]. *Adaptive Switching Circuits*. 1960 IRE WESCON Convention Record, Part 4, pp. 96–104.

Widrow B., Stearns S. D. [1985]. *Adaptive Signal Processing*, Prentice Hall, Englewood Cliffs, N.J.

Wiener N. [1942]. *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*, the MIT Press, Cambridge, Mass.

Wilburn J. B. [1998]. *Developments in Generalized Ranked-Order Filters*. Journal Optical Society of America-A. Optics, Image Science, and Vision, vol. 15, no. 5, pp. 1084–1099.

Windyga P. S. [2001]. *Fast Impulsive Noise Removal*. IEEE Trans. Image Processing, vol. 10, no. 1, pp. 173–179.

Wintz P. A. [1972]. *Transform Picture Coding*. Proc. IEEE, vol. 60, no. 7, pp. 809–820. [Имеется перевод: Уинц. *Кодирование изображений посредством преобразований*. — ТИИЭР, 1972, т. 60, № 7. с. 69–84.]

Witten I. H., Neal R. M., Cleary J. G. [1987]. *Arithmetic Coding for Data Compression*. Comm. ACM, vol. 30, no. 6, pp. 520–540.

Wolberg G. [1990]. *Digital Image Warping*. IEEE Computer Society Press, Los Alamitos, CA.

Wolff R. S., Yaeger L. [1993]. *Visualization of Natural Phenomena*, Springer-Verlag, New York.

Won C. S., Gray R. M. [2004]. *Stochastic Image Processing.* Kluwer Academic/Plenum Publishers, New York.

Woods J. W., O'Neil S. D. [1986]. *Subband Coding of Images.* IEEE Trans. Acoustic Speech Signal Processing, vol. ASSP-35, no. 5, pp. 1278–1288.

Woods R. E., Gonzalez R. C. [1981]. *Real-Time Digital Image Enhancement.* Proc. IEEE, vol. 69, no. 5, pp. 643–654. [Имеется перевод: Вудс Р. Э., Гонсалес Р. С. *Цифровые методы улучшения изображения в реальном времени.* — ТИИЭР, 1981, т. 69, № 5.]

Xu Y., Weaver J. B., Healy D. M., Jr., Lu J. [1994]. *Wavelet Transform Domain Filters: A Spatially Selective Noise Filtration Technique.* IEEE Trans. Image Processing, vol. 3, no. 6, pp. 747–758.

Xu R., Pattanaik S., Hughes C. [2005]. *High-Dynamic-Range Still-Image Encoding in JPEG 2000.* IEEE Computer Graphics and Applications, vol. 25, no. 6, pp. 57–64.

Yachida M. [1983]. *Determining Velocity Maps by Spatio-Temporal Neighborhoods from Image Sequences.* Computer Vision Graph. Image Processing, vol. 21, no. 2, pp. 262–279.

Yamazaki Y., Wakahara Y., Teramura H. [1976]. *Digital Facsimile Equipment 'Quick-FAX' Using a New Redundancy Reduction Technique.* NTC '76, pp. 6.2-1–6.2-5.

Yan Y., Cosman P. [2003]. *Fast and Memory Efficient Text Image Compression with JBIG2.* IEER Trans. Image Proc., vol. 12, no. 8, pp. 944–956.

Yang X., Ramchandran K. [2000]. *Scalable Wavelet Video Coding Using Aliasing-Reduced Hierarchical Motion Compensation.* IEEE Trans. Image Processing, vol. 9, no. 5, pp. 778–791.

Yates F. [1937]. *The Design and Analysis of Factorial Experiments.* Commonwealth Agricultural Bureaux, Farnam Royal, Burks, England.

Yin P. Y., Yin L. H., Chen L. H. [1997]. *A Fast Iterative Scheme for Multilevel Thresholding Methods.* Signal Processing, vol. 60, pp. 305–313.

Yitzhaky Y., Lantzman A., Kopeika N. S. [1998]. *Direct Method for Restoration of Motion Blurred Images.* Journal Optical Society of America-A. Optics, Image Science, and Vision, vol. 15, no. 6, pp. 1512–1519.

You J., Bhattacharya P. [2000]. *A Wavelet-Based Coarse-to-Fine Image Matching Scheme in a Parallel Virtual Machine Environment.* IEEE Trans. Image Processing, vol. 9, no. 9, pp. 1547–1559.

Yu D., Yan H. [2001]. *Reconstruction of Broken Handwritten Digits Based on Structural Morphology.* Pattern Recognition, vol. 34, no. 2, pp. 235–254.

Yu S. S., Tsai W. H. [1990]. *A New Thinning Algorithm for Gray-Scale Images.* Pattern Recognition, vol. 23, no. 10, pp. 1067–1076.

Yuan M., Li J. [1987]. *A Production System for LSI Chip Anatomizing.* Pattern Recognition Letters, vol. 5, no. 3, pp. 227–232.

Zadeh L. A. [1965]. *Fuzzy Sets.* Inform. and Control, vol. 8, pp. 338–353.

Zadeh L. A. [1973]. *Outline of New Approach to the Analysis of Complex Systems and Decision Processes.* IEEE Trans. Systems, Man, Cyb., vol. SMC-3, no. 1, pp. 28–44.

Zadeh L. A. [1976]. *A Fuzzy-Algorithmic Approach to the Definition of Complex or Imprecise Concepts.* Int. J. Man-Machine Studies, vol. 8, pp. 249–291.

Zahara E., Shu-Kai S., Du-Ming T. [2005]. *Optimal Multi-Thresholding Using a Hybrid Optimization Approach*. Pattern Recognition Letters, vol. 26, no. 8, pp. 1082–1095

Zahn C. T., Roskies R. Z. [1972]. *Fourier Descriptors for Plane Closed Curves*. IEEE Trans. Computers, vol. C-21, no. 3, pp. 269–281.

Zhang T. Y., Suen C. Y. [1984]. *A Fast Parallel Algorithm for Thinning Digital Patterns*. Comm. ACM, vol. 27, no. 3, pp. 236–239.

Zhang Y., Rockett P. I. [2006]. *The Bayesian Operating Point of the Canny Edge Detector*. IEEE Trans. Image Proc., vol. 15, no. 11, pp. 3409–3416.

Zhu H., Chan F. H. Y., Lam F. K. [1999]. *Image Contrast Enhancement by Constrained Local Histogram Equalization*. Computer Vision and Image Understanding, vol. 73, no. 2, pp. 281–290.

Zhu P., Chirlian P. M. [1995]. *On Critical Point Detection of Digital Shapes*. IEEE Trans. Pattern Analysis Machine Intelligence, vol. 17, no. 8, pp. 737–748.

Zimmer Y., Tepper R., Akselrod S. [1997]. *An Improved Method to Compute the Convex Hull of a Shape in a Binary Image*. Pattern Recognition, vol. 30, no. 3, pp. 397–402.

Ziou D. [2001]. *The Influence of Edge Direction on the Estimation of Edge Contrast and Orientation*. Pattern Recognition, vol. 34, no. 4, pp. 855–863.

Ziv J., Lempel A. [1977]. *A Universal Algorithm for Sequential Data Compression*. IEEE Trans. Info. Theory, vol. IT-23, no. 3, pp. 337–343.

Ziv J., Lempel A. [1978]. *Compression of Individual Sequences Via Variable-Rate Coding*. IEEE Trans. Info. Theory, vol. IT-24, no. 5, pp. 530–536.

Zucker S. W. [1976]. *Region Growing: Childhood and Adolescence*. Computer Graphics Image Processing, vol. 5, pp. 382–399.

Zugaj D., Lattuati V. [1998]. *A New Approach of Color Images Segmentation Based on Fusing Region and Edge Segmentation Outputs*. Pattern Recognition, vol. 31, no. 2, pp. 105–113.

Предметный указатель

Автокорреляции матрица 696
Автокорреляция 415, 1000
Аддитивность 108, 403
Активирующая функция 1015, 1021—1024, 1029
Анализ изображения 23
Аналого-цифровой преобразователь (АЦП) 55, 93
Ангиография 32, 113
Арифметико-логическое устройство (АЛУ) 56
Арифметическое кодирование 637—640
Аффинное преобразование *см. Геометрическое преобразование аффинное*
База знаний 51, 54, 143, 224—225, 889
Базис в функциональном пространстве 554
– ортонормированный 556—557
– Рисса 556
Базовый прямоугольник 941—943, 984
Байеса дискриминантная функция 1005, 1007—1009
– классификатор 1004—1013
– решающее правило 860
– формула 861
Банк фильтров 545—548
– – анализа 546—547, 585—586
– – синтеза 546—547, 579—581, 585—586
Белый шум *см. Шум*
Биортогональность 547, 556
Бит 56, 91—94
Битовая плоскость, вырезание 158—160
– –, кодирование 655—659
Ближайший сосед 99—100, 125, 196, 278 *см. также Интерполяция*
Блок, привязка 686
Блочное трансформационное кодирование 659—679
– – –, выбор преобразования 660—666
– – –, выбор размера блока 666—668
– – –, двоичное представление 668—669
– – –, зональное кодирование 669—671
– – –, пороговое кодирование 671—674
– – –, сжатие JPEG 674—679
Буфер кадров 57
Быстрое вейвлет-преобразование (БВП) 574—583, 583—587, 593—604
– –, банк фильтров анализа 576—577, 580, 596
– –, – – синтеза 579—581, 585—586
– –, вейвлет-пакеты 593—604
– –, двумерное 583—587
– –, кратномасштабная обработка 574—583, 583—587
– – обратное 579—581, 584
– –, сжатие изображения 700—713
– –, частотно-временно́е покрытие плоскости 582—583
Быстрое преобразование Фурье (БПФ) *см. Дискретное преобразование Фурье*
Вебера отношение 71
Вейвлет 53, 536—610
– биортогональный Коэна – Добеши – Фово 602—604
–, двумерное преобразование 583—593
–, интегральное (непрерывное) вейвлет-преобразование 571—573
–, масштабная последовательность 564
– «мексиканская шляпа» 572—573
–, обнаружение контуров 590
–, одномерное преобразование 566—573
–, разложение в ряд 566—569
–, устранение шума 590—593
Вейвлет-кодирование 700—713
–, выбор вейвлета 701—703
–, выбор уровня разложения 703—706
–, расчет квантователя 704—705
–, сжатие JPEG-2000 706—713
Вейвлет-пакет 593—604
–, банк фильтров анализа 596
–, дерево подпространств анализа 593—596
–, разложение в виде двоичного дерева 593—599
–, функция стоимости 599—600
Вейвлет-преобразование 566—573, 583—593
– быстрое *см. Быстрое вейвлет-преобразование*
– дискретное (ДВП) 569—571

