

**ЛУЧШИЙ ЗАРУБЕЖНЫЙ УЧЕБНИК**



**Л. ШАПИРО, Дж. СТОКМАН**

# **КОМПЬЮТЕРНОЕ ЗРЕНИЕ**



ИЗДАТЕЛЬСТВО

**БИНОМ**

# КОМПЬЮТЕРНОЕ ЗРЕНИЕ

# COMPUTER VISION

**Linda G. Shapiro**

*Department of Computer Science and Engineering  
Department of Electrical Engineering  
University of Washington  
Seattle, Washington  
shapiro@cs.washington.edu*

**George C. Stockman**

*Department of Computer Science and Engineering  
Michigan State University  
East Lansing, Michigan  
stockman@cse.msu.edu*

Prentice  
Hall

PRENTICE HALL, Upper Sadle River, New Jersey 07458



ЛУЧШИЙ ЗАРУБЕЖНЫЙ УЧЕБНИК

Л. ШАПИРО, Дж. СТОКМАН

# КОМПЬЮТЕРНОЕ ЗРЕНИЕ

Перевод с английского

**А. А. Богуславского**

под редакцией

**С. М. Соколова**

Рекомендовано

учебно-методическим объединением вузов Российской Федерации  
по образованию в области прикладной информатики  
в качестве учебного пособия для студентов  
высших учебных заведений, обучающихся по специальности  
«Прикладная информатика (в областях)»

3-е издание (электронное)



Москва

БИНОМ. Лаборатория знаний

2015



УДК 519.7  
ББК 32.81  
Ш23

*Серия основана в 2006 г.*

### **Шапиро Л.**

Ш23 Компьютерное зрение [Электронный ресурс] / Л. Шапиро, Дж. Стокман ; пер. с англ. — 3-е изд. (эл.). — Электрон. текстовые дан. (1 файл pdf : 763 с.). — М. : БИНОМ. Лаборатория знаний, 2015. — (Лучший зарубежный учебник). — Систем. требования: Adobe Reader XI ; экран 10".

ISBN 978-5-9963-3003-4

В данной книге теоретические аспекты обработки зрительных данных рассматриваются с привлечением большого количества примеров из практических задач. Наряду с классическими темами, в книге рассматриваются базы данных изображений и системы виртуальной и дополненной реальности. Приведены примеры приложений в промышленности, медицине, землепользовании, мультимедиа и компьютерной графике.

Студентам старших курсов и аспирантам, интересующимся современным состоянием дел в машинном зрении.

**УДК 519.7**  
**ББК 32.81**

**Деривативное электронное издание на основе печатного аналога:** Компьютерное зрение / Л. Шапиро, Дж. Стокман ; пер. с англ. — М. : БИНОМ. Лаборатория знаний, 2006. — 752 с. : ил., [8] с. цв. вкл. — (Лучший зарубежный учебник). — ISBN 5-94774-384-1.

**В соответствии со ст. 1299 и 1301 ГК РФ при устранении ограничений, установленных техническими средствами защиты авторских прав, правообладатель вправе требовать от нарушителя возмещения убытков или выплаты компенсации**

**ISBN 978-5-9963-3003-4**

Authorized Translation from the English language edition, entitled COMPUTER VISION; by LINDA SHAPIRO; and by GEORGE STOCKMAN; published by Pearson Education, Inc, publishing as Prentice Hall. Copyright © 2001 by Prentice Hall, 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. Electronic RUSSIAN language edition published by BKL PUBLISHERS. Copyright © 2013.

Авторизованный перевод издания на английском языке, озаглавленного COMPUTER VISION, авторы LINDA SHAPIRO и GEORGE STOCKMAN, опубликованного Pearson Education, Inc, осуществляющим издательскую деятельность под торговой маркой Prentice Hall © 2001 by Prentice Hall, Inc. Все права защищены. Воспроизведение или распространение какой-либо части/частей данной книги в какой-либо форме, какими-либо способами, электронными или механическими, включая фотокопирование, запись и любые поисковые системы хранения информации, без разрешения Pearson Education, Inc запрещены. Электронная русскоязычная версия издана BKL Publishers, Copyright © 2013.

© Перевод на русский язык, БИНОМ. Лаборатория знаний, 2006

---

# Предисловие редактора перевода

Системы технического зрения (СТЗ) призваны и во многих случаях уже решают задачи по дополнению или даже замене человека в областях деятельности, связанных со сбором и анализом зрительной информации. Уровень их использования в прикладных областях является одним из наиболее ярких и наглядных интегральных показателей уровня развития высоких технологий в самых различных отраслях промышленности.

Составные части СТЗ: оптическая система, преобразователь свет-сигнал, электронный тракт ввода элементов изображения в память ЭВМ или специального вычислителя, математическое обеспечение сбора и обработки необходимых зрительных данных, выдвигают самые высокие требования к соответствующим направлениям научно-технической мысли.

Несмотря на значительные успехи, достигнутые в области машинного зрения, эффективное его использование в качестве средства автоматизации приходится, прежде всего, на наиболее развитые производства с общей высокой культурой и технологией. В плохо организованных средах и исследовательских задачах на СТЗ возлагается роль дополнения, расширяющего возможности и повышающего эффективность человеческого зрительного анализа, при ведущей и определяющей роли человека-оператора или исследователя. Следует констатировать тот факт, что не существует универсального математического аппарата, который позволил бы сформировать общий формализованный подход к построению систем технического зрения. Поэтому, с точки зрения подготовки специалистов в области машинного зрения, очень важны публикации, в которых отдельные математические средства и модели рассматриваются в контексте системного подхода к решению той или иной практической задачи.

После полутора-двух десятилетий пониженного спроса в отечественной промышленности на новые разработки в области высоких технологий сейчас стали проявляться признаки оживления спроса на средства автоматизации производства и как на важную их часть — СТЗ. Не только для разработки систем машинного зрения, но и для их эффективного использования необходимо иметь специальную подготовку и понимание того, как работают подобные системы. Вместе с тем, по сравнению с другими областями информационных

технологий, относительно мало монографий, посвященных машинному зрению или отдельным его составляющим. Еще меньшее число переведено на русский язык, а отечественные публикации представлены в основном статьями в различных сборниках.

Здесь, как нельзя кстати, книга Линды Шапиро и Джорджа Стокмана, адресованная, прежде всего, студентам старших курсов и аспирантам, интересующимся современным состоянием дел в такой интересной и увлекательной области, как машинное зрение. Одним из достоинств данной книги является то, что при акценте на математическом обеспечении, в ней сбалансированно представлены и все другие составные части систем технического зрения, требующие учета при разработке или эксплуатации СТЗ. Теоретические аспекты обработки зрительных данных рассматриваются с привлечением большого количества примеров из практических задач, кроме того, во всех частях дано большое количество упражнений, закрепляющих усвоенный материал и вырабатывающих навыки решения практических задач. Наряду с классическими темами, в книге рассматриваются базы данных изображений и системы виртуальной и дополненной реальности. Представлен законченный обзор двух систем компьютерного зрения прикладного назначения. Приведены примеры приложений в промышленности, медицине, землепользовании, мультимедиа и компьютерной графике. На Web-сайте для поддержки книги находятся архивы изображений, исходные тексты программ для примеров обработки изображений и слайды презентаций по тематике книги.

Следует отметить очень качественно выполненный перевод предлагаемой книги. Все термины тщательно выверены с точки зрения их профессионального использования именно в рассматриваемой области.

Выбор терминов, которые наиболее точно соответствуют английскому слову в такой быстро развивающейся области, как машинное зрение, зачастую является довольно сложной задачей. В ряде случаев переводчик вынужден приводить несколько наиболее подходящих значений русских аналогов, а в ряде случаев, в соответствии с уже сложившейся практикой, принимается английский вариант в русской транскрипции, как например, со словом *pixel*, что в смысловом переводе означает элемент изображения, но де-факто используется без перевода — пиксел (реже пиксель).

*проф. С. М. Соколов*

---

# Предисловие к русскому изданию

Эта книга была написана в качестве учебника по компьютерному зрению, рассчитанного на студентов и аспирантов. Мы стремились достичь трех главных целей. Во-первых, мы хотели представить в книге наиболее важный базовый материал по основным разделам компьютерного зрения, который необходим студентам, желающим работать в данной области. Во-вторых, в книгу были включены ряд алгоритмов и прикладных задач повышенной сложности, на примере которых студенты и аспиранты смогут получить представление о некоторых направлениях современных исследований. И в-третьих (хотя и не в последнюю очередь), мы надеялись поделиться с читателями нашей увлеченностью компьютерным зрением, которое продолжает развиваться даже быстрее, чем мы ожидали. Мы выражаем благодарность к. ф.-м. н. А. А. Богуславскому и д. ф.-м. н. С. М. Соколову, работавшим над русским изданием книги. Мы надеемся, что книга будет полезна при изучении компьютерного зрения российскими студентами и аспирантами и что она будет способствовать расширению взаимодействия между американскими и российскими учеными и инженерами. Достигнутые ими результаты, несомненно, необходимы для прогресса в этой области.

*Линда Шапиро и Джордж Стокман*

---

# Предисловие

Эта книга представляет собой вводное руководство по компьютерному зрению для широкого круга читателей. В ней содержится необходимый теоретический материал и примеры для студентов и инженеров, планирующих работать в прикладных областях, в которых требуется автоматически извлекать из изображений некоторую существенную информацию. Материалы книги могут быть полезны и для профессионалов; книга может использоваться в качестве учебника для студентов и для начального обучения аспирантов, а также при выполнении исследовательских проектов в колледжах и в высшей школе. Наша цель состояла в том, чтобы привести базовый набор основных понятий и алгоритмов, а также обсудить некоторые наиболее интересные прикладные области. Данная книга уникальна наличием глав по таким захватывающе интересным и развивающимся в последнее время прикладным областям, как базы данных изображений (гл. 8) и системы виртуальной реальности (гл. 15). В заключительной главе (гл. 16) приведено подробное рассмотрение двух реальных систем, в которых применяется компьютерное зрение. Прогресс в области вычислительной техники привел к повсеместному распространению разнообразных недорогих приложений, связанных с использованием компьютерных изображений. Вычислительная обработка изображений теперь является не только предметом научных исследований. Она применяется даже в искусстве и в общественных науках и представляет интерес для отдельных любителей. Книга должна быть полезна существующей и постоянно расширяющейся аудитории, включая тех, кого кроме традиционных областей автоматизации, обработки изображений, получения медицинских изображений, бесконтактных измерений и компьютерной картографии интересуют такие области, как мультимедиа, искусство и дизайн, геоинформационные системы и базы данных изображений. На первый взгляд, настолько общих целей невозможно достичь. Однако подобные учебники уже существуют в других областях, например по физике, математике и компьютерным наукам. Мы надеемся, что сделали, по крайней мере, хорошую книгу для начинающих — мы хотели написать книгу,

которая была бы полезна и на аудиторных занятиях, и самостоятельному читателю. Мы считаем, что выбранные темы будут интересными и иногда увлекательными, и, надеемся, доступными для большой аудитории. Предполагается, что в случае применения книги для обучения дипломированных специалистов или аспирантов в учебном курсе по компьютерному зрению также будут использованы статьи из списка дополнительной литературы. Этот список не планировалось сделать всесторонним; в конце каждой главы приведены ссылки на довольно небольшой набор статей. Изложение материала в первых главах начинается на интуитивном уровне и затем происходит переход к использованию математических моделей. Это сделано с целью формирования интуитивного понимания до знакомства с формальным описанием. Разделы, отмеченные звездочкой (\*), являются более сложными и математически насыщенными. В учебном курсе, не затрагивающем технических деталей, эти разделы можно не рассматривать. Для усиления интуитивного подхода в первых одиннадцати главах рассматривается обработка плоских изображений, а трехмерное компьютерное зрение оставлено для более поздних глав. Опытные преподаватели без труда смогут перераспределить материал в расчете на конкретный учебный курс или стиль обучения. Существует большое количество полностью двумерных приложений. В двумерной форме оказывается проще изучать многие понятия и алгоритмы. В гл. 4 рассматривается ряд основных понятий, связанных с распознаванием образов. Таким образом студенты смогут получить представление о законченных системах распознавания до полного изучения характерных признаков изображений и методов их нахождения. После изучения гл. 4 читатель получит хорошее представление о приложениях обработки двумерных изображений. В гл. 5, 6 и 7 вводятся характерные признаки полутоновых и цветных изображений, а также текстурные признаки. В гл. 8 обсуждается популярная недавно возникшая прикладная область — базы данных изображений. Некоторые коллеги советовали нам поместить этот материал в конце книги, но мы расположили его раньше, чтобы закрепить усвоение понятий предшествующих глав и представить материал, который может пригодиться для полусеместровых проектов. Сегментация и распознавание образов на двумерных изображениях рассматриваются в гл. 10 и 11. Материал в них представлен в наиболее простой форме, без учета сложностей, связанных с геометрическими преобразованиями в трехмерном пространстве. Свойства трехмерного пространства кратко представлены в гл. 2. Намного более подробно они изучаются в гл. 12. В гл. 12 качественно рассматриваются многие аспекты восприятия трехмерного мира посредством двумерных изображений. Эта глава заканчивается описанием модели стереоскопической зрительной системы и нескольких примеров применения уравнения тонкой линзы. Переход к трехмерному компьютерному зрению выполняется в гл. 13. На основе собственного преподавательского опыта авторы обнаружили, что на данном этапе сложность материала для студентов резко возрастает. Матрицы для представления геометрических преобразований в однородных координатах используются непосредственно в материале главы, а не выносятся в при-

ложение. Трехмерные версии этих преобразований являются расширениями более простых двумерных преобразований, уже рассматривавшихся в гл. 11. Аппроксимация методом наименьших квадратов, представленная в контексте двумерных задач в гл. 11, в гл. 13 также расширяется на трехмерный случай. Нелинейная оптимизация сначала рассматривается применительно к простой «перспективной задаче 3 точек». Затем нелинейная оптимизация применяется в задаче калибровки камеры с учетом радиальной дисторсии объектива. В гл. 14 описываются трехмерные модели и их распознавание по данным дальнометрических измерений. Глава 15 посвящена обсуждению приложений виртуальной и дополненной реальности и роли в них методов компьютерного зрения.

### **Замечания относительно языка программирования**

Книга не ориентирована на какой-либо конкретный язык программирования, но в ней используется некоторая обобщенная система обозначений для записи алгоритмов. Выбирать определенный язык не было необходимо. Конкретный язык для многих читателей мог бы оказаться не самым подходящим. Студенты, знакомые с программированием, при реализации алгоритмов не должны встретить значительных проблем, что и продемонстрировали наши собственные студенты. Примеры реализации в конечном счете будут представлены через Интернет. Это будет сделано, когда примеры реализации будут доступными и подходящими, во-первых, чтобы студенты могли быстро экспериментировать с ними, и, во-вторых, чтобы они могли изучать примеры исходных текстов программ. Для преподавателей и студентов доступны ряд инструментальных средств и библиотек; например, Khoros, NIH-Image, XView, gimp, MATLAB, и т. д. Существуют также программные пакеты, которые можно приобрести у компаний-производителей аппаратного обеспечения для систем машинного зрения. Авторы решили не ориентировать материалы книги на какое-либо определенное программное обеспечение по двум причинам. Во-первых, большинство читателей использовали бы какие-то другие пакеты. Во-вторых, восприятие книги было бы затруднено, если бы сущность операций по обработке изображений была скрыта за рассмотрением сложного каркаса структур данных и методов, необходимых в промышленных прикладных системах. Читатель, сначала изучивший принципы в простой программной среде, будет лучше подготовлен к успешному выбору и использованию промышленных систем.