Вейвлет-функция 562—566
– двумерная разделимая 583—587
–, масштабные коэффициенты 564
– Хаара 564—566
–, частотно-временная плоскость 582—583
Вейвлеты и кратномасштабная обработка 536—610
Вектор признаков 991—998, 1005—1011, 1015—1036
Векторные операции *см. Операция*
Вероятностей оценка адаптивная контекстно-зависимая 639—640
Вероятностная модель 639—640
Весовая функция 400
Видимый диапазон электромагнитного спектра 29—30, 36—41
Винеровская фильтрация 414—418
Водораздел 889—899
Водораздел, алгоритм сегментации 894—896
–, использование маркеров 896—899
–, построение перегородок 892—894
–, сегментация 889—899
Водяной знак 713—721
– – в младших разрядах 715
– – видимый 714—715
– – закрытый (с ограниченным ключом) 717
– –, нанесение и выделение 715, 717—721
– – невидимый 715
– – – нестойкий 716
– – – стойкий 716—717
– – открытый (с неограниченным ключом) 717
– –, устойчивость к атаке 720—721
Воксел 880
Восстановление 52, 366—459 *см. также Реконструкция*
–, искажение трансляционно-инвариантное 403—406
–, минимизация сглаживающего функционала со связью 418—424
–, модель искажения 366—368, 403—406
–, – шума 368—378
–, оценка искажающей функции 406—411
–, подавление шума 378—393, 393—403
–, фильтр винеровский параметрический 424
–, – минимума среднеквадратического отклонения (винеровский) 414—418
–, – среднегеометрический 424
–, – эквализации спектра 424
–, фильтрация в пространственной области 378—393
–, – инверсная 411—414
–, – частотная 393—403
Выбор признаков *см. Дескриптор*, *Области дескриптор*
Выделение границы 235, 237, 745—746, 768, 799, 924
Выпуклая оболочка 751—753, 769, 936—937
– –, дескриптор 936—937
– –, определение 751
– –, построение 751—753
Выравнивание фона 113—114, 781, 785—787, 878—880
– области 798, 882—885
Вычитание (разность) изображений 112—113, 778—780, 884—885, 899—900
Гамма шум *см. Шум*
Гамма-код Элайеса 637
Гамма-коррекция 150
Гамут *см. Цветовой охват*
Гауссов фильтр 194, 208, 308—309, 314—317, 327—334, 336—337, 341—342, 346—347, 393, 396, 409, 540—541, 827—830, 832, 834, 837
– – пространственный *см. Фильтрация в пространственной области*
– – частотный *см. Фильтрация в частотной области*
– шум *см. Шум*
Гауссова пирамида 540
Геометрическое преобразование 122—129
– – аффинное 123
– –, изменение масштаба 124
– –, поворот 124
– –, сдвиг 124

––, скос 124
––тождественное 124
––, точки привязки 126—129
Гистограмма 160
–локальная 538
–цветного изображения 509—510
Гистограммы видоизменение 160—186, 509—510
––, нормировка 160—162
––, обратное преобразование (отображение) 163—164, 168
––, плотность распределения вероятностей 164—165
––, приведение (задание) 170—179
––, эквализация (линеаризация) 162—170
––, яркостное преобразование 162—179
Главные компоненты *см. Преобразование Хотеллинга*
Глаз человека *см. Зрительное восприятие*
Глобальное пороговое разделение *см. Пороговое преобразование*
Гомоморфная фильтрация 342—346
Градационное преобразование *см. Яркостное преобразование*
Градация яркости 25, 72, 92—99
Градиент 209—212, 620—624, 817—826
–, вектор нормали к контуру 818—819
–, контур 211—212, 521—524
–морфологический 778—779
–, направление 817
–, обнаружение контуров 817—826
–, операторы 210—211, 620—624, 819—824
–, повышение резкости 209—211
–, полутоновая морфология 778—779
–, производная первого порядка 209—211
–с пороговым разделением 824—826
–, свойства 817—819
–, цветовая сегментация 520—524
Градиентный оператор Превитта 820
––Робертса 210, 819
––Собела 211—212, 289—290, 318, 820—822
Граница, аппроксимация ломаной 925—932
–, ––, методы разбиения 932
–, ––, методы слияния 933
–, выделение 235—237, 745—746
–, дескриптор *см. Дескриптор*
–, диаметр 941
–, длина 941
–, кривизна 941
–, нахождение для сегментации 840—854
–, номер фигуры 942—943
–области 104
––внешняя 105
––внутренняя 105
–, описание 941—948
–, определение 104
–, пиксели 104—105
–, представление 919—940
–, прослеживание (алгоритм Мура) 920—922
–, разбиение на сегменты 936—937
–, связывание контура 840—854
–, сегмент 936—937
–, сигнатура 934—936
–, статистические характеристики 947—948
–, Фурье-дескрипторы 944—947
–, цепной код 922—925
–, эксцентриситет 941
Гранулометрия 781—783
Движение при сегментации 899—907
–––, накопленные разности 900—901
–––, опорное изображение 902—903
–––, пространственные методы 899—903
–––, частотные методы 903—907
–, компенсация при кодировании 684—693
–, оценка 686—690
Двойственность 555, 603, 737, 744, 754, 762, 764, 774
Декартово произведение 90, 225, 771
Декодер символов 625
Декодирование (восстановление) изображения 624—625
–, обратный преобразователь 625
Деконволюция 406 *см. также Реконструкция*
Дельта-модуляция (ДМ) 694—695

Дерево анализа, вейвлет-пакет 593—599
– двоичное 593
– подпространств анализа 593
Дескриптор, базовый прямоугольник 941
–, главные компоненты 969—979
–, диаметр 941
–, инварианты моментов 966—969
–, компактность 949
–, коэффициент округлости 949
–, коэффициенты Фурье 944—947
–, номер фигуры 942—943
–, периметр 949
–, площадь 949
–, реляционный 980—983
–, статистические характеристики 947—948
–, текстурный 954—966
–, топологический 950—953
–, число Эйлера 951—952
–, эксцентриситет 941
Детектор контуров *см. Обнаружение контуров*
Дефицит выпуклости 751, 936—937
Диаграмма цветностей 466—468
Диапазон динамический 91, 149—151, 155
– яркостей 70—71, 85, 112, 135, 155—158
Дилатация 734, 761, 768, 770, 772, 774, 784
Дискретизация 32, 85—92, 257—267 *см. также Квантование*
–, восстановление функции из отсчетов 267—268, 278—281
–, избыточная (передискретизация) 261, 264, 275—280
–, интервал 271—272
–, интерполяция 99—102, 278—281
–, координаты пространственные 85—92
–, муаровые узоры 281—282
–, наложение спектров 264—267, 276—282 *см. Наложение диапазонов частот*
–, основные понятия 85—87
– повторная (передискретизация) 99—102, 278—281, 922—925, 1003
–, повышение 539—541
–, представление изображения 87—92
–, преобразование Фурье 257—268, 274—276
–, прореживание 278—282
–, разрешение пространственное 92—99
–, – яркостное 92—99
–, ступенчатость 280—281
–, теорема отсчетов 261—264, 274—276
– функций двух переменных 274—276
– – одной переменной 258
–, частота Найквиста 262
Дискретное вейвлет-преобразование (ДВП) 569—571, 583—593 *см. также Вейвлет-преобразование*
Дискретное косинусное преобразование (ДКП) 662—664 *см. также Сжатие изображения, стандарт JPEG*
Дискретное преобразование Фурье (ДПФ) 268—304
– – –, быстрое преобразование Фурье (БПФ) 353—356
– – –, вывод 268—271
– – –, двумерное 284
– – –, дополнение нулями 300—301
– – –, круговая корреляция *см. Корреляция*
– – –, круговая свертка *см. Свертка*
– – –, ошибка перехлеста 299—300
– – –, пара преобразований Фурье 270, 284
– – –, периодичность 285—287
– – –, представление в полярных координатах 302
– – –, разделимость 302
– – –, реализация 351—356
– – –, свойства 284—304
– – –, спектр 254, 264—267, 274—277
– – –, среднее значение 294, 302
– – –, фазовый угол 293, 302
Дискретный единичный импульс 250—251, 272—273 *см. также Импульс*
Дискриминантная (решающая) функция 995
Дискриминантный анализ 860, 991
Дисперсия яркости *см. Момент статистический*
Дистанционное зондирование 36, 331—332, 1002, 1011