### **Варианты работы с книгой**

Материал книги может быть различными способами отобран и при необходимости переупорядочен, применительно к целям конкретного учебного курса и с учетом интересов преподавателя и студентов.

- Глава 3 и краткий обзор гл. 2. Минимальным вариантом использования книги могло бы быть добавление 1–3 лекций в курс по структурам данных

и алгоритмам. Гл. 3, с учетом некоторых предварительных сведений из гл. 2, содержит ряд показательных приложений и упражнений на программирование с использованием двумерных массивов, поиска «в глубину» и структур данных для представления множеств.

- Главы 1, 2 и 3 и избранные фрагменты гл. 4, 5 и 6. В данном случае книга может служить для дополнения учебных курсов колледжей и младших курсов университетов материалом, рассчитанным на срок от 1 до 3 недель. Цель использования книги может заключаться в написании курсовой работы или быть более сложной, скажем, как групповой проект по разработке какой-либо программы (например, двумерной системы распознавания деталей на основе анализа связанных компонент маркированных изображений, или системы распознавания моделей на основе сопоставления векторов характерных признаков).
- Большая часть материала гл. 1–11. Например, в таком варианте может быть сделан обзор материала по обработке двумерных изображений в рамках курса по выбору для студентов, специализирующихся в географии, природных ресурсах или микробиологии (при условии, что будут пропущены большинство необязательных разделов). Если будут рассмотрены большинство разделов гл. 1–11, то такой объем мог бы соответствовать семестровому курсу по обработке и анализу изображений с введением в область компьютерного зрения.
- Большая часть книги. Подобный объем позволяет построить семестровый курс по компьютерному зрению для студентов старших курсов или аспирантов первого года обучения. В книге существенно больше материала, чем можно успеть подробно рассмотреть в течение одного семестра. Некоторые разделы придется проигнорировать или рассмотреть кратко, и не следует полагать, что читатель сможет решать домашние задания из всех разделов книги. При обучении студентов по учебным планам, разбитым на четверти, гл. 1–4, 6–12, и 14 обеспечат хорошее введение в область компьютерного зрения. В учебном курсе для аспирантов в течение одной четверти можно предложить минимальное рассмотрение гл. 1–4, затем акцентированное рассмотрение гл. 6–14 и краткий обзор гл. 15. Предполагается, что в любом курсе для аспирантов в процессе обучения будут использоваться статьи из дополнительной литературы.

Мы благодарны многим нашим коллегам, преподавателям, и студентам, с которыми нас объединяют общие интересы. Многие великодушно поддержали эту книгу вкладом идей, иллюстраций и алгоритмов. Отдельные цитаты приведены в этой книге повсюду. К сожалению, много предоставленного материала мы не смогли включить в книгу — иначе она могла бы стать слишком большой. Нашу работу существенно улучшили несколько рецензентов и много коллег, с которыми мы общались в процессе написания книги. В особенности мы благодарны тем, кто внимательно редактировал книгу. Это Mohammad Ghavamzadeh, Nick Dutta, Kevin Bowyer, Adam Clark, Yu-Yu Chou, Habib



Abi-Rached, Valentin Razmov. За любые оставшиеся в книге ошибки и за их исправление в будущем ответственность несут авторы.

Подготовка этой книги продолжалась четыре года. Нам помогали Paul Becker из Addison Wesley-Longman, Tom Robbins из Prentice Hall, Rose Rummel-Eury и Chanda Wakefield из ICC, а также Cathy Davison и Lorraine Evans. Создание этой книги было непростой работой, и, конечно, нам помогла команда профессионалов, обладающих как опытом, так и чувством юмора.

Линда Шапиро  
shapiro@cs.washington.edu  
Джордж Стокман  
stockman@cse.msu.edu

# Введение

Эта книга служит введением в широкий круг вопросов компьютерного зрения. Без сомнения, можно сконструировать машины для обработки зрительных данных: например, на фабриках машины ежедневно контролируют миллионы осветительных ламп накаливания. Разработаны системы безопасности, сканирующие человеческий глаз для идентификации личности. Известны эксперименты по управлению автомобилем на основе компьютерной обработки данных, поступающих с видеокамеры. В данной главе представлен ряд важных прикладных областей, в которых возможно применение систем компьютерного зрения. После прочтения этой главы у читателя сложится общее представление о некоторых задачах и методах компьютерного зрения.<sup>1 2</sup>

**1 Определение.** Цель компьютерного зрения заключается в формировании полезных выводов относительно объектов и сцен реального мира на основе анализа изображений, полученных с помощью датчиков.

Для формирования выводов относительно объектов реального мира почти всегда необходимо построить некоторое описание или модель этих объектов на основе изображения. Поэтому многие эксперты заключают, что *цель компьютерного зрения состоит в формировании описаний сцен по изображениям*. Хотя наш учебник по компьютерному зрению имеет прикладной характер, в нем также обсуждается ряд фундаментальных вопросов. Перечислим важнейшие темы, которые будут обозначены в данной главе и затем будут изучаться в оставшейся части книги.

**Восприятие:** Как с помощью датчиков можно получить изображения сцен реального мира? Каким образом в изображениях представлены свойства

---

<sup>1</sup> В данной книге термины *машинное зрение* и *компьютерное зрение* обычно считаются синонимами. Но во многих других книгах и статьях они отличаются — машинное зрение употребляется в контексте промышленных приложений, а компьютерное зрение обозначает прикладную область обработки зрительных данных в целом.

<sup>2</sup> В русскоязычной литературе в качестве синонима *машинного зрения* часто используется термин *техническое зрение*. — *Прим. перев.*



	0	1	2	3	4	5	6	7
0	130	146	133	95	71	71	62	78
1	130	146	133	92	62	71	62	71
2	139	146	146	120	62	55	55	55
3	139	139	139	146	117	112	117	110
4	139	139	139	139	139	139	139	139
5	146	142	139	139	139	143	125	139
6	156	159	159	159	159	146	159	159
7	168	159	156	159	159	159	139	159

**Рис. 1.1.** (Слева сверху) изображение лица, (в центре) фрагмент изображения размером  $8 \times 8$  пикселей из окрестности правого глаза, (справа сверху) местоположение глаз, определенное компьютерной программой, и (внизу) значения яркости во фрагменте  $8 \times 8$  пикселей. (С любезного разрешения Vega Bakic.)

реальных объектов, например материал, форма, освещенность и пространственные отношения?

**Кодирование информации:** Как изображения дают информацию для понимания трехмерного мира, в том числе информацию для идентификации объектов, об их геометрических и текстурных свойствах и движениях?

**Представление:** В какой форме можно хранить описания объектов, их частей, свойств и взаимосвязей?

**Алгоритмы:** Какие существуют методы для обработки информации изображений и для построения описаний реальных объектов и сцен?

В последующих главах будут изучаться эти и некоторые другие вопросы. Теперь кратко перечислим различные прикладные задачи и ряд важных проблем, возникающих в контексте этих задач.

## 1.1 МОГУТ ЛИ МАШИНЫ ВИДЕТЬ?

Авторы фантастической и научно-популярной литературы были восхищены идеей конструирования разумных машин, и способность визуального восприятия мира стала безоговорочным требованием, которое некоторые из них предъявляли к таким машинам. Зрение является очень важной функцией

человеческого мозга. Многие задачи зрительного восприятия люди решают бессознательно и без напряжения, но пока наше понимание этого процесса очень слабое. Алан Тьюринг, один из изобретателей современного цифрового компьютера и основателей области искусственного интеллекта, верил, что цифровой компьютер сможет обладать разумом и способностью понимания сцен. Эти высокие цели труднодостижимы, и богатство человеческого воображения пока далеко превосходит инженерные возможности. Однако по ряду направлений исследований был достигнут впечатляющий прогресс. Хотя основной темой данного учебника является разработка прикладных систем, а не вопросы искусственного интеллекта, но иногда мы будем рассматривать более глубокие вопросы и, где возможно, приводить некоторые оценки достигнутого прогресса. Обсудим, например, следующий сценарий, который может воплотиться через несколько лет. Телевизионная камера на вашей двери передает изображения на домашний компьютер, обученный распознавать лица важных для вас людей. Когда вы связываетесь со своим домашним центром сообщений, то компьютер не только докладывает о телефонных звонках, но также сообщает о возможных визитах вашей сестры или соседа. Подобные направления современных разработок и исследований будут обсуждаться в различных разделах книги.

## 1.2 ПРИКЛАДНЫЕ ЗАДАЧИ

Возможности применения компьютеров для обработки изображений очень велики. Здесь будут перечислены только несколько различных задач, но и они смогут заинтересовать в дальнейшем изучении предмета и помочь сориентироваться в данной области.

### 1.2.1 Предварительное рассмотрение структуры цифрового изображения

Цифровые изображения могут представлять различные образы, например рисунок, страницу текста, человеческое лицо, карту Катманду, или товар для покупки по каталогу. Цифровое изображение состоит из фиксированного количества строк и столбцов *пикселей* (pixels), этот термин является сокращением от слов «*элемент изображения*» (picture element). Пикселы напоминают маленькие плитки, в которых хранятся дискретные значения — небольшие числа, часто от 0 до 255, представляющие яркость точек изображения. В зависимости от схемы кодирования 0 может соответствовать самой малой (темной), а 255 — самой большой (светлой) яркости, или наоборот.

На рис. 1.1 слева вверху напечатано цифровое изображение лица. Изображение содержит 257 строк пикселей в высоту и 172 столбца в ширину. Вверху в центре показан фрагмент изображения размерами  $8 \times 8$  пикселей, взятый из окрестности правого глаза на левом изображении. В нижней части рис. 1.1 приведена таблица с 64 числами — значениями яркости пикселей фрагмента изображения. Числа меньше 100 в правом верхнем углу фрагмента пред-

ставляют слабо отражающую свет темную область глаза (зрачок), а большие значения соответствуют более светлому белому цвету глаза.

В цветных изображениях для каждого пиксела может храниться три числа, например одно число для красной, одно для синей и одно для зеленой составляющей цвета. Цифровые изображения чаще всего отображаются на мониторе, который можно представлять себе как телевизионный экран с цифровой памятью для изображения. Цветное изображение размерами 500 строк и 500 столбцов примерно соответствует изображениям, которые вы видите на экране своего телевизора. Пиксел высвечивается путем передачи энергии маленькой области люминисцентного материала. Воспроизведение цвета требует передачи энергии трем соседним областям, сделанным из различных материалов. Размеры компьютерного дисплея высокого разрешения составляют примерно  $1200 \times 1000$  пикселей.

Более подробно структура цифровых изображений обсуждается в гл. 2, а кодирование и интерпретация цвета рассматривается в гл. 6.

### 1.2.2 Поиск в базе данных изображений

Разработка баз данных изображений стала возможной благодаря доступности цифровой памяти большого объема, линий связи с большой пропускной способностью и персональных компьютеров с мультимедиа-возможностями. Для эффективного использования большого количества существующих изображений необходимо разработать удобные средства доступа. Обычные технологии баз данных можно применять для доступа к изображениям, снабженным текстовыми метками. Однако также необходимы средства поиска изображений *на основе содержания*, которые в настоящее время являются предметом интенсивных исследований.

Представим, что недавно организованная фирма хочет разработать и зарегистрировать эмблему. Дизайнер представил на рассмотрение компании несколько вариантов эмблем. Эмблему нельзя использовать, если она слишком похожа на эмблему какой-либо существующей компании, поэтому надо проверить базу данных существующих эмблем. Эта процедура аналогична патентному поиску и выполняется вручную, но могла бы быть существенно упрощена с помощью методов машинного зрения. На рис. 1.2 показано несколько похожих эмблем.

Еще один пример: допустим, архитектору или историку-искусствоведу требуется найти изображения зданий, у которых вход оформлен в определенном стиле. Для поиска они могут указать изображение-образец, возможно, полученное из той же базы данных, и запросить систему найти остальные похожие изображения.

В гл. 8 будет показано, как можно использовать геометрические, цветовые и текстурные признаки для обслуживания подобных запросов к базе данных изображений.

Предположим, что рекламное агентство хочет найти существующие изображения маленьких детей, которые что-то едят с радостным видом. Это



**Рис. 1.2.** Запрос изображения по образцу: изображение-образец для запроса (слева) и два наиболее похожих изображения, найденных системой поиска в базе данных изображений. (С любезного разрешения Graphic-sha, Токио.)

семантическое описание, легко понимаемое людьми, очень сложно представить для системы машинного зрения. Понятия «дети», «радостно» и «есть» могут потребовать сложного комбинированного учета цветовых, текстурных и геометрических признаков.

Попутно отметим, что был разработан компьютерный алгоритм для обнаружения обнаженных людей на цветных изображениях. Он может быть полезен для родителей, желающих проконтролировать изображения, которые их дети загружают из Интернета. Методы извлечения изображений из баз данных подробнее рассматриваются в гл. 8.

### 1.2.3 Контроль отверстий в поперечных балках

В конце 1970-х гг. инженер из Милуоки разработал систему машинного зрения для подсчета количества болтовых отверстий в поперечных балках, изготовленных для компаний-производителей грузовых автомобилей. Эти компании требовали контроля каждой балки, так как отсутствие болтового отверстия на частично собранном грузовике является дорогостоящим дефектом. Для его исправления или надо останавливать конвейер и сверлить отверстие, или, что хуже, рабочий может пропустить установку болта, чтобы не останавливать конвейер.

Для получения цифрового изображения поперечной балки под транспортировочной линией были помещены осветители, а над линией была расположена видеокамера. При поступлении в поле зрения очередной балки выполнялся съем изображения. Темные пиксели в области тени от балки на изображении имели значение 1 и обозначали сталь, а пиксели в области светлых просверленных отверстий имели значение 0. Количество отверстий вычислялось как четверть разности количества *внешних* и *внутренних углов*. На рис. 1.3 показаны три светлых отверстия (значения 0) на темном фоне (значения 1). *Внешний угол* (external corner) — это окрестность  $2 \times 2$  соседних пикселей, содержащая 3 единицы, а *внутренний угол* (internal corner) — окрестность  $2 \times 2$  соседних пикселей, содержащая 3 нулевых значения. На рис. 1.3 показан пример обработки изображения из 7 строк и 16 столбцов. Ниже также схематично описан алгоритм обработки.

Подсчет отверстий является примером простой, но практически полезной операции над цифровыми изображениями. (Как показано в приведенных да-

1	1	1	0	0	1	1	1
1	0	1	1	1	1	0	1

(a) Шаблоны окрестностей  $2 \times 2$  пикселей, представляющие «внешние углы»

0	0	0	1	1	0	0	0
0	1	0	0	0	0	1	0

(b) Шаблоны окрестностей  $2 \times 2$  пикселей, представляющие «внутренние углы»



(c) Три светлых отверстия на темном фоне

	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	e	i
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
1	1	0	0	0	1	1	1	1	1	0	0	1	1	0	0	1		
2	1	0	0	0	1	1	1	1	1	1	0	1	1	0	0	1		
3	1	1	1	1	1	0	0	1	1	1	0	0	1	1	0	1		
4	1	1	1	1	0	0	0	0	1	1	0	0	0	0	0	1		
5	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1		
6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		

(d) Входное бинарное изображение с размерами 7 строк и 16 столбцов

	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	e	i
0	e			e					e		e		e		e		6	0
1									e	i							1	1
2	e			e	e		e				i	e	e	i			6	2
3				e	i		i	e				i		i			2	4
4				e	i		i	e		e					e		4	2
5					e		e										2	0
6																	0	0

(e) Символами 'e' и 'i' помечены окрестности, соответствующие внешним и внутренним углам

**Рис. 1.3.** Подсчет отверстий на бинарном изображении: 21 внешний угол (e) минус 9 внутренних углов (i) разделить на 4 равно количеству отверстий — 3. Верно ли это?

**Входные данные:** бинарное изображение

**Выходные данные:** количество найденных отверстий

Обозначения:

**M** бинарное изображение из **R** строк и **C** столбцов.

Значения '1' представляют материал, не пропускающий свет;

Значения '0' обозначают отсутствие материала, воспринимаемое за счет прохождения света при освещении объекта «на просвет».

Каждая область из '00' должна быть 4-связной и все граничные пиксели изображения должны быть равны '1'.

**E** — счетчик *внешних углов* (3 единицы и 1 ноль в окрестности  $2 \times 2$ )

**I** — счетчик *внутренних углов* (3 нуля и 1 единица в окрестности  $2 \times 2$ )

```
integer procedure Count_Holes(M)
{
  обработать все изображение по 2 строки на каждом шаге;
  подсчитать количество внешних углов E;
  подсчитать количество внутренних углов I;
  вернуть в качестве результата (number_of_holes = (E - I)/4);
}
```

**Алгоритм 1.1.** Каркас алгоритма подсчета отверстий на бинарном изображении

лее упражнениях, алгоритм подсчета отверстий корректно работает только с «4-связными» и «просто связными» отверстиями — т. е. внутри отверстий не должно быть фоновых пикселей. Эти понятия будут рассматриваться в гл. 3, а более детальное описание можно найти в книге Розенфельда.)

**Упражнение 1.1.** Подсчет количества отверстий

Даны три изображения с размерами  $4 \times 5$ ,  $4 \times 4$  и  $4 \times 5$ :

1	1	1	1	1
1	0	1	0	1
1	0	1	0	1
1	1	1	1	1

1	1	1	1
1	1	0	1
1	0	1	1
1	1	1	1

1	1	1	1	1
1	0	1	0	1
1	0	0	0	1
1	1	1	1	1

При поиске угловых шаблонов на трех вышеприведенных изображениях алгоритм 1.1 проверит 12, 9 и 12 окрестностей  $2 \times 2$ . Каждая из окрестностей  $2 \times 2$  считается окрестностью типа 'e', 'i' или 'n' — внешним углом, внутренним углом или «пустой». (а) Сколько окрестностей  $2 \times 2$  каждого типа содержится на каждом из трех данных изображений? (б) Работает ли формула подсчета отверстий для всех трех изображений?



### 1.2.4 Контроль медицинских изображений головы человека

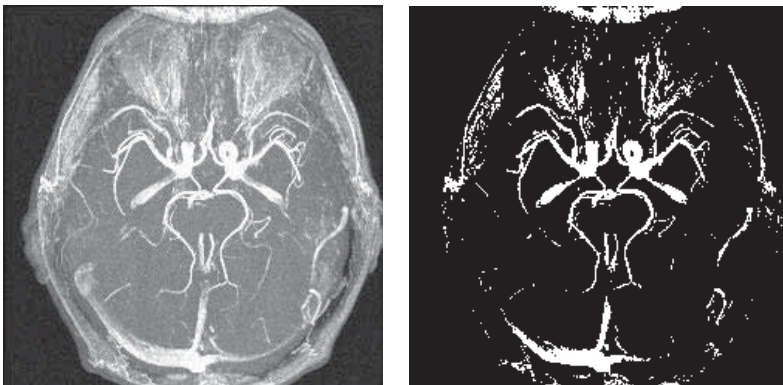
Датчики изображений на эффекте ядерного магнитного резонанса (ЯМР) могут воспринимать вещества внутри трехмерных объектов. На рис. 1.4 показано поперечное сечение человеческой головы: яркие области соответствуют движущемуся веществу, так что в действительности это изображение потока крови. Специалист может узнать на изображении важные кровеносные сосуды.

Нечетко видные кометоподобные структуры связаны с глазами. ЯМР-изображения используются в медицине для обнаружения опухолей и нарушений кровообращения, таких, как аномальное сужение или расширение сосудов. На рис. 1.4 справа приведена копия рис. слева, на которой все пиксели с яркостью 208 или ярче сделаны максимально яркими (255), а с яркостью менее 208 — темными (0). Большинство пикселей корректно показывают кровеносные сосуды на фоне, но также есть некорректно «окрашенные» пиксели обоих типов. Методы машинного зрения часто используются при анализе медицинских изображений, хотя обычно применяются для визуализации информации и данных измерений, а не для формирования диагноза. Конечно, было бы очень хорошо, если бы мы могли «увидеть» мыслительные процессы, происходящие в мозге! Оказывается, метод ЯМР может чувствовать органическую активность, связанную с мыслительными процессами, и это очень перспективная область современных исследований.

---

#### Упражнение 1.2. Сколько пикселей приходится на одно отверстие?

Более детально рассмотрите задачу подсчета отверстий на поперечных балках грузовиков. Предположим, что на изображении видна область балки длиной 120 см и шириной 25 см. Будем считать, что эта область почти заполняет



**Рис. 1.4.** ЯМР-изображение (слева), на котором яркость связана с потоком крови, и бинарное изображение (справа), полученное в результате замены всех пикселей со значениями 208 или больше на 255 и всех пикселей со значениями менее 208 на 0. С любезного разрешения James Siebert, Michigan State Radiology.

цифровое изображение из 100 строк и 500 столбцов пикселей. Допустим, что диаметр одного болтового отверстия в балке равен 1.2 см. Определите радиус и площадь этого отверстия на изображении в пикселях.

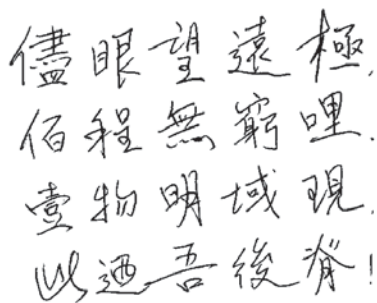
### Упражнение 1.3. Отображение монеты на пиксели

Эта задача по смыслу связана с предыдущей. Возьмите клетчатую бумагу (достаточно клеток со сторонами 0,5 см) и металлическую монету. Случайным образом пять раз подбросьте монету на бумагу и каждый раз обводите ее окружность. Для каждого из пяти положений оцените площадь изображения монеты в единицах пикселей, (а) решая, является ли пиксел частью монеты или нет (без дробных чисел), и (б) для каждого пикселя, по которому проходит окружность, оценивайте с точностью до десятой пикселя, какая его часть является частью изображения монеты. После выполнения этих измерений вычислите среднее значение и среднеквадратическое отклонение площади изображения отдельно для случаев (а) и (б).

### 1.2.5 Обработка сканированных страниц текста

Преобразование информации, содержащейся в бумажных документах, в цифровую форму для информационных систем является очень распространенной задачей. Например, надо предоставить доступ к старой книге через Интернет, или требуется преобразовать бумажный чертеж некоторой детали в файл с данными о его геометрии для последующего изготовления этой детали на станке с программным управлением.

На рис. 1.5 показано написание одной и той же фразы на китайском и английском языках. Китайские иероглифы были написаны на бумаге и отсканированы как изображение из 482 строк и 405 столбцов. Файл на языке Postscript с описанием графического изображения (и напечатанный на рис. 1.5) занял



儘眼望遠極  
佰程無窮哩  
壹物明域現  
此迺吾後脊!

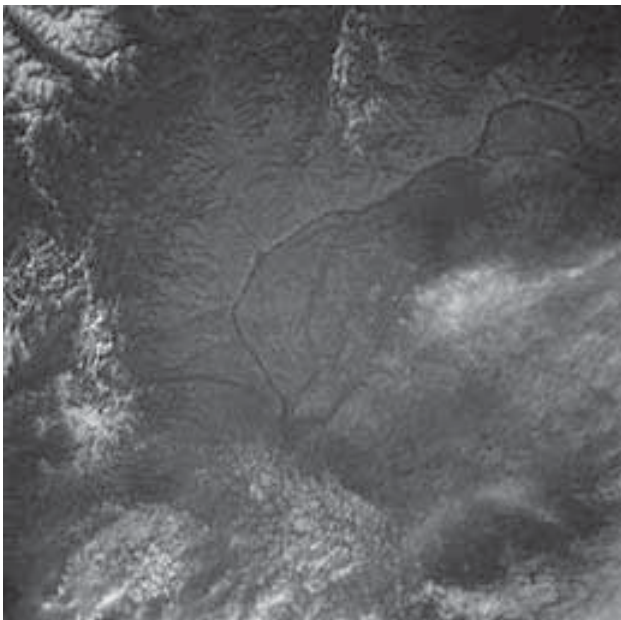
I looked as hard as I could see,  
beyond 100 plus infinity  
an object of bright intensity  
—it was the back of me!

**Рис. 1.5.** (Слева) китайские иероглифы и (справа) эквивалентный английский текст. Может ли машина автоматически перевести его? (С любезного разрешения John Weng и Stockman.)

68464 байта. Английская версия занимает в файле 115 байт, по одному байту на каждый символ таблицы ASCII. Применительно к этому случаю можно перечислить целый набор важных задач обработки документов. Распознавание отдельных символов по точкам со сканированного или факс-изображения в настоящее время реализовано на уровне коммерческих продуктов, при условии, что символы имеют стандартное начертание. Семантическая интерпретация информации, возможно, необходимая для индексации в большой базе данных, представляет собой намного более сложную задачу.

### 1.2.6 Оценка снежного покрова по спутниковым изображениям

Большая часть земной поверхности регулярно сканируется с помощью спутников, передающих на Землю цифровые изображения. Последующая обработка этих изображений позволяет получить много полезной информации. Например, оценка количества снега в водоразделе реки может быть критически важна для управления плотинами, системами водоснабжения, для охраны окружающей среды. Для оценки массы снега можно подсчитать количество пикселей на изображении, которые выглядят, как снег. Пиксел спутникового изображения может представлять довольно крупный участок поверхности, например  $10 \times 10$  метров, но у некоторых спутников разрешение намного выше.



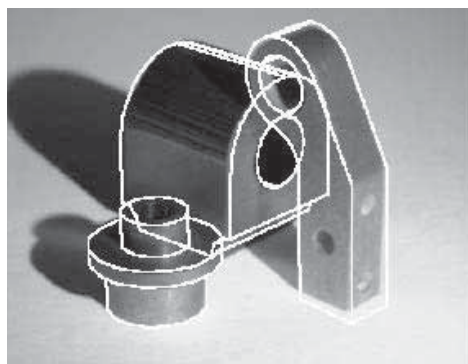
**Рис. 1.6.** Фотография рек Уэнатчи и Колумбия (штат Вашингтон). (С любезного разрешения Космического центра им. Л. Джонсона, см. <http://earthrise.sdsu.edu>.)

Часто спутниковое изображение требуется сравнить с картой или другими изображениями и определить, какие пиксели попадают в определенную область или водную поверхность. Это сравнение обычно проводится вручную с помощью специальных программ обработки изображений (более подробно сопоставление изображений описано в гл. 11). На рис. 1.6 показана фотография, сделанная во время полета космического корабля «Спейс-Шаттл». На ней снят город Уэнатчи, штат Вашингтон, около которого река Уэнатчи впадает в реку Колумбия.

Возможности компьютеров по обработке больших объемов данных хорошо известны. Спутниковая фотосъемка земной поверхности дает огромное количество данных, полезных для многих целей. Например, количество и местоположение снега могут быть входными данными для компьютерной программы, моделирующей гидросферу данного региона (также ей требуется информация о температуре). Другим возможным приложением является оценка посевов и предсказание урожая. Еще одна задача — учет городских зданий для налогообложения: это обычно делается вручную по изображениям, полученным при аэрофотосъемке.

### 1.2.7 Анализ сцен, содержащих детали промышленного производства

Во многих производственных процессах есть этапы, на которых отдельные детали перемещаются на конвейере или в упаковке. Детали требуется по отдельности помещать в механизмы, в упаковку, контролировать и т. п. Если подобная операция утомительна или опасна, то может быть целесообразно применение роботизированного комплекса с системой машинного зрения. На рис. 1.7 показано изображение с тремя деталями, расположенными в рабочем пространстве робота. Распознавая ребра и отверстия, зрительная система робо-



**Рис. 1.7.** В задаче инспекционного контроля или управления сборочным роботом зрительная система сравнивает трехмерные модели объектов с входным двумерным изображением. (С любезного разрешения Mauro Costa.)

та способна решить, какие именно детали видны и каково их местоположение и ориентация в рабочем пространстве. В процессе распознавания каждой детали используется трехмерная модель, предварительно сформированная в системе автоматизированного проектирования (САПР). Зрительная система сравнивает входное изображение с компьютерными изображениями, сгенерированными по данным моделей и данным о положении объекта в пространстве. Плохо совпадающие предположения отбрасываются, а правдоподобные уточняются. Яркими линиями на рис. 1.7 показаны три уточненных соответствия между изображением и моделями объектов, находящимися на этом изображении. После распознавания система управления роботом может сообщить манипулятору, как надо захватить деталь и куда ее поместить. Задачи и методы трехмерного компьютерного зрения в данной книге рассматриваются в гл. 13 и 14.

---

#### **Упражнение 1.4.** Другие типы задач

Приведите пример задачи, отличающейся от рассмотренных в тексте, в которой возможно применение компьютерного зрения. Какого типа сцены будут наблюдаться на изображениях? Что необходимо получать в качестве выходных данных?

---

#### **Упражнение 1.5.** Анализ специфики задачи

Прикладные задачи могут быть решены различными способами, и разработчик не должен слишком рано попадать в ловушку специфического подхода. Рассмотрите различные ситуации, в которых возникает задача идентификации автомобилей: (а) въезд на место парковки или в охраняемую область, (б) прохождение через ворота при въезде на платную дорогу, (в) превышение скорости. Несколько групп разрабатывают или уже разработали методы машинного зрения для считывания автомобильных номерных знаков. Предложите альтернативу системам машинного зрения. Каковы экономические и социальные затраты в сравнении с подходом машинного зрения?

### **1.3 ОПЕРАЦИИ ОБРАБОТКИ ИЗОБРАЖЕНИЙ**

В этой книге рассмотрено много различных операций обработки изображений. Их можно разделить на несколько категорий в зависимости от структуры, уровня или назначения. Некоторые операции предназначены исключительно для повышения качества изображений при человеческом восприятии, в то время как остальные предназначены для извлечения информации, пригодной для автоматической обработки. Некоторые операции создают новые выходные изображения, а некоторые в качестве выходных данных генерируют описания неграфической формы. Ниже перечислены несколько важных категорий операций обработки изображений.