Дифференциальная импульсно-кодовая модуляция (ДИКМ) *см. Кодирование с предсказанием*
Единица измерения 75, 92—93, 139, 253, 619—621
– – пространственного разрешения 92—93
– – электромагнитного спектра 75
– – яркостного разрешения 93
Задержка единичная 544
Замыкание 554, 738, 768, 770, 774
–, морфологическая операция 738, 774
–, полутоновая морфология 774
– с реконструкцией 764, 770, 785
Заполнение дырок 746—748, 765—766, 768, 770
Зональное кодирование 669—671
Зрение 23—24, 65—69, 70—74, 462—463, 691 *см. также Зрительное восприятие*
– машинное 23, 28—29, 1040
– скотопическое 68, 70
– фотопическое 67, 70
Зрительное восприятие 66—69, 461—468
– – и обработка цветных изображений 461—468
– –, контрастная чувствительность 70—74
– –, одновременный контраст 73
– –, оптические иллюзии 73—74
– –, отношение Вебера 71
– –, полосы Маха 73
– –, спектральная чувствительность 463
– –, строение глаза 66—69
– –, субъективная яркость 70—73
– –, формирование изображения в глазу 69
– –, яркостная адаптация 65—66, 70—73
Идеальный фильтр *см. Фильтрация в частотной области*
Избыточность 613—618
– визуальная (лишняя информация) 613, 617—618
– временна́я 613, 616—617
– данных относительная 612
– кодовая 613—616
– пространственная 613, 616—617
Измерение информации в изображении 618—621
Изображение акустическое 46—48
– в гамма-лучах 30—31, 45, 77, 79, 213—214
– в диапазоне радиоволн 44—46
– двоичное (двухградационное, бинарное) 105, 119, 146, 157—159, 627, 645—655, 729, 736, 741, 745—750, 805, 841, 852, 920—921, 931—932, 939
– –, граница 104—105
– –, логические операции 117
– –, морфологические операции 728—771
– –, прослеживание границы 920—922
– –, сегментация 515—517, 805—806, 841—842, 895
– –, сжатие 646—650, 655—659
– инфракрасное 36—41, 45, 76—77, 112, 462, 487, 492—493, 798, 949—953, 973, 1010—1011
– магнитно-резонансное (ЯМР) 45, 81, 126, 152—153, 358, 369
– микроволновое 43—44, 487
– многозональное (мультиспектральное) 36—38, 492—496, 952—953, 969—979, 1010—1013
– мозаичное 50
– накопленное разностное (НРИ) 900—901
– негативное 118, 121—122, 147—149
– полноцветное (в натуральных цветах) 460—483, 611—612
– полутоновое (многоградационное) 76, 103, 117—119, 771—787
– радиолокационное 44, 79, 83
– рентгеновское 32—34, 45, 52, 77, 94—96, 113, 115, 156, 200, 341—342, 369, 380, 486, 749—750, 773, 777, 778, 805—806, 847, 883—885, 887—888
–, система координат 87—90
– ультразвуковое 23, 46—47, 77, 432, 453
– ультрафиолетовое 34—36
– фрактальное 49—50
– эталонное 126—129, 1028—1029
Изотропный фильтр 203
Иллюзии оптические 73—74

Импликация в нечетких множествах 224, 227, 229—230
Импульс дискретный 189—192, 251—252, 273
– непрерывный 250—251, 272—273
–, последовательность 251—252, 254—256, 258—263
–, свойство отсеивания 250—252, 255, 272—273
– треугольный 802, 812—813
Импульсная характеристика (отклик) 315, 546—550, 564
Импульсный шум 199—200, 372—374
Импульсы периодические 251 *см. также Импульс, последовательность*
Инверсная фильтрация 411—414
Интеграл Фредгольма первого рода 405
Интенсивность 22, 76, 460—461, 475—483 *см. также Яркость*
Интерполяция 99—102, 122, 124—127, 267—268, 278—281, 539—540, 628, 685—686
– бикубическая 100
– билинейная 100
–, ближайший сосед 99—100
–, передискретизация изображения 99—102
Информации количество 619—620
Искажающий оператор 128, 368
Искажения 367—368 *см. также Восстановление*
– блоковые 666
– линейные, трансляционно-инвариантные 403—406
– модель 367—368
– оценка 406—411
Источник без памяти 619
–, *n*-кратное расширение 620
– марковский 621, 666—667
Кадр двунаправленный (B-кадр) 685—686
– независимый (опорный, I-кадр) 685
– предсказываемый (P-кадр) 685
Карта перепадов 822
Катаракта 66
Качество изображения, шкала оценки 623
Квадродерево 597—599, 886
Квадрообласть 886—888
Квантиль 200, 383, 868—870, 884
Квантование 55, 65, 85—87, 484—488, 618, 625, 659—660, 666, 690—691 *см. также Дискретизация*
– и интерполяция 99—102
–, мертвая зона 704—705
–, уровень квантования 698
–, уровень пороговый 698
Квантователь Ллойда – Макса 670, 699—700
– равномерный оптимальный 700
–, расчет 704—706
Класс образов 991—995
– с нормальным распределением 1005—1013
Классификатор байесовский 1003—1013
–, нейронная сеть 1013—1035
– по минимуму расстояния 996—999
– статистически оптимальный 1003—1013
– структурный 1036—1039
Клеточный комплекс 927—931
Ковариационная матрица 518, 881, 970—979, 1006—1012
Код 613, 614—616, 620—621, 624—625 *см. также Сжатие*
– READ модифицированный 647
– – дважды модифицированный 647
– арифметический 637—640
– блоковый 632
– Голомба 633—637
– Голомба – Райса 635—636
– Грея 655—659
– естественный двоичный 614
– мгновенный 632
– MKKTT окончания 1045
– – составной 1046
– неравномерный 614—615, 620—621
– однозначно декодируемый 632
– Райса 635—636
– стандартный для AC коэффициентов JPEG 1048—1049
– – – DC коэффициентов JPEG 1047
– унарный 633
– Хаффмана 630—633

– – модифицированный 646
– Элайеса (гамма-код) 637
Кода длина 614—615
Кодек 624
Кодер символов 625
Кодирование 543—550, 614—616, 624—625, 630—713 *см. также Сжатие*
– *N*-наибольших 672
– READ модифицированное 647
– – дважды модифицированное 647
– арифметическое, MQ-coder 639
– –, Q-кодер 639
– длин серий 616—617, 624, 644—651
–, избыточность 613—616
–, квантователь 625
–, кодер символов 625
–, методы сжатия изображений 626—630, 630—713
– на основе шаблонов 651—655
–, преобразователь 625
– с предсказанием 679—700
– – – без потерь 680—683
– – –, дельта-модуляция 694—695
– – –, дифференциальная импульсно-кодовая модуляция (ДИКМ) 695—697
– – – и квантование 693—695, 698—700
– – –, компенсация движения 684—693
– – –, оптимальное квантование 698—700
– – –, оптимальный предсказатель 695—698
– – –, ошибка предсказания 679—683, 695—698
– – – с потерями 693—695
– субполосное 543—550
– Хаффмана 630—633
– –, модифицированное 646
Кодирования алгоритм Лемпеля — Зива — Уэлша (LZW) 640—644
Кодовое слово 613
Кодовый символ 613
Компактный носитель 560
Компенсация движения 684—693
Комплексные числа 249—250
Компоненты системы обработки изображений 54—57
Контейнеры для сжатия изображений 626—630
Контраст 23, 28, 32, 91, 99, 113, 134—135, 145, 152—156, 161—162, 168, 181—185, 232—234, 237, 307, 342—344, 876, 958, 961—962, 974—975 *см. также Улучшение изображения*
– локальный 183, 876
–, мера 134, 181, 974
– одновременный 73
–, повышение (усиление) 145—146, 155—156, 168, 232—234, 337, 342—344
Контрастное вещество 32, 112—113, 158
Контрольная (экзаменационная) выборка 1012, 1029—1032
Контур 104—105, 203, 206—208, 211, 216, 280—281, 388, 520—524, 590, 816—854, 867—870 *см. также Граница, Перепад яркости, Обнаружение контуров*
–, градиент 210—212, 520—524, 695, 778—779, 817—826
–, ложный 73, 96—97, 160, 195, 724
– и вейвлет-преобразование 590
–, нормаль 818—819
–, операторы 820
–, пересечение нулевого уровня 203, 813—814, 826—832
–, связывание 840—854
–, улучшение 200—212, 332—342, 778—779
–, цветное изображение 520—524
–, чувствительность к шуму 814—816
Координатные метки 132—133 *см. также Ризо-маркировка*
Координаты пространственные 22, 85—88
Корреляция 188—193, 302, 999—1003 *см. также Свертка*
–, дескриптор 958, 961—963
–, коэффициент 720, 1000
–, сопоставление 999—1003
–, теорема о 303
Котельникова теорема 262 см. *Дискретизация, теорема отсчетов*
Коэффициент корреляции 720, 1000
– округлости 949
– отражения 84, 342—344, 856