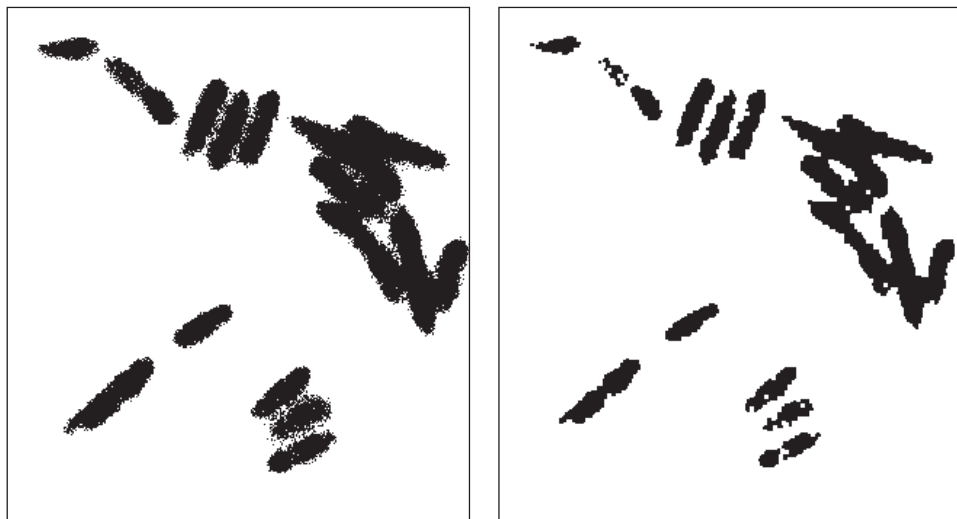
### 1.3.1 Модификация пикселей в малых окрестностях

Значения пикселей можно изменять с учетом их взаимосвязи с небольшим числом близлежащих пикселей, например, пикселей из соседних строк или столбцов. Часто на бинарных изображениях изолированные значения 1 и 0 меняются на противоположные значения, чтобы они совпадали со своими соседями. Назначением этой операции может быть удаление шума, появившегося в процессе оцифровки. Или, это делается для упрощения изображения; например, для игнорирования мелких островов на изображении озера или дефектов на изображении листа бумаги. Еще одна распространенная операция — изменение значений *граничных пикселей* на значения *фоновых пикселей* (рис. 1.8). Изображения отдельных бактерий на рис. 1.8 имеют нечеткие границы и часто сливаются вместе. После замены черных граничных пикселей на белые, изображения бактерий хотя и становятся меньше, но получают более явные границы. Некоторые ранее сливавшиеся пары разделяются. Подобные операции будут рассмотрены в гл. 3.

---

#### Упражнение 1.6

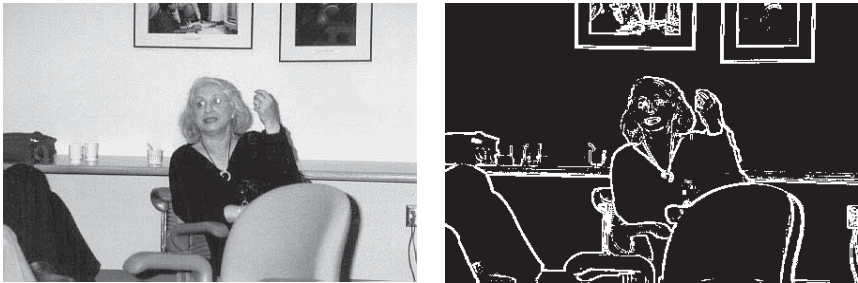
Укажите некоторые дефекты, оставшиеся на правом изображении на рис. 1.8. Опишите простые операции в окрестности пикселей для улучшения этого изображения.



**Рис. 1.8.** (*Слева*) бинарное изображение бактерии (на оригинальном микроскопическом изображении бактерия была синей из-за флуоресценции); (*справа*) очищенное изображение, полученное путем удаления черных пикселей, у которых есть один или несколько белых соседей. (С любезного разрешения Frank Dazzo.)

### 1.3.2 Глобальное улучшение качества изображения

Некоторые операции выполняют однородную обработку целого изображения. Изображение может быть слишком темным, например, максимальная яркость равна 120, так что все значения яркости можно увеличить в два раза для улучшения вида изображения при выводе на экран. Для удаления шума или несущественных деталей можно заменить значение каждого входного пиксела средним значением 9 пикселей его окрестности. С другой стороны, детали можно подчеркнуть, если заменить каждый пиксел значением контраста между ним и его соседями. На рис. 1.9 показан результат вычисления контраста для всех пикселей входного изображения. Отметим, что границы большинства объектов оказались хорошо выделенными. Выходное изображение получено вычислением контраста в окрестностях размером  $3 \times 3$  на входном изображении. В гл. 5 описано несколько типов подобных операций. Возможно, изображение было получено с помощью широкоугольного объектива («рыбий глаз») и требуется получить выходное изображение без искажений объектива: в данном случае можно «переместить» значения пикселей в другие положения на изображении. Такая операция называется *деформацией изображения* (см. гл. 11).

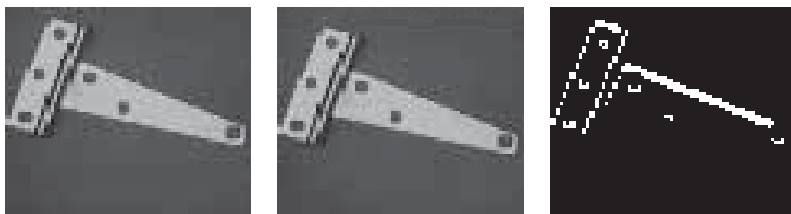


**Рис. 1.9.** Данные о контрасте левого изображения представлены в виде изображения справа. 10% наибольших значений пикселей (в единицах контраста) показаны яркими, а остальные 90% — темными. Контраст был вычислен для каждого пиксела по окрестности  $3 \times 3$ .

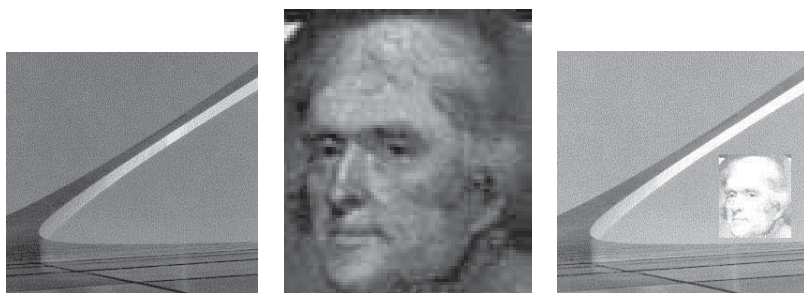
### 1.3.3 Комбинация нескольких изображений

Изображение можно сформировать путем сложения или вычитания двух исходных изображений. Вычитание изображений часто применяется для обнаружения изменений в течение некоторого промежутка времени. На рис. 1.10 показаны два изображения движущейся детали и разностное изображение, полученное путем вычитания пикселей второго изображения из соответствующих пикселей первого изображения. Вычитание изображений сохраняет границу движущегося объекта, хотя и не идеальным образом. (Так как отрицательные





**Рис. 1.10.** Изображения движущейся детали (*слева и в центре*) и разностное изображение, на котором заметна граница детали.



**Рис. 1.11.** Изображение большой арки в Сент-Луисе (*слева*), портрет Джефферсона (*в центре*) и комбинация двух изображений (*справа*).

значения пикселов не используются, то в выходном изображении представлены не все изменения.) В другом примере новые городские постройки могут быть выявлены путем вычитания изображения с аэрофотосъемкой города пять лет назад из изображения, полученного недавно.

Сложение изображений также является полезной операцией. На рис. 1.11 показано изображение Томаса Джефферсона, «добавленное» к изображению арки в Сент-Луисе<sup>1</sup>. В данном случае чуть больше усилий требуется для лучшего *смешения* изображений.

### 1.3.4 Вычисление характерных признаков изображения

Выше была рассмотрена задача подсчета отверстий. В общем случае, области значений 0 или 1, которые могут быть изображениями объектов (например, это отверстия в поперечной балке или микробы в образце воды), часто называются *пятнами*. Важными характерными признаками являются средняя площадь объектов, периметр, направление и др. Подобные признаки описываются в гл. 3. В главах 6 и 7 обсуждаются способы количественного учета цвета или текстуры во фрагментах изображений. В гл. 4 показано, как по этим

<sup>1</sup> Эта арка была возведена в честь Луизианской Покупки — приобретения у Франции штата Луизиана (США). — *Прим. перев.*



Объект	Площадь	Описывающий прямоугольник	Центр тяжести
1	247	[(20 , 26) , (32 , 56)]	(26.1 , 42.0)
2	6	[(25 , 22) , (26 , 24)]	(25.5 , 23.0)
3	116	[(35 , 72) , (54 , 86)]	(44.1 , 79.4)
4	4	[(37 , 69) , (38 , 70)]	(37.5 , 69.5)
5	15	[(46 , 86) , (50 , 89)]	(47.6 , 87.4)
6	586	[(49 , 122) , (95 , 148)]	(71.7 , 134.8)
7	300	[(54 , 91) , (77 , 112)]	(65.6 , 101.9)
8	592	[(57 , 138) , (108 , 163)]	(83.6 , 150.8)
9	562	[(57 , 158) , (104 , 183)]	(81.2 , 171.4)
10	5946	[(74 , 195) , (221 , 313)]	(138.0 , 256.5)
11	427	[(204 , 115) , (229 , 151)]	(217.1 , 132.3)
12	797	[(242 , 42) , (286 , 97)]	(264.9 , 71.8)
13	450	[(248 , 170) , (278 , 204)]	(262.7 , 188.1)
14	327	[(270 , 182) , (291 , 216)]	(279.9 , 200.3)
15	264	[(293 , 195) , (311 , 221)]	(300.8 , 206.7)
16	145	[(304 , 179) , (316 , 193)]	(310.4 , 186.4)

Суммарная площадь объектов равна 10784 пикселей

**Рис. 1.12.** Связные компоненты (области изображения), автоматически обнаруженные на изображении бактерий на рис. 1.8 (справа). Отдельная бактерия характеризуется площадью порядка нескольких сотен пикселей. Большая компонента состоит из нескольких соприкасающихся бактерий, а мелкие объекты 2, 4 и 5 появились из-за шума.

признакам выполняется классификация объектов; например, решается, является ли обнаруженная на изображении область микробом вида А или В? На рис. 1.12 показаны выходные данные известного алгоритма, примененного к изображению бактерий на рис. 1.8, который вычислил характерные признаки отдельных областей, найденных на изображении, в том числе площадь и местоположение каждой области. Области с площадью несколько сотен пикселей соответствуют отдельным бактериям, а большая область соответствует нескольким соприкасающимся бактериям.

### 1.3.5 Формирование неграфических описаний

Высокоуровневые операции обычно формируют неграфические описания изображений, т. е. структуры данных, не являющиеся изображениями. (Напомним, что формирование таких описаний часто считается основной целью компьютерного зрения.) На рис. 1.12 показано неграфическое описание изображения бактерий. Кроме уже описанных примеров, можно назвать отчет о подсчете микробов видов А и В на слайде с микроскопа или об объеме транспортного потока между двумя частями города, вычисленного по видеоданным. В другой важной задаче графическое изображение может быть сканированной журнальной статьей, а выходным представлением — гипертекстовая структура, содержащая части распознанного текста и рисунки из статьи. Еще один пример (рис. 1.7) — система машинного зрения, которая должна в качестве выходных данных сформировать набор из трех распознанных объектов, для каждого

указать идентификационный код, три параметра положения и три параметра ориентации. Это описание сцены, в свою очередь, может быть передано в подсистему планирования движения для осуществления взаимодействия робота-манипулятора с этими тремя деталями.

## 1.4 УСПЕХИ, СЛОЖНОСТИ И НЕУДАЧИ КОМПЬЮТЕРНОГО ЗРЕНИЯ

При рассмотрении прикладных задач машинного зрения мы не можем обойтись без констатации того факта, что успех в них обычно достигается дорогой ценой. Часто разработчики предполагают наличие ряда ограничений в окружающей среде, что заставляет идти на компромиссы с требованиями к гибкости системы. Например, может потребоваться управляемое освещение сцены или механическая изоляция и позиционирование объектов перед вводом изображения. Реальный мир демонстрирует чрезвычайную изменчивость входных изображений, поэтому необходимо разрабатывать все более совершенные компьютерные алгоритмы для вычисления «сущности», или, *инвариантных признаков* объектов. Вид объекта на изображении может значительно меняться при изменениях освещенности или в присутствии других объектов, которые система может даже не ожидать. Например, это тени на рис. 1.7 и рис. 1.9.

Выводы относительно структуры объекта часто должны делаться путем учета различной информации от большого количества пикселей изображения. Например, яркость на верхней части стаканов на рис. 1.9 такая же, как и у стены, поэтому на уровне пикселей граница стекло-стена неочевидна. Для распознавания каждого стакана как отдельного объекта потребуется сгруппировать и совместно обрабатывать пиксели из некоторой довольно большой области. Люди делают это очень хорошо, но разработка гибких методов группировки для машинного зрения является очень сложным делом. Проблемы загораживания объектов серьезно затрудняют распознавание трехмерных объектов. Может ли зрительная система распознать человека или кресло на рис. 1.9, даже если у них не видно ног? Даже на высоком уровне какая модель *собаки* могла бы позволить машине распознать различных собак, которые могут оказаться на изображениях? Эти и другие трудности будут обсуждаться далее на протяжении всей книги.

---

### Упражнение 1.7

Какие инвариантные признаки следующих объектов позволят вам распознавать их в дождливую или солнечную погоду, по отдельности или в присутствии других объектов, спереди или сбоку: (а) теннисные туфли, (б) входная дверь, (в) автомобиль вашей любимой марки?

## 1.5 ПРИМЕНЕНИЕ КОМПЬЮТЕРОВ И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Широко известны возможности компьютеров в области точной обработки количественной информации. Вычислительные методы обработки изображений развиваются уже более 30 лет — сначала преимущественно в исследовательских лабораториях на больших ЭВМ или на производственных предприятиях на компьютерах специального назначения. В последнее время недорогая память и высокая скорость процессоров общего назначения дали возможность работы с изображениями практически на любом персональном мультимедиа-компьютере, и даже любители в домашних условиях могут экспериментировать в этой области.

Вычислительную обработку изображений можно выполнять различными способами. Проще всего взять готовую программу, в которой реализованы большинство необходимых операций обработки. Некоторые подобные программы общедоступны, другие являются коммерческими продуктами. В Интернете можно получить большое количество свободно распространяемых изображений. Для обеспечения ввода собственных изображений вы можете приобрести планшетный сканер или цифровую камеру, эти устройства стоят порядка нескольких сотен долларов. Существуют программные библиотеки, содержащие множество подпрограмм для обработки изображений: пользователь может написать свою программу, которая вызывает библиотечные подпрограммы для выполнения нужных операций над данными изображений пользователя.

Большинство компаний, которые продают устройства ввода для систем машинного зрения, также поставляют библиотеки обработки изображений и даже оболочки для них, имеющие удобный графический пользовательский интерфейс. Для ускорения операций обработки, которые на процессоре общего назначения могут требовать многих секунд или даже минут, доступно специальное аппаратное обеспечение. Многие первые параллельные компьютеры стоимостью в миллионы долларов были разработаны в расчете на задачи обработки изображений; однако сегодня большинство основных операций могут быть выполнены с помощью набора плат ценой несколько тысяч долларов. Обычно специальная аппаратура нужна только для высокоскоростной обработки изображений и в системах реального времени.

Были разработаны специальные языки программирования, в которых операции по обработке изображений были оформлены в виде элементов языка. Иногда эти языки содержали также операции для управления промышленными роботами. Сегодня стало очевидно, что хорошие системы обработки изображений можно и нужно писать на языке программирования общего назначения, например на Си, а компьютер общего назначения можно купить по почте или во многих обычных магазинах. Эти обстоятельства очень хороши для области компьютерного зрения, так как исследовательские задачи теперь будут разрабатываться со всех возможных направлений! Приглашаем читателя присоединиться к этой деятельности.

## 1.6 СМЕЖНЫЕ ОБЛАСТИ

Компьютерное зрение связано со многими другими дисциплинами. В данной книге не представляется возможным глубоко рассмотреть все эти связи. Во-первых, важно отличать *обработку* и *распознавание изображений*. В обработке изображений главное внимание уделяется преобразованию одних изображений в другие, в то время как распознавание изображений сосредоточено на формировании выводов на основе изображений и явном получении необходимых для этого описаний сцен. Обработка изображений часто используется совместно с распознаванием изображений и поэтому будет рассмотрена в этой книге довольно подробно. Книги, посвященные обработке изображений, обычно используют модель изображения в виде непрерывной функции  $f(x, y)$  от двух пространственных координат  $x$  и  $y$ . В данной книге в качестве модели изображения обычно рассматривается дискретный двумерный массив значений яркости  $I[r, c]$ . В этой книге термины *компьютерное зрение*, *машинное зрение* и *понимание изображений* используются как синонимы; однако специалисты могли бы указать между ними ряд различий.