– пропускания 84
– сжатия 612
Коэффициенты деталей (горизонтальные, вертикальные, диагональные) 566, 569, 575—581, 584 *см. также Коэффициенты приближения*
– приближения 546—549, 566—569, 575—581 *см. также Коэффициенты деталей*
Кратномасштабная обработка, вейвлеты 536—610
– –, масштабирующие функции 557—562, 583—586
– –, основы 537—550
– –, пирамида изображений 538—543
– –, преобразование Хаара 550—554
– –, разложение 554—566
– –, разложения в ряды 554—557, 566—569
– –, субполосное кодирование 543—550
– –, уравнение КМА 561—562
Кратномасштабный анализ (КМА) 554, 557—562
Критерии верности воспроизведения 621—623
Кусочно-линейная функция преобразования 155—160
Лапласиан 203—207, 304, 338—339
–, восстановление 420
– гауссиана 827—833
–, градационная коррекция 206
–, изотропность 807
–, линий обнаружение 806—811
–, обнаружение 805—811, 826—833
–, оператор 204
–, пересечение нулевого уровня 826—830, 908
–, пирамида 541—542
–, порога выбор 867—870
–, производная второго порядка 203—205
–, резкости повышение 203—207, 213—216, 514
–, сегментация 869—871
–, сравнение с градиентом 209—216
– цветного изображения 514
Линейно разделимый класс 1017—1019
Линий обнаружение 806—811
– пары, разрешение 92—93
Лифтинг-схема 708
Ложный контур 73, 96—97, 160, 195, 724
Локальное среднее значение и дисперсия 122, 181—183, 400—401
Локальной гистограммы обработка 179—181
Ломаная линия минимальной длины (ЛМД) 925—932
–, аппроксимация границы 925
–, разбиения метод 933—934
–, слияния метод 932—933
Макроблок 685—691
Малла иерархический алгоритм 574
Маркер 761—767, 770, 784—786, 868, 896—899
–, морфологическая реконструкция 761—771, 784—786
–, сегментация 896—899
Маркер-изображение 868
Маска 145, 186 *см. также Фильтр пространственный*
– зональная 669—670
– пороговая 671
– фильтра линейного 193
Масштабирование геометрическое 122—129 *см. также Геометрическое преобразование*
– яркостное 115
Масштабирующая функция 554, 557—562, 583—586
– –двумерная разделимая 583—584
– –, коэффициенты 561—562
– – Хаара 558—562
Масштабирующее (уточняющее) равенство 561—562
Масштабная последовательность 561
Масштабное равенство 561—562
Матрица сочетаемости яркостей 956—957
Матричные операции *см. Операция*
Маха полосы 73
Медианный фильтр 141, 199—200, 382—386
– –адаптивный 390—393

Международная комиссия по освещению (МКО) 462—463, 466—467, 504—505, 707
– организация по стандартизации (ISO) 626
– электротехническая комиссия (МЭК) 626
Международный консультационный комитет по телефонии и телеграфии (МККТТ) 626
– союз электросвязи — Сектор стандартизации телекоммуникаций (ITU-T) 626
Межклассовая дисперсия 860—864, 870, 872
Мексиканская шляпа, вейвлет 572
––, оператор 827
Мертвая зона 704—705
Метрика 105
Микроволны 30, 43—44, 75—77, 488
Микроденситометр 80
Микроскоп электронный просвечивающий (ПЭМ) 48
–– сканирующий (СЭМ) 49, 114, 156, 184, 305
Микроскопия в оптическом диапазоне 36
Множество пустое 116
Модель формирования изображения 83—85, 342
– цветовая *см. Цветовая модель*
Модуляция 545—549, 694—695
–, функция 400
Момент статистический 134, 947—948, 954—956, 966—969, 992
–, инварианты 966—969
Морфологическая обработка изображения 728—796
–––, выделение границ 745—746
–––, – связных компонент 748—751
–––, градиент 778—779
–––, гранулометрия 781—783
–––, дилатация 734—737, 768, 772, 784
–––, замыкание 737—742, 774—777
–––, заполнение дырок 746—748, 768
–––, коррекция неравномерности фона 780
–––, начальные сведения 728—732
–––, операции и свойства (сводная таблица) 768—770
–––, основные операции 115—121, 729—732
–––, отражение множества 729
–––, очистка краев 766—767, 770
–––, полутоновая 771—787
–––, попадание/пропуск 742—745, 768
–––, попеременная последовательная фильтрация 777
–––, построение выпуклой оболочки 751—753, 769
–––, – остова 755—758 *см. также Остов области*
–––, преобразование «впадина» (черный выступ) 780
–––, – «выступ» (белый выступ) 779, 786
–––, размыкание 737—742, 764—765, 768, 774—777
–––, реконструкция 761—767, 784—787
–––, сглаживание 777—778
–––, сдвиг (параллельный перенос) 729
–––, структурообразующее множество (примитив) 730
–––, текстурная сегментация 783—784
–––, усечение 758—761, 769
–––, утолщение 754—755, 769
–––, утончение 753—754, 769
–––, фильтрация 734, 741—742, 777—779
–––, эрозия 732—734, 768, 772, 784
Морфологическая реконструкция 761—767, 784—787
––, геодезическая эрозия и дилатация 761—762, 770, 784—785
––, заполнение дырок 765—766, 770
––, очистка краев изображения 766—767, 770
–– полутонового изображения 784—787
––, преобразование «выступ» 785
––, размыкание с реконструкцией 764—765, 770, 786
––, через дилатацию 761—762, 770, 784—785
––, через эрозию 761—762, 770, 784—785
Муар 264, 276, 281—283, 348

Наименьший средний квадрат, дельта-правило 1019
Наложение диапазонов частот (спектров) 264—267, 276—283
– спектров временно́е 276
– –, интерполяция и повторная дискретизация 278—281
– –, муар 281—283
– – пространственное 276
– –, устранение артефактов 262—264, 277—278
– –, фильтрация 262—264, 277—278
Насыщение 91
Насыщенность 465—466, 469, 475—481
Нейронная сеть 1013—1035
– – многослойная без обратной связи 1021—1035
– –, обучающая выборка 1013
– –, обучение 1013—1032
– –, – с обратным распространением ошибки 1024—1032
– –, обучения алгоритм 1017—1021
– –, персептрон 1014—1023, 1033
– –, сложность разделяющих поверхностей 1033—1035
Неопределенности принцип Гейзенберга 582
Непрерывное (интегральное) вейвлет-преобразование 571—573
– – –, масштабирование и сдвиг 571
– – –, условие допустимости 571—572
Неравномерный код 614—615, 630—644
Неразделимые классы 1019—1021
Нерезкое маскирование 207—209, 339—342
Нечеткое множество 120—121, 217—237
– –, импликация 224—227, 229—230
– –, использование 222—232, 232—235, 235—237
– –, начала теории 218—222
– –, объединение 220—221, 226—227
– –, операции над множествами 120—121, 217—222
– –, определение 219—221
– –, устранение нечеткости 227—232
– –, фильтрация пространственная 235—237
– –, функция принадлежности (характеристическая функция) 120, 217—222
– –, яркостное преобразование 232—235
Номер фигуры 942, 1036
Норма евклидова 130, 420, 517, 881, 996
Областей разделение и слияние 886—889 *см. также Сегментация на основе областей*
Области выращивание 882—885 *см. также Сегментация на основе областей*
Области дескриптор 948—969
– –, компактность 949
– –, контраст 958, 961—962
– –, корреляция 958, 961—962
– –, коэффициент округлости 949
– –, максимум вероятности 958, 961
– –, матрица сочетаемости яркостей 956—957
– –, однородность 955—956, 958, 962
– –, периметр 949
– –, площадь 949
– – реляционный 980—983
– – текстурный 954—966
– – топологический 950—953
– –, Эйлера число 951—953
– –, энтропия 955—956, 958, 963
Область 104—105
– интереса 114, 502—504, 710, 760, 843—848, 965
– преобразования (трансформационная) 130, 143
– пространственная, корреляция *см. Корреляция*
– –, определение 88
– –, свертка *см. Свертка*
– –, фильтрация *см. Фильтрация в пространственной области*
– частотная 131, 246, 253
Область-квадрант (квадрообласть) 886—888
Обнаружение контуров 520—524, 811—840 *см. также Контур, Перепад яркости*
– –, градиент с пороговой обработкой 824—826
– –, детектор Кэнни 833—839
– –, – Марра — Хилдрета 826—833
– –, – Превитта 820—822