Психология человеческого восприятия является очень важной областью по двум причинам: во-первых, ее надо учитывать при формировании изображений для восприятия человеком, и, во-вторых, изучение огромных способностей человека к распознаванию изображений может помочь в разработке компьютерных алгоритмов. Хотя в данной книге приведено некоторое обсуждение человеческого восприятия и распознавания, но основная направленность все-таки — это практическое решение прикладных задач.

Для нашего изучения важен ряд разделов физики, в том числе оптика и теория цвета. Основной материал будет представлен по необходимости; однако читатели, желающие стать экспертами в области освещения датчиков и объективов, должны будут обратиться к соответствующей литературе.

В книге используется много математических моделей. Для их усвоения читатель должен быть знаком с основами математического анализа, теории вероятностей и аналитической геометрии. Интуитивные понятия обработки изображений часто усиливают понимание математических моделей.

И, наконец, любая книга о компьютерном зрении должна быть тесно связана с компьютерной графикой. Обе области имеют дело со способами отображения и моделирования объектов; главное различие между ними в том, что компьютерное зрение связано с описанием и распознаванием объектов по изображениям, а компьютерная графика — с генерацией изображений по описаниям объектов. В последнее время наблюдается тесная взаимосвязь между этими областями по двум направлениям: компьютерная графика необходима для отображения результатов алгоритмов компьютерного зрения, а компьютерное зрение требуется для построения моделей объектов. Цифровые изображения очень часто используются в качестве входных данных для программ компьютерной графики.

## 1.7 СТРУКТУРА ОСТАВШЕЙСЯ ЧАСТИ КНИГИ

В предыдущих разделах были неформально введены многие понятия данной книги и отмечены главы, в которых эти понятия рассматриваются подробнее. Читатель получил представление о диапазоне задач компьютерного зрения и некоторых его методах. В нескольких следующих главах будут описываться вопросы двумерного компьютерного зрения.

При двумерной обработке изображение анализируется как самодостаточный объект в терминах пикселей, строк, точек пересечения отрезков, цвета, текстур и др. Знание о том, как изображение было получено из реального трехмерного мира, присутствует, но взаимосвязь между пикселями изображения и элементами реального мира очевидна — только различен масштаб. Например, рентгенолог по изображению сразу может сделать заключение о сужении потока крови без знания о том, каково устройство датчика или какой объем тела представлен одним пикселем. Аналогично может поступить и программа компьютерного зрения. Похожим образом, устройство алгоритма распознавания символов не зависит от реального размера сканированного шрифта. Материал гл. 2–11 носит двумерный характер и является более простым, чем материал гл. 12–16.

Для методов, описанных в гл. 12–15, трехмерная природа объектов и информация о точках наблюдения являются критически важными. В трехмерном компьютерном зрении анализ нельзя выполнить по координатам объектов на отдельном изображении, так как нам необходимо узнать взаимосвязь нескольких изображений или изображений и моделей; или требуется установить связь между полем зрения датчика и рабочим пространством робота. В гл. 12–15 будет рассматриваться обработка трехмерных сцен, а не двумерных изображений. Наиболее важным инструментом анализа является аналитическая геометрия в трехмерном пространстве. Как и в компьютерной графике, переход от двумерных задач к трехмерным оказывается весьма значительным и в смысле используемых абстракций и моделей, и в смысле вычислительных усилий.

## 1.8 ЛИТЕРАТУРА

Литература по компьютерному зрению отличается высокой степенью специализации в конкретных прикладных областях. Например, в статье Fleck и др. рассматривается вопрос обнаружения порнографических изображений для защиты детских компьютеров. Наше рассмотрение алгоритма подсчета отверстий взято из работы [Kopydlowski, 1983], в которой описано его применение для контроля балок грузовиков. Проектирование датчиков для спутников сильно отличается от проектирования медицинских инструментов, а производственные системы отличаются и от тех, и от других. В качестве работ по специальным областям можно назвать статьи [Nagy, 1972] и [Hord, 1982] по дистанционному зондированию, [Glasby, 1995] по биологическим наукам, [Ollus, 1987] и [QCAV, 1999] по промышленным приложениям, [ASAE, 1983] по

сельскохозяйственным приложениям. Одна из первых работ по цветным ПЗС-камерам принадлежит [Dillon и др., 1978].

Задачи, методы, и теория, общие для различных прикладных областей, являются, конечно, предметом учебников и включены в данную книгу. Вероятно, первым учебником по обработке изображений с помощью компьютера является книга Розенфельда [Rosenfeld, 1969], в которой содержится материал по обработке изображений без использования высокоуровневых моделей. В книге [Ballard, Brown, 1982], возможно, первом учебнике по компьютерному зрению, основное внимание уделено анализу изображений с использованием высокоуровневых моделей. Книгу [Levine, 1985] можно отметить за включение существенного материала о зрительной системе человека. Двухтомник Харалика и Шапиро [Haralick, Shapiro, 1992] является современным сборником алгоритмов и их математических обоснований. Книга [Jain, Kasturi, Schunk, 1995] представляет собой современное введение в компьютерное зрение, преимущественно с инженерной точки зрения.

1. ASAE. 1983. Robotics and intelligent machines in agriculture. *Proc. 1st Int. Conf. Robotics and Intelligent Machines in Agriculture* (2–4 Oct. 1983), American Society of Agricultural Engineers, Tampa: FL, St. Joseph, MI.
2. Ballard, D. H., and C. M. Brown. 1982. *Comput. Vision*. Prentice-Hall, Englewood Cliffs, NJ.
3. Dillon, P., D. Lewis, and F. Kaspar. 1978. Color imaging system using a single CCD area array. *IEEE Trans. Electron Devices*, ED-25(2):102–107.
4. Fleck, M., D. Forsyth, and C. Pregler. 1996. Finding naked people [in images]. *Proc. European Conf. Comput. Vision*. Springer-Verlag, New York, 593–602.
5. Glasby, C. A., and G. W. Horgan. 1995. *Image Analysis for the Biological Sciences*. John Wiley & Sons, Chichester, England.
6. Haralick, R., and L. Shapiro. 1992/3. *Computer and Robot Vision, Volumes I and II*. Addison-Wesley, New York.
7. Hord, R. M. 1982. *Digital Image Processing of Remotely Sensed Data*. Academic Press, New York.
8. Igarashi, T., ed. 1983. *World Trademarks and Logotypes*. Graphic-sha, Tokyo.
9. ——— ed. 1987. *World Trademarks and Logotypes II: A Collection of International Symbols and Their Applications*. Graphic-sha, Tokyo.
10. Jain, R., R. Kasturi, and B. Schunk. 1995. *Machine Vision*. McGraw-Hill, New York.
11. Kopydlowski, D. 1983. 100% inspection of crossbars using machine vision. Publication MS83-210, Society of Manufacturing Engineers, Dearborn, MI.
12. Levine, M. D. 1985. *Vision in Man and Machine*. McGraw-Hill, New York.
13. Nagy, G., 1972, Digital image processing activities in remote sensing for Earth resources. *Proc. IEEE*, v. 60(10):1177–1200.
14. Ollus, M., ed. 1987. Digital image processing in industrial applications. *Proc. 1st IFAC Workshop*, Espoo, Finland (10–12 June 1986), Pergamon Press, Oxford.

15. Pratt, W. 1991. *Digital Image Processing*, 2nd ed. John Wiley, New York.
16. QCAV. 1999. Quality control by artificial vision. *Proc. 5th Int. Conf. Quality Control by Artificial Vision* (18–21 May 1999), Trois-Rivieres, Canada.
17. Rosenfeld, A. 1969. *Picture Processing by Computer*. Academic Press, New York.

## 1.9 ДОПОЛНИТЕЛЬНЫЕ УПРАЖНЕНИЯ

Ниже приведено несколько дополнительных вопросов. Предполагается, что ответы на них будут представлены в форме рефератов или эссе. Часть вопросов требуют количественных выкладок. Большинство вопросов более подробно будут рассмотрены в оставшейся части книги.

---

### Упражнение 1.8. Задача о розничной продаже

Рассмотрим задачу, решение которой важно для автоматизации бакалейно-гастрономического магазина. Технология штрих-кодов упрощает идентификацию помеченных товаров: товар надо только провести по считывателю штрих-кода, пока не раздастся звуковой сигнал. Эта система не подходит для продукции, которую покупатель выбирает в неупакованном виде и в неограниченном количестве, например, фруктов. Тогда контролер вынужден останавливаться и предпринимать особые действия. Что это за действия? Как вы думаете, можно ли поместить камеру выше или внутри считывателя штрих-кодов, чтобы сообщать кассовому аппарату, какого типа продукция была взята? Предположим, что с помощью описанных в данной книге методов можно разработать систему компьютерного зрения, способную определять разницу между зеленым шпинатом и зелеными огурцами, яблоками разных сортов и т. д. Опишите, как подобную систему компьютерного зрения можно было бы интегрировать с другими существующими технологиями для учета товаров и формирования счета при розничной продаже.

---

### Упражнение 1.9. Подсчет бактерий

Сравните изображение бактерий на рис. 1.8 и образец автоматически вычисленных признаков на рис. 1.12. Можно ли по этим признакам определить количество бактерий, скажем, с точностью 10%? Объясните ответ.

---

### Упражнение 1.10. Формирование трехмерной модели по видеосъемке

Допустим, что есть видеосъемка Нотр-Дамского кафедрального собора. Видеосъемка подготовлена человеком, обошедшим собор вокруг и внутри, т. е. доступны виды с многих точек зрения. Как вы думаете, можно ли сформировать правдоподобную трехмерную модель собора по видеосъемке? (Допустим,

для использования архитекторами.) Если нет, то почему? Если да, как вы предполагаете строить трехмерную модель по двумерным изображениям?

---

**Упражнение 1.11.** Вычисление контраста

Предложите метод вычисления контраста по окрестности пикселей размером  $3 \times 3$ , аналогичный показанному на рис. 1.9. Допустим, что значения 9 входных пикселей являются значениями яркости в диапазоне от 0 до 255 и выходное значение, являющееся мерой контраста, тоже должно быть в диапазоне от 0 до 255. (На рис. 1.9 (справа) представлены только два значения пикселей 0 и 255, но вы можете использовать весь диапазон.)

---

**Упражнение 1.12.** Интерпретация лица

(а) Легко ли вы можете определить пол и приблизительный возраст людей на фотографиях в журнале? (б) Физиологи могут сказать, что люди обладают способностью по виду лица немедленно определить возраст, пол и степень враждебности человека. Как вы думаете, основана ли эта способность на характерных признаках изображения, и что это могли бы быть за признаки? Если вы думаете, что признаки изображения здесь ни при чем, то объясните, как могут люди принимать подобные решения?

---

**Упражнение 1.13.** Правда ли, что изображение дороже тысячи слов?

Рассмотрите следующий фрагмент из *Шум и ярость* (У. Фолкнер). Как вы думаете, может ли компьютер сформировать такое описание по видеосъемке описываемой во фрагменте сцены?

*Сквозь сумерки пахнет речными излучками и последний отсвет мирно лег на плесы, как на куски расколотого зеркала, а за ними в бледном чистом воздухе уже показались огни и слегка дрожат удаленными мотыльками.*

(Пер. с англ. Ю. Палиевской, изд-во Правда, 1989)

---

**Упражнение 1.14.** К вопросу о корректности подсчета отверстий.\*

Для решения этой задачи требуется дополнительный анализ и материал, выходящий за рамки данной главы, поэтому задачу можно рассматривать как факультативную. (а) Сколько возможных шаблонов окрестности  $2 \times 2$  может быть на бинарном изображении? Перечислите их все. (б) Какие шаблоны из пункта (а) не могут встретиться на 4-связном бинарном изображении? Будем считать *границей точки* центр решетки в пиксельной окрестности  $2 \times 2$ , которая содержит значения и 0, и 1. (в) Объясните, что отдельное отверстие можно найти просто подсчетом количества шаблонов 'e' и 'i' вдоль его границы и что формула  $n = (e - i)/4$  корректна при наличии только одного отверстия.



- (г) Объясните, что у двух отверстий не может быть общей граничной точки.
- (д) Объясните, что эта формула корректна при наличии любого количества отверстий.

---

**Упражнение 1.15.** Удобны ли бинарные изображения?

Рассмотрите изображения перечисленных ниже сцен и их преобразование в бинарные изображения, напоминающие изображения бактерий, на которых объекты отмечены как области значений 0, а фон, не относящийся к объектам, как области значений 1 (или наоборот). Подумайте, возможно ли это для перечисленных сцен. По вашему мнению, можно ли создать такие бинарные изображения?

1. Входное изображение — это изображение бумажного листа с напечатанными символами и отсканированное с помощью планшетного сканера. Конечная цель состоит в распознавании большинства напечатанных символов и формировании текстового файла, который можно было бы редактировать в текстовом редакторе.
2. Входное изображение — рентгеновский снимок человеческой головы. Надо найти инородные тела или опухоли как области из 0 на фоне значений 1.
3. Входное изображение — спутниковая фотография окрестностей Ричмонда, сделанная весной. Настройкой датчика или с помощью какого-либо простого компьютерного алгоритма требуется создать изображения, где 0 означает наличие зарослей азалии, а 1 — отсутствие этого растения.
4. Требуется проверить диаметр стержня клапана в автомобильном двигателе, подсчитывая количество пикселей вдоль тени от стержня. Поскольку ежедневно требуется контролировать тысячи изготавливаемых клапанов, то можно обосновать необходимость управления освещением и применение дорогого специализированного оборудования.

# Формирование и представление изображений

Значительный объем информации об окружающем мире люди получают посредством зрительного восприятия. Свет, отраженный от объектов, а иногда и прошедший сквозь них, формирует изображение на сетчатке каждого глаза. Из этой пары изображений человек извлекает много сведений о структуре трехмерной окружающей среды. Следовательно, важными компонентами зрительного восприятия являются: (а) сцена с объектами, (б) освещение объектов и (в) восприятие освещения, отраженного от объектов (или прошедшего сквозь них).

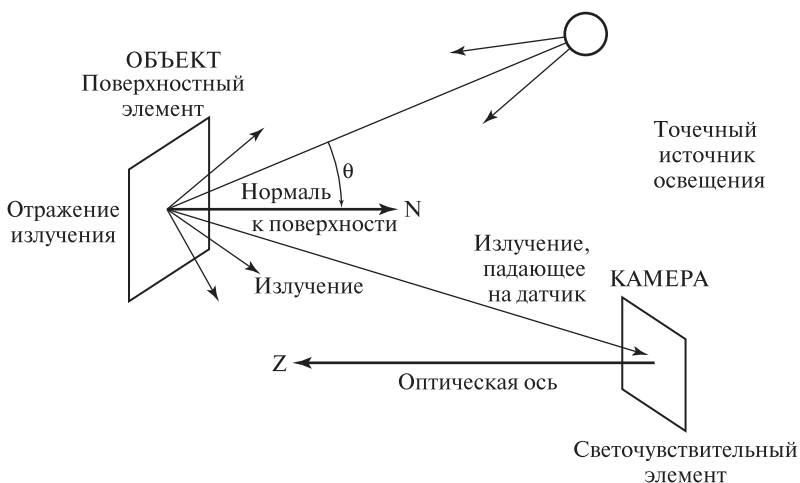
Основная цель данной главы заключается в рассмотрении способов формирования цифровых изображений двумерных или трехмерных сцен с помощью датчиков. С помощью существующих устройств формирования изображений можно воспринимать излучение различных видов, отражаемое или проникающее внутрь объектов физического мира. Двумерное цифровое изображение представляет собой массив значений интенсивности (яркости) света, отраженного или прошедшего сквозь объекты: это изображение обрабатывается машиной или компьютерной программой с целью формирования решений относительно сцены. Часто двумерное изображение является проекцией трехмерной сцены; это наиболее распространенное представление, используемое в машинном зрении и в этой книге. В конце главы мы обсудим некоторые взаимосвязи между структурами в трехмерном мире и структурами на двумерном изображении.

Некоторые разделы этой главы помечены символом \*, означающим, что в них содержатся технические подробности, которые читатель может пропустить, если соответствующая тема сейчас не представляет для него особого интереса.

## 2.1 ВОСПРИЯТИЕ СВЕТА

Существенную часть истории науки можно описать, рассматривая развитие приборов для восприятия и генерации различных видов электромагнитного излучения, например, радиоволн, рентгеновского излучения, микроволнового излучения и др. Химические соединения в рецепторах человеческого глаза чувствительны к излучению (свету) с длиной волны примерно от 400 нм (фиолетовый свет) до 800 нм (красный). ПЗС-датчики (см. рис. 2.2), кроме света, способны воспринимать излучение с длиной волны более 800 нм (инфракрасное излучение). Существуют устройства для детектирования коротковолнового рентгеновского излучения, а также длинноволнового излучения радиодиапазона. Излучение разных длин волн обладает различными свойствами; например, рентгеновские лучи могут проходить сквозь человеческие кости, а длинные инфракрасные волны не могут преодолеть даже облака.

На рис. 2.1 показана простая модель обычного фотографирования. Поверхностный элемент, освещаемый одиночным источником (солнцем или лампой-вспышкой), отражает излучение в направлении камеры, которая воспринимает это излучение с помощью специального светочувствительного покрытия на фотопленке. Более подробно данная модель рассматривается в гл. 6. Излучение светового диапазона возникает в результате генерации или отражения вблизи поверхности объектов. К тематике данной книги имеют отношение многие свойства электромагнитного излучения. Однако обычно мы будем давать качественное описание явлений, оставляя количественные детали книгам по физике или оптике. В прикладных инженерных задачах часто требуется знать некоторые свойства материала наблюдаемых объектов, воспринимаемого излучения и используемого датчика.



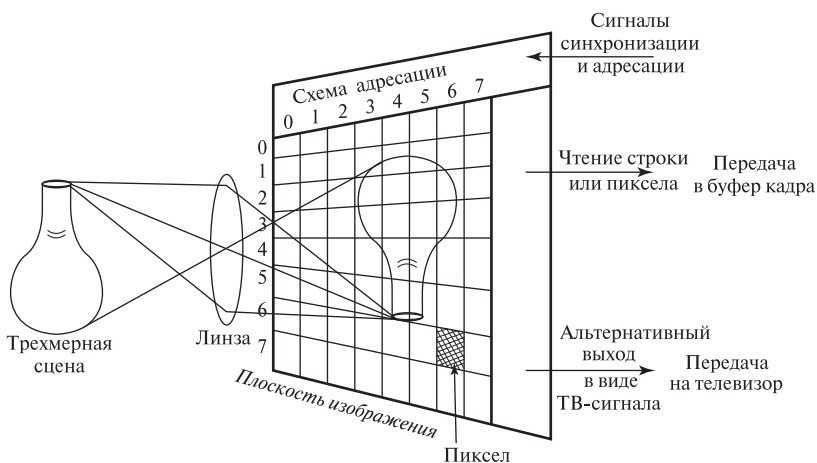
**Рис. 2.1.** Отражение излучения, распространяющегося от одиночного источника света.

## 2.2 УСТРОЙСТВА ДЛЯ ФОРМИРОВАНИЯ ИЗОБРАЖЕНИЙ

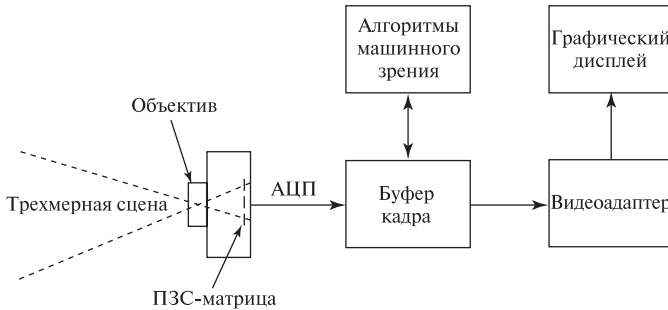
Для генерации цифровых изображений существует много различных устройств. Они различаются как принципом восприятия электромагнитного излучения, так и своим электромеханическим устройством. В данной главе рассматриваются несколько различных датчиков. Наиболее распространенные из них обсуждаются в данном разделе, ряд других оставлен для дополнительного разд. 2.9. При рассмотрении датчиков основная цель заключалась в демонстрации наиболее важных функциональных и принципиальных особенностей каждого датчика. Технические подробности могут быть найдены в дополнительной литературе.

### 2.2.1 Камеры на основе ПЗС

На рис. 2.2 показана камера, разработанная с использованием технологии приборов с зарядовой связью (ПЗС). Это наиболее гибкий и часто используемый тип устройств ввода изображений для систем машинного зрения. ПЗС-камера очень похожа на фотокамеру с фотопленкой 35 мм, которая часто используется для домашних фотографий. Различие в том, что в плоскости изображения у ПЗС-камеры вместо фотопленки со светочувствительным химическим покрытием располагается матрица из маленьких твердотельных ячеек, преобразующих световую энергию в электрический заряд. Каждая ячейка преобразует получаемую световую энергию в электрический заряд. Сначала все ячейки очищаются, а затем под действием падающего света они начинают накапливать заряд. Для управления временем накопления может использоваться затвор. Плоскость изображения действует как цифровая память, которую можно по-



**Рис. 2.2.** Получение изображения вазы с помощью ПЗС-камеры. Дискретные ячейки преобразуют световую энергию в электрические заряды, которые при вводе в компьютер представляются в виде небольших чисел.



**Рис. 2.3.** Центральное место *буфера кадра* в процессе обработки изображений.

строчно считывать в управляемом компьютером процессе ввода изображений. На рис. 2.2 приведена схема простой монохромной камеры.

Если цифровое изображение состоит из 500 строк и 500 столбцов однобайтных значений яркости, то для хранения такого изображения потребуется массив размером четверть миллиона байт. Иногда ПЗС-камера подключается к компьютерной плате, называемой *фреймграббером*, которая содержит память для изображения и, возможно, предоставляет средства для управления камерой. В настоящее время разработаны способы непосредственного обмена цифровыми данными между ПЗС-камерами и компьютерами (например, стандарт IEEE-1394). Производители современных камер предлагают цифровые камеры, которые могут хранить несколько десятков изображений непосредственно внутри камеры; во многих из них для хранения изображений используются карты памяти. Записанные в памяти камеры изображения можно в любой момент ввести в компьютер для дальнейшей обработки. На рис. 2.3 приведена схема компьютерной системы с вводом данных от камеры и графическим выводом. Это типичная система для промышленных или медицинских зрительных задач. Она также типична для *мультимедиа*-компьютеров, которые могут быть укомплектованы недорогой камерой для проведения телеконференций. Центральное место в этой схеме отводится *буферу кадра* (frame buffer) — высокоскоростному хранилищу изображения: камера генерирует входное изображение, которое после аналого-цифрового преобразования сохраняется в цифровой форме в буфере кадра. Изображение в буфере кадра доступно для отображения пользователю и для обработки различными компьютерными алгоритмами. В действительности буфер кадра может хранить несколько исходных или преобразованных изображений.

Компьютерная программа, обрабатывающая цифровое изображение, может обращаться к значениям пикселей как к элементам массива  $\mathbf{I}[\mathbf{r}, \mathbf{c}]$  или  $\mathbf{I}[\mathbf{r}][\mathbf{c}]$ , где  $\mathbf{I}$  — имя массива, а  $\mathbf{r}$  и  $\mathbf{c}$  — номера строк и столбцов, соответственно. Такая форма обозначений используется в представленных в данной книге алгоритмах. Некоторые камеры можно настроить так, что они будут генерировать *бинарные изображения* (binary image) — изображения, пиксели которых

принимают значения 0 или 1 для представления низкой и высокой яркости (или наоборот). Подобное изображение можно получить также с помощью простого алгоритма, заменяющего все пикселы с яркостями ниже некоторого *порогового значения* (threshold)  $t$  на 0, а все пикселы, яркость которых равна или больше  $t$ , на значение 1. Подобный пример был приведен в гл. 1, где ЯМР-изображение было бинаризовано с целью выделения сильных потоков крови на фоне слабых.

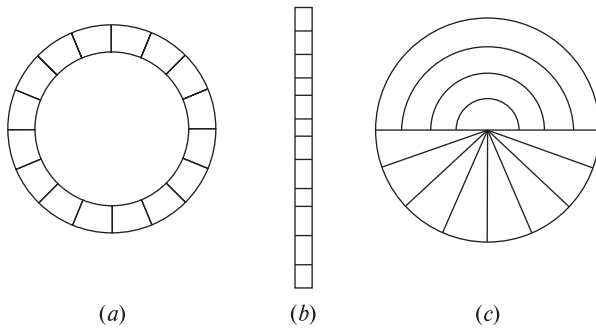
### 2.2.2 Формирование изображения

В качестве геометрической модели формирования изображения можно рассматривать проектирование каждой точки трехмерной сцены через *центр проектирования* или *центр объектива* на плоскость изображения. Яркость точки изображения связана с интенсивностью излучения от точки трехмерной поверхности: эта взаимосвязь довольно сложна и будет рассмотрена позже. Данная модель может быть реализована физически — *камеру-обскуру* (камеру с дырочным объективом, pin-hole camera) действительно можно сделать, используя камеру с корпусом в виде коробки и маленьким отверстием вместо объектива. В ПЗС-камерах обычно применяются линзовые объективы, так же как и в традиционных 35-мм фотоаппаратах. На рис. 2.2 показана одиночная линза с двумя выпуклыми поверхностями. Большинство настоящих объективов являются составными и содержат больше двух преломляющих поверхностей. Необходимо сделать два важных замечания.

Во-первых, объектив работает как собиратель света: точки изображения достигают свет, распространяющийся внутри некоторого конуса световых лучей, исходящих из трехмерной точки. На рис. 2.2 показаны три луча, проектируемых с верхней части вазы; они определяют границы конуса световых лучей, собираемых линзой для формирования только верхней части вазы. Похожие конусы лучей существуют для всех других точек сцены. Из-за того, что линзы объектива геометрически неидеальны, из-за зависимости преломления светового излучения от длины волны и некоторых других явлений, конус лучей в действительности формирует на плоскости изображения размытое пятно, называемое *кружком рассеяния* (circle of confusion).

Во-вторых, массив элементов ПЗС-датчика построен из физически дискретных ячеек, а не из бесконечно малых точек. Поэтому каждая ячейка датчика накапливает отклик от лучей, приходящих от некоторой окрестности точки трехмерной поверхности. Эти два эффекта приводят к *размытию* (blurring) изображения и определяют предел резкости и размеров наименьших видимых деталей на изображении.

Массивы ячеек ПЗС-датчиков производятся в виде микросхем и обычно имеют размеры около  $1 \times 1$  см. В массиве из  $640 \times 480$  или  $512 \times 512$  пикселов стороны пикселов равны примерно 0.001 дюйма. Существует ряд других способов размещения ячеек ПЗС-датчика на плоскости (или на строке) изображения (рис. 2.4). Линейный массив может быть использован в задачах, когда требуется измерить только ширину объектов или когда надо получить изображение



**Рис. 2.4.** Другие полезные конфигурации массивов ячеек ПЗС-камеры: (а) круговая, (b) линейная, (c) ROSA.

и проконтролировать длинный фрагмент материала, перемещающегося перед камерой. Существуют линейные датчики, содержащие в строке от 1000 до 5000 пикселей. Такие датчики можно использовать подобно ручному сканеру, когда линейный датчик перемещается над сканируемой поверхностью. Аналогичный метод применяется в планшетных сканерах, но в них для формирования изображения листа бумаги обычно используется оптическая система, перемещаемая с помощью высокоточного механического устройства. В настоящее время многие планшетные сканеры стоят порядка нескольких сотен долларов. Они используются для получения цифровых изображений из цветных фотографий или печатных документов. Для фокусирования «линейного отрезка» реальной сцены на линейный ПЗС-датчик обычно применяются цилиндрические линзы.

Круговой массив может быть полезен для контроля круговых шкал, например, на часах или спидометрах: объект надо точно разместить относительно камеры и затем сканировать круговой массив для получения изображения стрелки прибора. На рис. 2.4,(c) показана интересная конфигурация ROSA. Она обеспечивает аппаратное решение для накопления всей световой энергии, падающей или в секторы, или в круговые полосы. Эта конфигурация была разработана для дискретизации спектра мощности изображения, но может использоваться и в других задачах. Технология производства микросхем предоставляет возможности реализации и других конфигураций датчиков.

### Упражнение 2.1. Осмотр ПЗС-камеры

Если у вас есть доступ к ПЗС-камере, получите разрешение на изучение устройства этой камеры. Снимите объектив и обратите внимание на его устройство. Есть ли у него затвор для перекрытия светового потока, есть ли возможность регулировки апертуры для изменения размера конуса лучей, проходящих через объектив? Есть ли средства для изменения фокусного расстояния? Каково расстояние между объективом и ПЗС? Посмотрите на массив элементов ПЗС-датчика. Каковы размеры чувствительной области?

Видите ли вы отдельные элементы? Видны ли они с помощью увеличительного стекла?

---

### Упражнение 2.2

Предположим, что есть аналоговые часы и надо считывать их показания с помощью ПЗС-камеры, установленной прямо напротив часов. Центр часов отображается в центр цифрового изображения размерами  $256 \times 256$  пикселей. Часовая стрелка в два раза шире минутной. Длина часовой стрелки равна 0.7 длины минутной стрелки. Для обнаружения образов стрелок часов надо обрабатывать пиксели цифрового изображения в круговом порядке. (а) Приведите формулу для вычисления  $\mathbf{r}(t)$  и  $\mathbf{c}(t)$  для пикселей  $\mathbf{I}[\mathbf{r}, \mathbf{c}]$  на окружности радиуса  $R$  с центром в центре изображения  $\mathbf{I}[128, 128]$ , где параметр  $t$  — это угол между лучом из центра изображения к пикселу  $\mathbf{I}[\mathbf{r}, \mathbf{c}]$  и горизонтальной осью. (б) Требуется ли контролировать значение  $t$ , чтобы генерировать уникальную последовательность пикселей *цифровой окружности*? (в\*) Обратитесь к одной из дополнительных книг по компьютерной графике и найдите практический способ генерации цифровой окружности.

### 2.2.3 Видеокамеры

Видеокамеры предназначены для получения изображений для человеческого восприятия путем записи последовательностей изображений с типичной частотой 30 раз в секунду, что позволяет представить движение объекта в течение времени, а не только пространственные характеристики объекта, которые можно представить одиночными изображениями или *кадрами*. Для обеспечения непрерывности восприятия движения человеком используются 60 полукадров в секунду. Последовательность полукадров представляет собой последовательность поочередно следующих порций, состоящих из всех четных и всех нечетных строк изображения. При видеосъемке обычно так же записывается звуковой сигнал. Видеокамеры, предназначенные для получения изображений для машинной обработки, могут записывать изображения с необходимой в прикладных задачах скоростью и не нуждаются в использовании технологии полукадров.