––, – Робертса 210, 819—821
––, – Собела 211—212, 214, 216, 290, 318, 522, 820—824, 847
––, лапласиан гауссиана 827—833
––, модели 811—816
––, на основе вейвлетов 590
––, наклонный перепад 802—804, 811—815
––, нахождение границы 840—854
––, ошибка ложной тревоги 836
––, – пропуска 836
––, подавление немаксимальных точек 834—837
––, пороговое разделение с гистерезисом 836
––, производные 201—203, 801—804
––, пространственные фильтры 804—811, 819—825
––, связывание контуров 840—854
––, спагетти-эффект 830
––, ступенчатый перепад 802—804, 811—813
––, треугольный импульс 802, 812—813
Обработка гистограммы, гистограммные статистики 181—186
––, глобальная 160—179
––, локальная 179—181
––, цветовые преобразования 509—510
Обработка изображения в натуральных цветах 460, 495—496
Обработка поэлементная 121—122, 145—146
Обработка цветного изображения 460—535
–––, выделение контуров 520—524
–––, вырезание цветового диапазона 502—504
–––, гистограммы видоизменение 509—510
–––, диаграмма цветности 466—468
–––, квантование по яркости 484—488
–––, координаты цвета 465—466
–––, модель CMY и CMYK 468—469, 474—475
–––, – HSI 468—469, 475—483
–––, – $L^*a^*b^*$ (CIELAB) 504—505
–––, – RGB 468—469, 469—474
–––, повышение резкости 514—515
–––, преобразование яркости в цвет 489—494
–––, преобразования 496—510
–––, псевдоцвета 460, 483—494
–––, сглаживание 511—514
–––, сегментация 517—524
–––, сжатие 527—528
–––, цветовая коррекция 504—509
–––, цветовой «градиент» 521—522
–––, цветовые модели 468—483
–––, шум 524—527
Образ 990—995 *см. также Распознавание, Классификатор*
–, дискриминантный анализ 991—992, 995
–, класс образов 990—995
–, линейно разделимые классы 1014—1019
–, нейронная сеть 1013—1035
–, неразделимые классы 1019—1021
–, нормально распределенный класс 1005—1013
–, обучение 990, 1012—1032
–, – путем обратного распространения ошибки 1015, 1024—1032
–, построение вектора признаков 991—995
–, разделяющая поверхность 995—997, 1005—1006, 1008—1009, 1014—1016, 1019, 1033—1035
–, распознавание нескольких классов 1021—1035
–, сопоставление 996—1003, 1036—1039
Обратная проекция 425—428
––с веерным пучком фильтрованная 445—452
––с параллельным пучком фильтрованная 439—445
––фильтрованная 439—445, 445—452
––, эффект ореола 427
Обратное отображение 125
– распространение (ошибки), обучение нейронной сети 1024—1032
Обратное преобразование *см. Преобразование*

––Фурье *см. Фурье преобразование, Дискретное преобразование Фурье*
Обучающая выборка 1012—1021, 1029—1032
Обучение 990, 1012—1032
Объединение нечетких множеств 220—221, 226—227
Объект (на изображении) 23—24, 51—54, 116, 119, 198—199, 502, 537, 615, 729, 742—744, 855—859, 867 *см. также Передний план, Фон, Распознавание*
Одновременный контраст 73
Однородность 108, 404, 955
Окно 145
–Ханна 441
–Хэмминга 441
Окрестность 144—145
Оператор 186, 204, 210—212 819—822, 827
Операция арифметическая 109—115
–векторная 129—130, 193—194, 495
––и матричная 129—130
––, обработка цветных изображений 495—496
––, представление линейной фильтрации 193
–линейная 108—115
–логическая 119—120
–матричная 89, 107—108, 129—130
–над множествами 115—121, 728—742 *см. также Нечеткое множество*
–––, дилатация 734—737
–––, замыкание 737—742
–––, морфологическая обработка 728—761
–––, нечеткое множество 120—121, 217—237
–––, основы 115—121
–––, размыкание 737—742
–––, четкое множестве 120
–––, эрозия 732—734
––окрестностью 122, 144—146, 186—212
–над окрестностью 122, 144—146, 186—212 *см. также Фильтрация в пространственной области*
–нелинейная 108—109
–поэлементная 107—109, 118—122, 144—146, 313, 406, 479, 784—787, 817, 899, 1013
––и матричная 107—108
–пространственная 121—129
Описание 941—983 *см. также Представление, Дескриптор, Области дескриптор*
–, строка символов 993—994, 1036—1039
Ортонормированность 548
Освещение 78—85, 857—858 *см. также Освещенность*
–, источник 78—83
–, коррекция неравномерности 110—111, 780—781, 857—858, 874—880
–неравномерное 110—111, 780—781, 857—858, 874—880
–, сегментация 857—858
–, стандартный источник 505
–структурированное 40, 42
Освещенность 83—85 *см. также Освещение*
–, модель изображения 83—85, 342—343
Остов области (скелетонизация) 938—940
Отображение 125, 166, 171—173, 617—618 *см. также Яркостное преобразование*
–обратное 125, 168, 626
–прямое 125, 166, 171—173, 617—618
Отображения функция 145
Отражение 47, 66, 78, 84, 342—344, 856—858
Отсеивания свойство 250—252, 272—273 *см. также Импульс*
Очистка краев изображения 766—767, 770
Ошибка предсказания 540, 681, 683
Ошибки мера, среднее абсолютное искажение 686
Памяти объем для обработки изображений 56
Пара преобразований 131
––Фурье 253, 273—274, 284
Параметры шума, построение оценок 375—378
Пары длин серий 617, 645
Перегрузка по крутизне 694—695

Передискретизация (избыточная дискретизация) 261, 264, 275—276, 278
– (повторная дискретизация) 99—102, 278—281, 922—925, 1003
Передний план 119—120, 729, 746, 748, 765—766 *см. также Объект, Фон*
Перекрестная модуляция 546—549
Перепад контурный *см. Контур*
Перепад яркости (контур), определение 105
– –, виды 202—203, 802—804
– –, модели 811—817
– –, модуль градиента 209—212, 817—824
– –, наклонный 202—203, 802—803, 811
– –, направление 817—818
– –, ступенчатый 202—203, 802—803, 811
– –, треугольный импульс 802—803, 811
Пересечение нулевого уровня 202—203, 813—814, 826—832
Персептрон 1014, 1018—1024, 1033
ПЗС-матрица 69, 82, 93, 111, 368, 524
Пиксель 22, 53, 89
–, отношения между 102
Пирамида гауссова 540
– изображений 538—539
– лапласианов 541
– ошибок 539—542
– приближений 539—542
– прореживания 540
– усреднения 540
Плотность распределения вероятностей 164—166, 170, 369—374, 634, 682, 699—700, 935, 1004—1005
– – –, гамма-распределение 371
– – –, распределение Гаусса 240, 369—370, 1005
– – –, – импульсного шума 372
– – –, – Лапласа 670, 682, 699—701
– – –, – равномерное 371
– – –, – Рэлея 370
– – –, – экспоненциальное 371
– – –, – Эрланга 371
Поверхность разделяющая 995—997, 1005—1009, 1014—1016, 1019, 1033—1035
– –, сложность 1033—1035
Полный класс функций 120
Полосы Маха 73
Полутоновая морфология 771—787 *см. также Морфологическая обработка изображения*
– –, градиент 778—779
– –, гранулометрия 781—783
– –, дилатация 772—774, 784
– –, замыкание 774—777
– –, преобразование «впадина» 779—781
– –, преобразование «выступ» 779—781
– –, размыкание 774—777
– –, реконструкция 784—787
– –, сглаживание 777—778
– –, текстурная сегментация 783—784
– –, эрозия 772—774, 784
Понижающая дискретизация 539—540
«Попадание/пропуск» преобразование 742—744, 751, 753, 768
Порог 146, 155—156, 338, 515, 854—881 *см. также Пороговое преобразование*
–, байесовский классификатор 860—861, 1003—1013
–, гистерезис 837
– глобальный 858—860
–, использование с градиентом 824—826, 836—840
–, – с лапласианом 867—870
–, – со сглаживанием 198—199, 865—867
– локальный (динамический, адаптивный) 855, 874—880
– многоуровневый 855—856, 870—874
–, несколько переменных 517—520, 880—881
– оптимальный 863, 1003—1013
– переменный 855—856, 874—880
–, цветовая сегментация 517—520
Пороговое кодирование 666—674
Пороговое преобразование 146, 155—156, 338, 515, 591, 671—674, 807—809, 824—826, 830—831, 836—838, 854—881
– – адаптивное 855, 874—880
– –, байесовский классификатор 860—861, 1003—1013
– – глобальное 858—860
– – жесткое 591
– –, использование контуров 867—870
– –, – с градиентом 824—826, 836—840
– –, – с лапласианом 867—870

– – локальное 855, 874—880
– –, кодирование 668—674
– –, мера разделимости классов 862
– – многоуровневое 855—856, 870—874
– – мягкое 591
– – нескольких переменных 517—520, 880—881
– – оптимальное 863, 1003—1013
– –, основы 155—158, 855—858
– –, Оцу метод 860—865, 869—875, 879—880
– –, переменный порог 855—856, 874—880
– –, роль освещения и отражения 857—858
– –, – распределения яркости 855—856
– –, сглаживания применение 198—199, 865—867
– –, сегментация 854—881
– –, скользящее среднее 878—880
– –, точка объекта 855
– –, – фона 855
– –, функция 146, 155—156
– –, шума влияние 856—857
Последовательного удвоения метод 353
Последовательной развертки система 674
Построение перегородок для водоразделов 892—894
Поток лучистой энергии 462
Поэлементные операции *см. Операция*
Правило постоянного коэффициента коррекции 1017—1018
Предикат 798—800, 876—878, 883—889
Предобработка 23, 52, 56, 151—152, 331, 897
Предсказатель по предыдущему элементу 681
Представление 51, 53, 919—940 *см. также Описание*
–, аппроксимация ломаной 925—934
–, остов области 938—940
–, прослеживание границы 920—922
–, сигнатура 934—936
–, цепной код 922—925
Преобразование 122—133, 143—245, 432—438, 550—554, 566—593, 659—679, 742—745, 779—781
– аффинное 123—125 *см. также Геометрическое преобразование*
– вейвлет *см. Вейвлет-преобразование*
– «впадина» 779—781
– «выступ» 779—781
– – с реконструкцией 785—786
– геометрическое *см. Геометрическое преобразование*
–, главные компоненты 969—979
– двумерное линейное 130—133
– к главным осям (ПГО) 938
– Карунена – Лоэва дискретное 665, 972—973
– косинусное дискретное 133, 627, 662—663
– линейное 108—115, 130—133
– логарифмическое 148—150
– морфологическое 728—796 *см. также Морфологическая обработка изображения*
–, морфология полутоновая 777—784
– обратное 168
– «попадание/пропуск» 742—744, 751, 753, 768
– пороговое *см. Пороговое преобразование*
– поэлементное 145—146 *см. также Яркостное преобразование*
– пространственное 121—129, 143—245
– Радона 429, 432—438
– резинового листа 122—125
– степенное (гамма-коррекция) 150—155
– Уолша – Адамара 133, 661—662
– Фурье *см. Фурье преобразование*
– – дискретное (ДПФ) 268—304 *см. также Дискретное преобразование Фурье*
– Хаара 133, 550—554
– Хафа 848—854
– Хотеллинга 971—973
– яркостное 143—146
Преобразований пара 131
Преобразования выбор для блочного трансформационного кодирования 660—666
– ядро 130—133, 145
Приведение к главным компонентам 973
Привязка блока 686—687
Признак 23—24, 51—54, 116, 800, 872, 898, 919, 954, 965, 990—998