Кадры видеопоследовательности разделяются маркерами. Для уменьшения объема данных обычно используется некоторый способ сжатия изображений. Аналоговые телевизионные стандарты были тщательно спроектированы для удовлетворения множества требований: ряд интересных особенностей стандартов позволяют использовать один и тот же сигнал и для цветных, и для черно-белых телевизоров, передавать совместно с изображениями звуковой или текстовый сигнал. Интересующийся читатель может обратиться к дополнительному материалу и краткому описанию способа сжатия MPEG в разделе 2.5. Далее мы продолжим рассмотрение цифровых видеоданных, считая их последовательностью двумерных цифровых изображений.

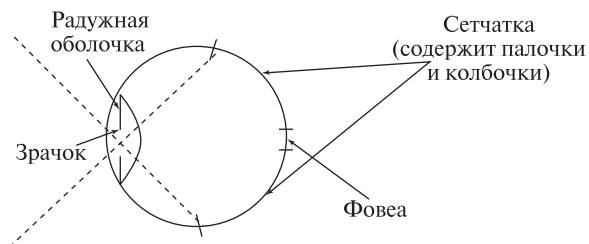


Технология ПЗС-камер для машинного зрения иногда испытывает негативное влияние со стороны телевизионных стандартов, разработанных в расчете на человеческое восприятие. Во-первых, это чередование в видеопоследовательности четных и нечетных полукадров, необходимых для достижения непрерывного восприятия движения человеком. Это привносит дополнительные сложности в системы машинного зрения. Во-вторых, отношение размеров сторон пикселей многих ПЗС-датчиков равно 4:3, так как у большинства дисплеев для просмотра изображений людьми отношение ширины к высоте также равно 4:3. Квадратные пикселы и единый масштабный множитель для систем машинного зрения были бы предпочтительнее. Огромный потребительский рынок оказывает влияние на разработку аппаратных устройств, приближая их к особенностям человеческого восприятия. Разработчики в области машинного зрения вынуждены или приспосабливаться к этим обстоятельствам, или пользоваться дорогими промышленными устройствами, изготавливаемыми в ограниченных количествах.

#### 2.2.4 Человеческий глаз

Грубо говоря, человеческий глаз представляет собой сферическую камеру с объективом с фокусным расстоянием 20 мм, который фокусирует изображение на *сетчатке*, расположенной напротив объектива и закрепленной на внутренней поверхности сферы (см. рис. 2.5). *Радужная оболочка* управляет количеством света, проходящего через объектив, путем регулирования размера *зрачка*. В каждом глазу есть примерно 100 миллионов ячеек-рецепторов — намного больше, чем в типичном ПЗС-датчике. Кроме того, сетчатка заполнена чувствительными ячейками неравномерно. В центральной области сетчатки, называемой *фовеа*, с высокой плотностью располагаются цветочувствительные рецепторы — *колбочки*. По мере удаления от центра, плотность колбочек уменьшается, а плотность черно-белых рецепторов, *палочек*, возрастает.

В световом излучении, поступающем в область фовеа от небольшой области поверхности наблюдаемого объекта, человеческий глаз воспринимает интенсивности трех различных основных цветов, так как спроектированный на сетчатку свет падает на колбочки трех различных типов. Колбочки каждого



**Рис. 2.5.** Грубая модель человеческого глаза как камеры для получения изображений. (Более подробное описание см. в книге [Levine, 1985].)

типа содержат специфический пигмент, чувствительный к свету с длинами волн в определенном диапазоне.

Одно из наиболее загадочных свойств человеческой зрительной системы глаз-мозг заключается в ее способности непрерывно воспринимать гладкие и стабильные образы пространственных сцен, хотя глаза постоянно двигаются. Эти *саккадические* движения глаз человека необходимы для корректного зрительного восприятия. В обработке зрительных входных данных участвует значительная часть человеческого мозга. Другие характеристики зрительной системы человека будут обсуждаться в различных местах этой книги: в частности, восприятие цвета более подробно будет рассмотрено в гл. 6.

---

### Упражнение 2.3

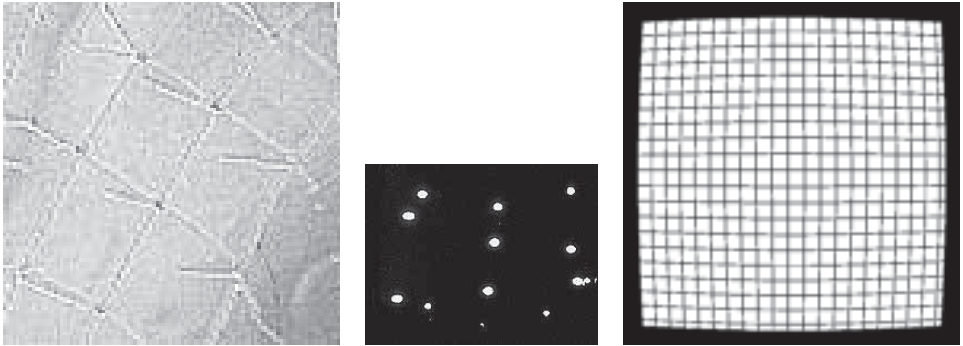
Предположим, что диаметр человеческого глаза составляет 2,5 см и что на части внутренней поверхности глаза, равной  $1/\pi$  от общей площади, расположены  $10^8$  палочек и колбочек. Чему равна средняя площадь, занимаемая одним рецептором? (Напомним, что плотность расположения рецепторов вблизи фovea превышает среднюю, а плотность расположения периферийных рецепторов ниже средней.)

## 2.3 ПРОБЛЕМЫ ФОРМИРОВАНИЯ ЦИФРОВЫХ ИЗОБРАЖЕНИЙ

В процессе восприятия изображений с помощью датчиков возникает ряд проблем. Ряд наиболее важных из них перечислен далее. Идеализированная модель формирования изображения, рассмотренная выше, обычно может служить лишь приближенной моделью реального физического процесса. Суммарный эффект различных проблем, связанных с формированием изображений, приводит к появлению на изображении некоторых искажений (дисторсий) геометрических и яркостных характеристик. Методы для коррекции искажений нескольких типов будут рассмотрены в гл. 11; однако в компьютерном зрении чаще применяются методы для принятия решений в условиях присутствия дефектов изображений.

### 2.3.1 Геометрические искажения

Геометрические искажения в процессе формирования изображения возникают по различным причинам. Из-за дефектов при изготовлении объектива лучи света, собираемые от поверхностного элемента сцены, не преломляются точно нужным образом. Бочкообразная дисторсия часто наблюдается при использовании короткофокусных объективов; при этом прямые линии в периферийных областях сцены выглядят согнутыми в сторону от центра изображения, как показано на рис. 2.6 (справа).



**Рис. 2.6.** Изображения с примерами различных искажений. (*Слева*) на пересечении некоторых ярких полос заметен эффект отсечения уровня яркости в процессе аналого-цифрового преобразования. (*В центре*) блюминг заключается в увеличении яркости соседей ярких пикселей. (*Справа*) при использовании короткофокусных объективов часто наблюдаются бочкообразные искажения.

### 2.3.2 Дисперсия

При прохождении через вещество световые лучи могут преломляться и рассеиваться. Изображения, полученные путем аэрофотосъемки или со спутников, особенно подвержены таким дефектам, которые появляются из-за водяных паров или температурных градиентов, придающих атмосфере свойства линзы.

### 2.3.3 Блюминг (избыточная яркость)

Поскольку дискретные детекторы, такие, как ячейки ПЗС, не идеально изолированы друг от друга, то заряд, накопленный в одной ячейке, может стекать в соседние ячейки. Термин *блюминг* относится к явлению, когда утечка заряда возникает в области яркой засветки плоскости изображения. В результате на изображении возникает яркий «цветок» с размерами больше, чем должны быть в действительности (рис. 2.6, в центре).

### 2.3.4 Неоднородности ПЗС-матрицы

Из-за дефектов производства у элементов ПЗС-матрицы может оказаться различная чувствительность, так что при одинаковой освещенности различные ячейки будут генерировать различный выходной сигнал. Тогда для точной оценки интенсивности может потребоваться определить полный массив масштабных множителей  $s[\mathbf{r}, \mathbf{c}]$  и смещений  $\mathbf{t}[\mathbf{r}, \mathbf{c}]$ , по одному для каждого пиксела. Это делается путем калибровки при однородном освещении, когда интенсивность можно вычислить как  $I_2[\mathbf{r}, \mathbf{c}] = s[\mathbf{r}, \mathbf{c}]I_1[\mathbf{r}, \mathbf{c}] + \mathbf{t}[\mathbf{r}, \mathbf{c}]$ . В предельном случае, на ПЗС-матрице могут быть несколько *мертвых ячеек*, которые совсем не генерируют выходной сигнал. Эти дефекты обнаруживаются в процессе инспекционного контроля. Их можно компенсировать программными сред-

ствами, присваивая мертвой ячейке выходной сигнал, равный усредненному отклику соседних ячеек.

### 2.3.5 Отсечение и циклический возврат

При аналого-цифровом преобразовании очень большие значения интенсивности могут быть усечены до максимального значения или старшие биты этого значения могут быть утеряны, так что значение интенсивности будет представлено значением некоторого меньшего значения (произойдет циклический возврат к началу диапазона значений интенсивности). На полутоновом изображении результат циклического возврата АЦП выглядит как яркая область с темным центром; на цветном изображении этот эффект может проявляться в изменении цвета. На рис. 2.6 (слева) показан пример циклического возврата: в области пересечения некоторых ярких полос присутствуют темные пиксели.

### 2.3.6 Хроматическая дисторсия

Световые волны различной длины преломляются линзой по-разному (*показатель преломления* линзы зависит от длины волны). В результате, световые волны различной длины от одного и того же малого участка сцены могут попасть в несколько различных пикселей изображения на чувствительной поверхности датчика. Например, на изображении очень резкой черно-белой границы на периферии сцены может наблюдаться постепенное (линейное) изменение интенсивности в пределах нескольких пикселей изображения.

### 2.3.7 Эффекты дискретизации

В процессе дискретизации значение интенсивности формируется для некоторой дискретной области сцены. Оно представляется одним из дискретных значений интенсивности и поэтому подвержено ошибкам смешивания и округления. Эти проблемы более подробно описаны в следующем разделе.

## 2.4 ФУНКЦИИ ИНТЕНСИВНОСТИ И ЦИФРОВЫЕ ИЗОБРАЖЕНИЯ

В данном разделе будут рассмотрены ряд понятий и система обозначений, которые важны и для теоретического рассмотрения, и для программирования алгоритмов обработки изображений.

### 2.4.1 Типы изображений

В обработке изображений удобно работать с понятиями *аналогового* и *цифрового изображения*. Функция интенсивности (picture function) — это математическая модель, часто используемая при анализе задач, в которых плодотворные результаты может принести рассмотрение изображения как функции двух переменных. Тогда для анализа изображений оказывается доступен весь аппарат математического анализа. Цифровое изображение — это просто двумер-

ный прямоугольный массив дискретных значений. Пространственная область изображения и диапазон значений интенсивности квантуются на дискретный набор значений, так что изображение может быть сохранено в компьютерной памяти с двумерной структурой. Обычно для представления интенсивности выбирают 8-битные (однobaйтные) числа, допускающие хранение значений от 0 до 255. Точность датчика часто не превосходит 256 градаций интенсивности, и для большинства задач этого достаточно. Кроме того, байты удобны для компьютерной обработки. Например, изображение можно представить в программе на языке Си как `char I[512][512]`. Каждый пиксел цветного изображения потребовал бы трех значений интенсивности (по одному для каждого из основных цветов). В некоторых медицинских задачах используется 10-битное кодирование, позволяющее представить 1024 различных значения интенсивности (что приближается к пределам человеческих возможностей их различения).

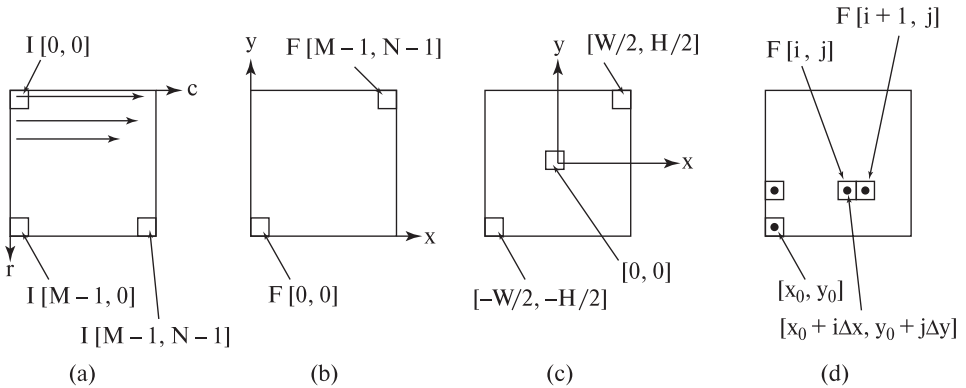
Следующие определения предназначены для разъяснения важнейших понятий обработки изображений, а также для введения системы обозначений, используемой далее в этой книге. Мы начнем с абстрактного понятия аналогового изображения, формируемого идеальной оптической системой (будем полагать, что точность этой системы бесконечна). Цифровое изображение формируется путем *выборки* значений этого аналогового изображения в дискретных положениях и представления интенсивности в этих положениях в виде дискретных значений. При формировании всех реальных изображений существуют пределы точности координат и интенсивности, обусловленные физическими процессами.

**2 Определение. Аналоговое изображение** — это двумерное изображение  $F(x, y)$ , характеризующееся бесконечной точностью представления по пространственным параметрам  $x$  и  $y$  и бесконечной точностью представления значений интенсивности в каждой пространственной точке  $(x, y)$ .

**3 Определение. Цифровое изображение** — это двумерное изображение  $I[r, c]$ , представленное в виде двумерного массива дискретных значений интенсивности, каждое из которых представлено с ограниченной точностью.

Математическая модель изображения как функции двух действительных пространственных переменных оказывается чрезвычайно полезной для описания изображений и для определения операций над ними. На рис. 2.7 (d) показано соответствие между пикселями и выборками непрерывного изображения в различных точках  $[x, y]$  плоскости изображения. Если в направлении  $X$  произведено  $M$  выборок на расстоянии  $w$  между ними, то расстояние между пикселями по этому направлению  $\Delta x = w/M$ . Формула, связывающая координаты центра пиксела с индексами элемента массива, в котором хранится интенсивность данного пиксела, приведена в правой части рис. 2.7.

**4 Определение. Функция интенсивности** — это математическое представление изображения как функции  $f(x, y)$ , зависящей от двух простран-



**Рис. 2.7.** Различные системы координат, применяемые в обработке изображений. (а) В растровой системе координат используются координаты строк и столбцов, а начало отсчета  $[0, 0]$  расположено в верхнем левом углу изображения. (б) Прямоугольная система координат с началом  $[0, 0]$  в левом нижнем углу. (в) Прямоугольная система координат с началом  $[0, 0]$  в центре изображения. (д) Связь координат центра пиксела  $[x, y]$  с площадкой аналогового изображения, значение интенсивности которой в результате дискретной выборки было помещена в элемент массива  $I[i, j]$ .

ственных переменных  $x$  и  $y$ . Переменные  $x$  и  $y$  принимают действительные значения, которые задают положение точки на изображении. Значения  $f(x, y)$  обычно также являются действительными и определяют интенсивность изображения в точке  $(x, y)$ .

**5 Определение. Полутоновое изображение** — это монохромное (черно-белое) цифровое изображение  $I[r, c]$ , у которого каждому пикселу соответствует одно значение интенсивности<sup>1</sup>.

**6 Определение. Многоспектральное (мультиспектральное) изображение** — это двумерное изображение  $M[x, y]$ , у которого каждой пространственной точке или пикселу соответствует вектор значений. В наиболее часто встречающемся случае цветных изображений вектор содержит 3 элемента.