Примитив (непроизводный элемент) 993—994
– (структурообразующее множество) 730—732, 771—772 *см. также Морфологическая обработка изображения*
– изотропный 761
– неплоский (неоднотонный) 771—772 *см. также Полутоновая морфология*
– плоский (однотонный) 771—772 *см. также Полутоновая морфология*
Принадлежности функция 120, 217—222
Программное обеспечение обработки изображений 54—57
Проекция (реконструкция изображения по проекциям) 424—452
Производная 201—201, 801—802 *см. также Градиент, Лапласиан*
Прореживание 278—280, 540—541
Пространственные операции 121—129
– переменные 88
Прямое отображение 125
Псевдоцветная обработка изображения 472, 483—494
–––, квантование по яркости 483—494
–––, преобразование яркости в цвет 489—494
Пустое множество 116
Путь дискретный 103
Равного предпочтения кривые 98—99
Разбиение границы на сегменты 936—937
Разделимость классов 862—865, 869—870, 873, 992, 1014—1015, 1017—1021
Разделяющая поверхность 995—997, 1005—1009, 1014—1016, 1019, 1033—1035
Разложение 554—556, 556—569, 594—604, 701—704
–, базисные функции 554
– биортогональное 553, 556
–, вейвлет 593—602, 701—704
–, вейвлет-ряд 566—569, 569—571, 574—575
–, вейвлет-функции 562—566, 569, 583—589
–, выбор уровня для вейвлет-кодирования 703—704
–, дерево анализа вейвлет-пакетов 593—597
–, коэффициенты 554
–, кратномасштабный анализ 554, 557—562
–, масштабирующая функция 557—562
– ортонормированное 553, 556
– переполненное 556
–, ряд 554—557, 566—569
Размывание *см. Фильтр пространственный сглаживающий*
Размыкание 737—742, 756, 758, 764—765, 768, 774—777
– с реконструкцией 764—765, 770, 785—787
Разность первого порядка 923—925, 942—943
Разрешение пространственное 92—99
– яркостное и квантование 92—99
Распознавание 54, 990—1044 *см. также Образ, Классификатор*
–, байесовский классификатор 1004—1013
–, выбор признаков 992
–, дискриминантный анализ 991
–, классификатор минимума расстояния 996—999
–, – оптимальный 1003—1013
–, коэффициент корреляции 1000
–, нейронные сети 1013—1035
–, образы 990—1035
–, обучение 990, 1012—1032
–, сопоставление 996—1003, 1036—1039
–, – корреляционное 999—1003
–, – номеров фигур 1036—1037
–, – строк символов 1037—1039
–, структурные методы 1036—1039
–, теория принятия решений 995—1036
Расстояние D_4 (метрика L_1) 106
– D_8 (метрика L_∞) 106
– евклидово 105, 129—130 *см. также Расстояния меры*
– Махаланобиса 881 *см. также Расстояния меры*
– между пикселями 105—107
Расстояния меры 105—107, 129—130, 517, 881, 935, 941, 951, 996—999, 1008, 1036
Растровый элемент воспроизведения полутонов 282—283

Регистрация изображения 51—52, 55, 78—87, 91, 112, 114, 126, 410, 460, 471, 475, 922, 949, 969—970
Резкости повышение 200—217, 332—346 *см. также Фильтр пространственный подъема высоких частот*
Реконструкция (деконволюция) 81, 264, 267—268, 424—452, 761—767
–, восстановление функции из отсчетов 267—268
– вслепую 407
–, компьютерная томография (КТ) 428—432 *см. также Томография*
–, ламинограмма 437
– морфологическая 761—767, 784—787
– – полутоновая 784—787
–, обратная проекция 425—428, 436—452
–, – – с веерным пучком 445—452
–, – – с параллельным пучком 439—445
– по проекциям 424—452
–, Радона преобразование 432—438
–, Рам – Лак фильтр 440
–, синограмма 435—437
–, теорема о центральном сечении 438—439
–, Шеппа – Логана фантом 435—436, 454
Рентгеновское излучение 27—30, 32—34, 45, 76—77, 79, 81, 112, 341, 425, 428—432, 487, 489, 883—885, 887
Рентгенография с использованием масок 112
Ризо-маркировка 27, 127 *см. также Совмещение изображений*
Рэлея шум 371
Свертка 188—193, 256—257, 270, 302 *см. также Корреляция*
– круговая 270, 298
– линейная 188—193, 256—257
– пространственная дискретная 188—193
– – непрерывная 256—257
–, интеграл 405
–, маска 193
–, теорема о 257, 298—301, 303, 314, 406, 444, 912, 1000
–, фильтр 192
–, цифровая фильтрация 543—546
– ядро 192
Свет 74—77, 461—468 *см. также Электромагнитный спектр*
– ахроматический (монохроматический) 76, 461—463
– и зрение *см. Зрительное восприятие*
–, поглощение 464
– хроматический (цветной) 76, 461
–, цвета первичные и вторичные 462—464
Светлота 462, 465, 475
Световой поток 76, 462
Связная компонента 103
– –, выделение 748—751, 768
– –, описание 951—953
– –, сегментация 882—885, 892—894
Связное множество 103
Связные элементы (пиксели) 103
Сглаживание 195—199, 320—346, 511—514, 777—778, 816 *см. также Фильтр пространственный сглаживающий*
– морфологическое 777—778
Сегментация 797—918
– избыточная 896—898
–, линий обнаружение 806—811
– на основе движения 899—907
– – – контуров *см. Обнаружение контуров*
– – – областей 881—889
– – – –, выращивание 882—885
– – – –, разделение и слияние 886—889
– – – пороговой обработки 854—889 *см. также Пороговое преобразование*
– – – текстур 888—889, 954—969
–, определение 797—798
–, основы 798—804
– по водоразделам 889—899 *см. также Водораздел*
–, точек обнаружение 805—806
– цветного изображения 515—524
–, частотные методы 903—907
Сенсор (чувствительный элемент) 55, 78—82, 85 *см. также Регистрация изображения*
–, линейка 80—81, 87
–, матрица 82, 87
– одиночный 79—80, 86—87
–, считывание и регистрация изображения 78—85

Сжатие 53, 527—528, 611—727
–, арифметическое кодирование 637—640
–, блочное трансформационное кодирование 659—679
–, вейвлет-кодирование 700—713
– видео, контейнер Quicktime 626, 629
– –, метод HDV 626, 629, 680
– –, стандарт DV 626, 628
– –, стандарты H.261, H.262, H.263 и H.264 626, 628, 638, 659, 680, 690
– –, – MPEG-1, MPEG-2, MPEG-4 AVC (H.264) 627—634, 638—639, 644—645, 659—660, 680—681, 690—691
– –, формат AVS 626, 629
– –, – M-JPEG 626, 629, 644, 659
– –, – VC-1 626, 629, 659, 680, 690
– –, – WMV9 626, 629, 690
–, временна́я избыточность 613, 616—617
– изображения, стандарт JBIG 626, 627, 638, 655
– –, – JBIG-2 626, 627, 630, 638, 644, 680
– –, – JPEG 626, 628, 644, 659, 674—679, 680
– –, – JPEG-2000 626, 628, 638, 655, 706—713
– –, – JPEG-LS 626, 628, 680
– –, – MKKTT 646—651
– –, формат BMP 626, 629, 644
– –, – GIF 626, 629,
– –, – PDF 626, 629, 641, 652, 655, 657
– –, – PNG 626, 629, 641
– –, – TIFF 626, 629, 641, 645
–, квантование 618, 625
–, код Голомба 633—637
–, – Хаффмана 630—633
–, кодирование битовых плоскостей 655—659
–, – длин серий 644—651
–, – на базе шаблонов 651—655
–, – с предсказанием 679—700
–, кодовая избыточность 613, 614—616
–, контейнер 626—630
–, коэффициент 612
–, критерии верности воспроизведения 621—623
–, лишняя информация 613—614, 617—618
–, метод кодирования Лемпеля – Зива – Уэлша (LZW) 640—644
–, методы 630—713
–, модели 624—625
–, основные понятия 612—623
–, пространственная избыточность 613, 616—617
–, стандарты 626—630
–, форматы 626—630
–, цветное изображение 527—528
Сигнал-шум отношение 416, 622
Сигнатура 934—936
Симлет 587—589
Синтетическое (искусственное) изображение 29, 49—50
Система линейная 108—109, 190, 256, 367—368, 403—406
Скелетонизация *см. Остов области*
Скользящее среднее 241, 878—880
– – в пороговой обработке 878—880
Скорость передачи данных 625
Скотопическое (сумеречное) зрение 68
Слепое пятно 67—68
Смежность пикселей 102—105
Совмещение изображений 110, 126—129, 131, 605, 900, 969—970
Соответствие пространственных и частотных фильтров 314—320
Сопоставление 996—1003, 1036—1039
– корреляционное 999—1003
– номеров фигур 1036—1037
– строк символов 1037—1039
Соседство пикселей 102
Спектр видимого света 75, 462
Спектр *см. Фурье спектр, Дискретное преобразование Фурье*
– энергетический 293, 302, 321, 415, 418
Среднеквадратическое (стандартное) отклонение (СКО) 370, 416, 621—622
Средний квадрат отклонения, минимизация 415
– риск 1003
Средняя яркость *см. Момент статистический*

Стандарты сжатия изображений 626—630 *см. также Сжатие изображения*
Статистический момент *см. Момент статистический*
Стохастические (вероятностные) методы обработки изображений 133—135
Структурообразующее множество *см. Примитив*
Ступенчатость контуров 280—281
Субполосное кодирование 543—550
Сумма (интеграл) вдоль луча 425
Считывание изображения 78—85 *см. также Регистрация изображения*
Тайл-компонента 707
Текстура 783—784, 888—889, 898, 954—966
–, дескриптор 954—966
–, использование гистограммы яркости 954—956
–, матрица сочетаемости 956—961
–, морфологический градиент 783—784
–, сегментация 783—784, 888—889
–, спектральный подход 954, 964—966
–, статистический подход 954—964
–, структурный подход 963—964
Телевидение высокого разрешения (HDTV) 139, 612, 628
– стандартного разрешения 611
Тематические зоны спектра 36
Теорема кодирования канала без шума 620
– о корреляции 303
– о свертке 257, 298—301, 303, 314, 406, 444, 912, 1000
– о центральном сечении (о проекциях) 438—439 *см. также Реконструкция*
Теория информации 618—621
Тихоновская регуляризация 420
Томография 27—28, 30, 34, 81, 153, 345 *см. также Реконструкция, компьютерная томография*
– компьютерная (аксиальная, КТ, КАТ) 27—28, 428—432, 453
– – с электронным лучом 431
– магнитно-резонансная (МРТ, ЯМР) 81, 126, 153, 432, 453
– позитронная эмиссионная (ПЭТ) 30, 81, 126, 345, 432, 453
Топологический дескриптор 950—953
Точек обнаружение 805—806
Точка привязки 126
Трансформационное кодирование блочное *см. Блочное трансформационное кодирование*
Увеличение и уменьшение изображения 99, 122—124, 278 *см. также Дискретизация повторная (передискретизация)*
Улучшение изображения 52, 146—147, 152, 156, 160, 170—186, 233—234, 248, 504—517
– –, адаптивное 170, 386—393
– –, вычитание изображений 112—113
– –, гистограммы видоизменение 160—186
– –, гомоморфная фильтрация 342—346
– –, комбинированные методы 212—217
– –, локальные методы 179—181, 183—186, 386—390
– –, медианный фильтр 199—200, 382—383, 390—393
– –, нечеткие методы 232—237
– –, повышение контраста 145—146, 155—156
– –, – резкости 200—212, 332—346
– –, пространственные фильтры 186—212
– –, сглаживание 186, 194—198, 320—332
– –, усиление контраста 150—155, 155—156, 232—237, 342—346
– –, усреднение изображений 110—112
– –, фильтр на основе порядковых статистик 199—200, 382—386, 390—393
– –, частотная область 306—351
– –, яркостные преобразования 147—158, 162—179
Ультрафиолет 34—36, 67, 77, 462
Уменьшение изображения *см. Увеличение и уменьшение изображения*
Унарный код 633
Уолша – Адамара преобразование 133, 661—662
Уровень серого *см. Яркость*
Усечение 758—761, 769

Условные цвета *см. Псевдоцветная обработка изображения*
Усреднение по окрестности 194—196, 201, 496, 512, 801
Утолщение 754—755, 769
Утончение 753—754, 769
Фазовый угол *см. Фурье преобразование, Дискретное преобразование Фурье*
Фигур нумерация 942—943, 1036—1037
Фиксированные RGB-цвета 472—474
Фильтр адаптивный *см. Фильтр пространственный*
– Баттерворта 320, 324—327, 336, 346—349
– – высоких частот 333—334, 336, 337
– – заграждающий (режекторный) 346—349, 393—394, 396—397
– – низких частот 324—327, 328
– – полосовой 346—347
– – узкополосный 347—348, 396—397
– –, повышение резкости 333—334, 336, 337
– –, сглаживание 324—327
– высоких частот 200—212, 339—342
– – – пространственный *см. Фильтрация в пространственной области*
– – – частотный *см. Фильтрация в частотной области*
– заграждающий (режекторный) 346—347, 349—350, 393—395
– интерполяционный 540
– максимума 194, 200
– минимума 200
– на основе порядковых статистик 199—200, 382—386, 390—393
– низких частот пространственный *см. Фильтрация в пространственной области*
– – – частотный *см. Фильтрация в частотной области*
– полосовой 346—347, 395—396
– пространственный *см. также Фильтрация в пространственной области*
– –, векторное представление 193
– – взвешенного среднего 196
– – градиента 209—212
– – изотропный 203
– – лапласиан 203—207
– – линейный 186—188, 193
– – локальный адаптивный 386—390
– – максимума 200, 383
– –, маски формирование 193—194
– – медианный 199, 382—383
– – – адаптивный 390—393
– – минимума 200, 383
– – нерезкого маскирования 207—209
– – низкочастотный 195
– –, основы 143—146
– – повышения резкости 200—212
– – подъема высоких частот 207—209
– – порядковых статистик 199—200, 382—386
– – сглаживающий 194—200, 378—393
– – срединной точки 383
– – среднего арифметического 378—379
– – – гармонического 379
– – – геометрического 379
– – – контрагармонического 379—380
– – усеченного среднего 384
– Рам – Лак 440
– реконструкции (реконструирующий) 264, 406
– с конечной импульсной характеристикой (КИХ-фильтр) 315
– с нулевым сдвигом фазы 312, 333, 347
– срединной точки 383
– среднего *см. Фильтр пространственный*
– точного восстановления 546—549
– усеченного среднего 384
Фильтр цифровой 543—550 *см. также Фильтр пространственный, Фильтрация в пространственной области*
– – с конечной импульсной характеристикой (КИХ) 315, 544—545
– – точного восстановления 546
– –, JPEG-2000 необратимый 706
– –, банк фильтров 545
– –, биортогональный 547, 602—604
– –, – вейвлет Коэна – Добеши – Фово 602—604
– – время-инвертирующий 545, 548
– – знако-инвертирующий 545
– –, импульсная (передаточная) характеристика 546—550
– –, коэффициенты 546

––, –Хаара 578
––модулирующий 545
––ортонормированный 548—550, 564, 578, 580
––, порядок фильтра 544
––, прототип 548
––, свертка 543
––, симлет ортонормированный 4-го порядка 588—589
––, элемент 543—544
–, окно Ханна 441
–, –Хэмминга 441
–, передаточная функция 306
–, цифровой 8 элементный ортонормированный Добеши 549
–, частотная область *см. Фильтрация в частотной области*
–, ядро 186, 192 *см. также Фильтр пространственный*
Фильтрация в пространственной области 143—245, 378—393 *см. также Фильтр пространственный, Фильтрация в частотной области*
––––адаптивная локальная 386—390
––––––медианная 390—393
––––линейная 186—199
––––нелинейная 199—200, 378—393
––––, векторное представление 193
––––, корреляция 188—193
––––, маски 193—194
––––, механизмы 186—188
––––, нечеткие множества 217—237
––––, определение 145
––––, основы 186—194
––––, повышение резкости 200—212
––––, подавление шума 379—394
––––, порядковые статистики 199—200, 382—386
––––, свертка 188—193
––––, сглаживание 195—199
––––, улучшения методы 212—217
Фильтрация в частотной области 304—351 *см. также Фильтрация в пространственной области*
––––, гауссов фильтр 315—317, 327—330, 336—337, 346—347, 393, 396
––––, гомоморфный фильтр 342—346
––––, заграждающий фильтр 346—351, 393—403
––––, идеальный фильтр 263—265, 275, 310—311, 320—324, 334—335, 346, 393, 396—397
––––, лапласиан 338—339
––––, нерезкое маскирование 339—341
––––, основы 306—313
––––, повышение резкости 332—346
––––, полосовой фильтр 346—351, 393—403
––––, последовательность шагов 313
––––, прямоугольная функция 353—354
––––, связь с пространственной фильтрацией 314—320
––––, сглаживание 320—332
––––, узкополосный фильтр 346—351, 393—403
––––, усиление (подъем) высоких частот 339—342
––––, фильтр Баттерворта 320, 324—327, 328, 333—334, 336—337, 346—349, 393—396
––––, –высоких частот 307, 332—338
––––, –низких частот 263—264, 307, 320—332
Фильтрация и обратное проецирование в параллельных пучках 425—427, 439—445
–линейная 193, 298
–методом минимизации сглаживающего функционала со связью 418—424
–морфологическая 734, 741—742, 777—778, 782
–нелинейная 186, 194—195, 199—200, 209—212, 213, 717
–с подъемом высоких частот 207—209, 339—342
Фильтр-пробка 346
Флуоресцентная микроскопия 34
Фон (фоновая область) 104—105, 113—114, 119, 729 *см. также Передний план, Объект*
–, выравнивание 113—114, 781, 785—787, 878—880
Формат сжатия изображений *см. Сжатие изображения, формат*

Форматы файлов, контейнеры, стандарты сжатия 626—630
Формирование изображения 29—51, 78—85 *см. также Считывание изображения, Регистрация изображения*
Фотон 29—30, 34, 75
Фотопреобразователь 79
Фрейм жесткий 557
Функции базисные 536, 550—551, 554—556, 581—583, 661—663, 700—702
– – ДКП 660—663
– –, разложение в ряд 554
– – Уолша — Адамара 661—662
– – Хаара 551—552
Функция активации *см. Активирующая функция*
– вероятности 634
– дискриминантная (решающая) 995
– маскирующая 664—665
– рассеяния точки 405
– с ограниченной полосой частот 261—263, 264—266, 274—276
– стоимости 599—600
Фурье преобразование 252—304
– – дискретное *см. Дискретное преобразование Фурье*
– – непрерывное 252, 273
– –, дискретизация 257—268, 274—284
– –, история 247—248, 358
– –, пара 132, 252—253, 257, 269—270, 273—274, 284—285, 290—291, 303, 1000
– –, свертка *см. Свертка*
– –, энергетический спектр 293, 302, 321, 415, 418
– ряд 250, 255—256
– спектр 149—150, 254, 293
– спектр, логарифмическое преобразование 149—150
– –, фазовый угол 293
Фурье-дескрипторы 944—947
Фурье-образ 252
Фурье-пара 253
Цвет, основы теории 461—468
–, фиксированный RGB 472—474
–, фиксированный Web 472
Цвета основные вторичные 463
– – первичные 462—464
Цветное изображение, обработка 483—524
Цветность 465
–, координаты 465—465
Цветовая модель 468—483
– – CMY и CMYK 474—475
– – HSI 475—483
– – –, использование 481—483, 496—500, 509—510, 515—517
– – –, концепции 475—479
– – –, сегментация 515—517
– – $L^*a^*b^*$ (CIELAB) 504—505
– – RGB 469—474
– – –, сегментация 517—520
– – равноконтрастная (колориметрическая) 505
– –, преобразование RGB—HSI 479—483
– –, фиксированные RGB-цвета 472—474
– –, цветовой куб RGB 470—471
Цветовое преобразование 483—494, 496—510
– –, вырезание цветового диапазона 502—504
– –, гистограммы видоизменение 509—510
– –, постановка задачи 496—500
– –, цветовое дополнение 501—502
– –, цветовой круг 500
– –, яркостная и цветовая коррекция 504—509
Цветовой охват (гамут) 467—468
– тон 469, 475—477, 479—482
Цветовые координаты (L^*, a^*, b^*) 504—505
– – (R, G, B) 462—463, 469—474
– – (X, Y, Z) 465—466
Центральная точка последовательности 288
Центральное отражение 729—730, 734, 737, 739, 772, 774—775
Центроид 882, 934, 977—978
Цепной код Фримена 922—925
Цифровая обработка изображений 22—24
– – –, высокоуровневые процессы 23
– – –, история 24—29
– – –, области 29—51

———, основы 65—142
———, сенсоры 55, 78—82, 85
———, стадии 51—54
——сигналов (ЦОС) 543—545
Цифровое изображение, определение 22
Цифровой видео-диск (DVD) 611
—водяной знак, нанесение 713—721 *см. также Водяной знак*
———, необходимость 713—714
—фильтр *см. Фильтр цифровой*
Частота дискретизации 93, 137, 258
—Найквиста 262, 264, 577 *см. также Дискретизация*
—среза 321
Частотная область 246—247, 253
——, дискретизация *см. Дискретизация*
——, дополнительные характеристики 304—306
——, импульс *см. Импульс*
——, использование движения при сегментации 903—907
——, наложение спектров *см. Наложение диапазонов частот (спектров)*
——, преобразование Фурье *см. Фурье преобразование, Дискретное преобразование Фурье*
——, ряд Фурье 250, 255—256
——, свертка *см. Свертка*
——, спектр Фурье
——, фильтрация *см. Фильтрация в частотной области*
Частотно-временное покрытие изображения 582—583, 595—596
Частотный интервал 271—272
—спектр 254, 293, 274—276 *см. также Фурье спектр, Дискретное преобразование Фурье*
——, БВП 576—577, 594—598
——, вейвлет-пакет 594—598
——, субполосное кодирование 545—550
Четные и нечетные функции 287
Число точек (пикселей) на единицу длины 92, 283
Чувствительный элемент *см. Сенсор*
Шаблон 651—655
Шеннона теорема отсчетов 261—264
—формула 619
Шеппа – Логана фантом 435—436, 451
Шкала полутонов 76, 86 *см. также Яркость*
Шум 86, 91
—белый 369, 416, 591, 833
—биполярный импульсный 372
—выпадения 372
—гамма 371
—гауссов 369
—импульсный 199, 372
—квантования
—, модели 369—374
—на цветном изображении 524—527
—, оценка параметров 375—378
—периодический 374—375, 378, 393—403
—пиковый 372
—, плотность распределения вероятностей 369—372
—, подавление (устранение) 110 *см. также Фильтр пространственный сглаживающий*
—, пространственные и частотные свойства 369
—равномерный 371—372
—Рэлея 370
—«соль и перец» 199, 372
—униполярный 372
—экспоненциальный 371
—, энергетический спектр 415
—Эрланга (гамма шум) 371
Эйлера формула 249, 952
—число 951—953
Экспоненциальный код Голомба 633—637
Электромагнитный спектр 23, 30—46, 74—77
——, видимый диапазон 35—43, 76—77
——, гамма-излучение 30—31, 77
——, диапазон радиоволн 44—46, 77
——, единицы измерения 75
——, инфракрасные диапазоны 35—43, 77
——, источники изображений 29
——, микроволновый диапазон 43—44, 77
——, рентгеновское излучение 32—34, 77
——, свет 74—77
——, ультрафиолетовый диапазон 34—35
——, формирование изображений 30—46

Электронная микроскопия 28—29, 46, 48—49, 77, 79, 114, 156, 184, 305, 365,
Элемент изображения *см. Пиксель*
Энтропия 610, 619—620, 631, 646, 682, 955—956, 958, 962
Эрозия 732—734, 768, 772, 784
Ядро преобразования 131, 145, 186, 192, 368, 405—408, 411, 660—663
– – разделимое 132, 661
Ядро преобразования симметричное 132, 661
Яркости разрыв 799—800
Яркостное преобразование (градационное) 145
– –, вырезание битовой плоскости 158—160
– –, – диапазона яркости 156—158
– – кусочно-линейное 155—158
– – логарифмическое 148—149
– –, негатив 147—148
– –, нечеткие методы 232—235
– –, повышение контраста 145—146, 155—158
– –, приведение (задание) гистограммы 170—179
– – степенно́е (гамма-коррекция) 150—155
– –, эквализация гистограммы 162—170
Яркость 22, 70—71 *см. также Интенсивность*
–, адаптация зрения 70—71
–, дисперсия 181—182 *см. также Моменты статистические*
–, квантование 85—87
–, масштабирование 147—155
–, обработка цветных изображений 461, 469, 471
–, пороговое разделение 854—881
–, преобразование *см. также Яркостное преобразование*
–, среднее значение *см. также Моменты статистические*
–, статистические дескрипторы 133—135, 181—186
– субъективная 70
–, функция отображения 125—127, 145—149, 162—165, 497
–, хроматический свет 76, 461
–, шкала полутонов 86
Ячейка Гейзенберга 582

Производство книг на заказ
Издательство «Техносфера»
тел.: (495) 234-01-10
e-mail: knigi@technosphera.ru

Реклама в книгах:
• модульная
• статьи

Подробная информация о книгах на сайте
http://www.technosphera.ru

Рафаэл С. Гонсалес, Ричард Е. Вудс
Цифровая обработка изображений
Третье издание, исправленное и дополненное

Компьютерная верстка – С.С. Бегунов
Корректор – Н.А. Шипиль
Дизайн книжных серий – С.Ю. Биричев
Дизайн – А.А. Давыдова
Выпускающий редактор – О.Н. Кулешова
Ответственный за выпуск – С.А. Орлов

Формат 70 x 100/16. Печать офсетная.
Гарнитура Ньютон
Печ.л.75. Тираж 3000 экз. (1-й завод 1500 экз.) Зак. №
Бумага офсет. №1, плотность 65 г/м^2

Издательство «Техносфера»
Москва, ул. Краснопролетарская, д.16, стр.2

Отпечатано в типографии филиала ОАО «ТАТМЕДИА»
«ПИК «Идел-Пресс»
420066, г. Казань, ул. Декабристов, 2