**7 Определение. Бинарное изображение** — это цифровое изображение, пикселы которого имеют значения 0 или 1.

**8 Определение. Маркированное изображение** — это цифровое изображение  $L[r, c]$ , пикселы которого представлены в виде *символов* конечного алфавита. Символьное значение пиксела обозначает результат принятия

<sup>1</sup> В случае полутонового изображения интенсивность часто также называется яркостью пикселов. — *Прим. перев.*

некоторого решения относительно данного пиксела. Родственными понятиями являются **классификационное изображение** и **псевдоцветовое изображение**.

Система координат требуется для указания отдельных пикселов изображения, для работы с ними в компьютерной программе, для ссылки на них из математических формул или для указания пикселов относительно системы координат устройства. Различные системы координат, используемые в данной книге (и во многих других), показаны на рис. 2.7. К сожалению, в различных компьютерных средствах часто используются различные системы координат и пользователю в каждом случае необходимо изучать, какие именно. К счастью, основные понятия обработки изображений и компьютерного зрения не привязаны к системе координат. В данной книге понятия обычно рассматриваются применительно к прямоугольной декартовой системе координат, согласующейся с математическими учебниками, а в алгоритмах обработки изображений в основном применяется растровая система координат.

#### 2.4.2 Дискретизация изображений и пространственные измерения

Каждый пиксел цифрового изображения представляет собой результат измерения интенсивности (выборку) некоторой элементарной площадки аналогового изображения (рис. 2.2). Если спроектировать некоторый пиксел с плоскости изображения обратно на поверхности объектов наблюдаемой сцены, то размер соответствующего элемента сцены позволит оценить *номинальное разрешение* (nominal resolution) датчика. Например, если квадратный лист бумаги со стороной 25 см представляется в виде цифрового изображения размерами  $500 \times 500$  пикселов, то номинальное разрешение датчика в данном случае будет составлять 0.05 см. Понятие номинального разрешения не слишком полезно для сцен с большими различиями по глубине (дальности) наблюдаемых объектов, так как номинальное разрешение будет меняться в зависимости от глубины и ориентации поверхностей объектов.

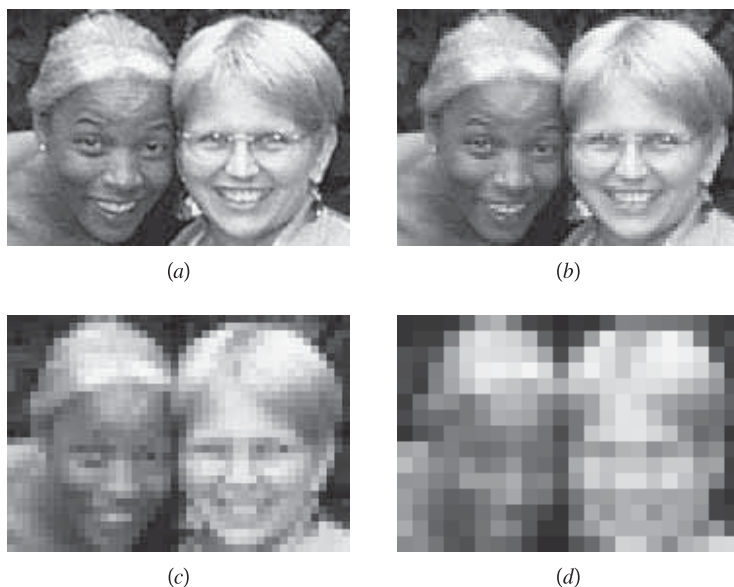
*Поле зрения* датчика изображений является характеристикой пространственного объема сцены, которую можно наблюдать с помощью данного датчика. *Разрешение* датчика характеризует его точность при выполнении пространственных измерений или при обнаружении характерных признаков. (При аккуратном использовании и при наличии некоторых данных о модели изображение размерами  $500 \times 500$  пикселов может быть использовано для выполнения измерений с точностью до  $1/5000$ , что обычно называется *субпиксельным разрешением*.)

**9 Определение.** Номинальное разрешение ПЗС-датчика равно размеру стороны поверхностного элемента сцены, которому соответствует отдельный пиксел плоскости изображения.

**10 Определение.** Термин **разрешение** относится к точности датчика применительно к выполнению измерений. Формально разрешение определяется

несколькими способами. Оно может быть определено в терминах реального мира, через номинальное разрешение, как «разрешение этого сканера составляет один метр на земной поверхности», или как количество пар отрезков на миллиметр, которые можно различить на полученном изображении. Совершенно другим понятием является количество доступных пикселей — «разрешение этой камеры равно  $640 \times 480$  пикселей». Последнее утверждение позволяет указать, на сколько частей может быть разделено поле зрения. Это важно как для выполнения точных измерений, так и для наблюдения заданной области сцены. Если точность измерений составляет дробную часть от номинального разрешения, то оно называется **субпиксельным разрешением**.

Понятие разрешения иллюстрируется на рис. 2.8 с помощью четырех изображений одних и тех же лиц. Человек способен различить знакомое лицо в разрешении  $64 \times 64$ , и, возможно,  $32 \times 32$ , но разрешения  $16 \times 16$  оказывается недостаточно. При решении задачи с применением компьютерного



**Рис. 2.8.** Четыре цифровых изображения двух лиц. (a) Изображение размерами 127 строк на 176 столбцов. (b) Изображение ( $126 \times 176$ ) получено путем осреднения каждой окрестности пикселей размерами  $2 \times 2$ . (c) Изображение ( $126 \times 176$ ) получено аналогичным образом из изображения (b). (d) Изображение ( $126 \times 176$ ) получено аналогичным образом из изображения (c). Эффективные номинальные разрешения равны, соответственно, ( $127 \times 176$ ), ( $63 \times 88$ ), ( $31 \times 44$ ) и ( $15 \times 22$ ). (Попробуйте посмотреть на изображения с блочной структурой «искоса». При этом резкие границы квадратов обычно выглядят более размытыми.) (С любезного разрешения Frank Biocca.)



зрения разработчик должен выбрать подходящее разрешение. Слишком низкое разрешение приведет к плохим результатам распознавания или неточным измерениям. Слишком высокое может чрезмерно замедлить выполнение алгоритмов и повлечь излишние затраты памяти.

**11 Определение.** **Поле зрения** датчика (field of view, FOV) определяется размерами наблюдаемой сцены, например,  $25 \times 25$  см. Так как эти размеры зависят от глубины, более полезным может оказаться понятие **углового поля зрения** (angular field of view). Например, угловое поле зрения может быть равно «55 на 40 градусов».

Поскольку пиксел изображения соответствует не точке, а некоторой области физического мира, то значение интенсивности пиксела часто определяется излучением, отраженным или излученным фрагментами различных материалов. Например, рассмотрим спутниковое изображение, в котором каждый пиксел соответствует участку земной поверхности размерами  $10 \times 10$  м. Очевидно, что на этом участке может одновременно оказаться вода, грунт, растительность и т. п. Тогда при формировании бинарного изображения возникает серьезная проблема. Снова рассмотрим упоминавшийся выше пример изображения листа бумаги, на котором напечатан текст плотностью 4 символа на сантиметр. Многие пикселы изображения будут попадать на границу символов и фона. При этом в значении пиксела будет смешиваться высокая интенсивность отраженного излучения от фона (бумаги) и низкая интенсивность излучения от символа. Суммарным результатом будет значение интенсивности, промежуточное между значениями, соответствующими чистой бумаге и символу. То есть каким бы ни было значение граничного пиксела (0 или 1), оно будет не вполне корректным.

На рис. 2.9 иллюстрируются проблемы, связанные с дискретизацией при формировании изображений. В качестве двумерной сцены рассмотрим массив  $10 \times 10$  из темных (яркость равна 0) и светлых плиток (яркость равна 8). На изображении плиткам соответствуют образы в виде двух ярких пятен и двух ярких линий различной толщины.

Предположим, изображение сцены проектируется на ПЗС-матрицу размерами  $5 \times 5$  так, что каждое множество  $2 \times 2$  смежных плиток проектируется в точности на один элемент датчика. Результирующее цифровое изображение приведено на рис. 2.9 (b). Левый верхний элемент ПЗС-датчика воспринимает интенсивность  $2 = (0 + 0 + 0 + 8)/4$ . Это значение равно усредненному значению интенсивности излучения от четырех смежных плиток. Множество из четырех светлых плиток в правой верхней части сцены проектируется на два элемента датчика. Каждый из них воспринимает излучение, отраженное от двух светлых и двух темных плиток. Одиночная строка из светлых плиток с яркостью 8 отображается на строку ПЗС-матрицы, элементы которой будут воспринимать интенсивность 4. Двойная строка светлых плиток отображается в две строки с интенсивностью 4, но в каждой из этих строк изображения смешиваются по две строки плиток сцены (одна из светлых, одна из темных). Если изображение подвергнуть бинаризации с порогом  $t = 3$ , то яркий образ

0	0	0	0	0	0	0	0	0	0
0	8	0	0	0	0	8	8	0	0
0	0	0	0	0	0	8	8	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
8	8	8	8	8	8	8	8	8	8
0	0	0	0	0	0	0	0	0	0
8	8	8	8	8	8	8	8	8	8
8	8	8	8	8	8	8	8	8	8
0	0	0	0	0	0	0	0	0	0

(a)

2	0	0	4	0
0	0	0	4	0
4	4	4	4	4
4	4	4	4	4
4	4	4	4	4

(b)

0	0	0	0	0	0	0	0	0	0
0	8	0	0	0	8	8	0	0	0
0	0	0	0	0	0	8	8	0	0
0	0	0	0	0	0	0	0	0	0
8	8	8	8	8	8	8	8	8	8
0	0	0	0	0	0	0	0	0	0
8	8	8	8	8	8	8	8	8	8
8	8	8	8	8	8	8	8	8	8
0	0	0	0	0	0	0	0	0	0

(c)

2	0	4	4
0	0	0	0
4	4	4	4
8	8	8	8

(d)

**Рис. 2.9.** (a) Массив  $10 \times 10$  из плиток с яркостью 0 и 8. (b) Полутоновое изображение размерами  $5 \times 5$  пикселей. Значение каждого пиксела равно средней яркости окрестности плиток размерами  $2 \times 2$ . (c) Изображение, наблюдаемое камерой после смещения на одну плитку вниз и вправо. Дискретные значения яркости зависят и от физических размеров пиксела, и от его положения относительно сцены. (d) Значения пикселей изображения, полученного после смещения камеры. Проблемы интерпретации характерных признаков наблюдаемой сцены возникают и с изображением (b), и с (d).

от одной плитки будет потерян, а три остальных характерных признака на бинарном изображении сольются в одну область!

Если сместить ПЗС-датчик в вертикальном или горизонтальном направлении на расстояние, равное стороне плитки, то будет получено изображение, показанное на рис. 2.9(d). Форма яркой области из 4 плиток на рис. 2.9(d) искажена иначе, чем на рис. 2.9(b). Две яркие линии сцены на изображении (d) приведут к появлению наклонного перепада, а не однотонной области, как в случае (b). Кроме того, на рис. (d) заметны три отдельных объекта, в отличие двух на рис. (b). Рис. 2.9 демонстрирует нестабильность характерных признаков сцены с размерами образов порядка одного пиксела.

Рис. 2.9 демонстрирует влияние *пространственной дискретизации* на точность измерений и на возможности обнаружения образов объектов на изображениях. Небольшие по размеру характерные признаки могут быть пропущены или могут слиться с другими признаками. Даже при обнаружении крупных признаков возможны ошибки при определении их размеров по изображениям.

Обратите внимание, что множество из четырех ярких плиток отображается на пару вертикально или горизонтально расположенных элементов ПЗС, генерирующих значение интенсивности 4. При определении положения границы можно ожидать ошибку порядка 0.5 пиксела за счет округления значения *смешанного пиксела* в процессе бинаризации изображения. Это приведет к ошибке порядка одного пиксела при измерениях, выполняемых с использованием образа с двумя границами «объект-фон».

Если требуется обнаружить некоторые признаки на бинарном изображении, необходимо, чтобы размеры их образов составляли не менее двух пикселов. Это относится и к промежуткам между объектами. Рассмотрим изображение *точки*, завершающей предложение на изображении факсимильного документа. Допустим, образ точки занимает ровно один пиксел. Если точка будет проектироваться точно на границу четырех элементов ПЗС-матрицы, то каждый из них будет воспринимать больше излучения от чистых областей бумаги, чем от напечатанной точки. В данном случае при формировании бинарного изображения велика вероятность потери образа точки.

**12 Определение. Смешанным пикселом** (mixed pixel) называется пиксел изображения, значение которого получено в результате измерения интенсивности излучения от поверхностей объектов сцены, сделанных из различных материалов.

---

#### **Упражнение 2.4.** Изменчивость площади образов объектов

Рассмотрим темный прямоугольник на белом листе бумаги. На изображении этот прямоугольник занимает область размерами  $5.9 \times 8.1$  пикселов. При формировании бинарного изображения пикселам присваиваются значения 0 или 1 в зависимости от того, похож пиксел больше на объект или на фон. Лист бумаги с прямоугольником можно смещать параллельным переносом (стороны листа остаются параллельны строкам и столбцам ПЗС-матрицы). Чему равна минимально и максимально возможная площадь образа прямоугольника на бинарном изображении?

---

#### **Упражнение 2.5.** Потеря признаков в виде тонких отрезков

На изображении печатной платы границы проводника выглядят как два ярких параллельных отрезка. Ширина каждого отрезка равна 0.8 пиксела. Возможна ли ситуация, когда на бинарном изображении (сформированном аналогично предыдущему упражнению) одна из двух линий будет пропадать?

В гл. 13 будет рассматриваться известное в оптике уравнение тонкой линзы, а также связь этого уравнения с разрешением камеры, резкостью изображения и глубиной поля зрения. В настоящий момент, после качественного обсуждения ряда характеристик восприятия изображений и понятий разрешения и смешанных пикселов, объема предварительного материала будет достаточно, чтобы читатель смог начать работу над некоторыми задачами двумерного

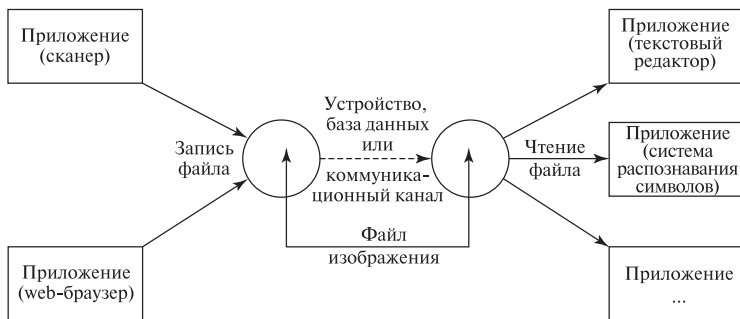
компьютерного зрения. Например, такими задачами являются поиск заданных объектов на микроскопических изображениях, контроль печатных плат, или обнаружение теней от освещенного трехмерного объекта. В этих задачах среда, в которой формируются изображения, должна быть организована так, чтобы образы характерных признаков на изображении гарантированно присутствовали и имели достаточные для обнаружения размеры. В предположении, что сцены специально организованы так, чтобы на изображениях не присутствовало существенных пространственных деталей, для анализа изображений можно будет применять двумерные методы из гл. 3–10.

### Упражнение 2.6. Распознавание банкнот

Обдумайте возможное устройство датчика для торгового автомата, принимающего банкноты достоинством \$1, \$5, \$10, и \$20. Требуется рассмотреть только наиболее общие особенности датчика. Разрабатывать алгоритм распознавания или рассматривать вопросы обнаружения фальшивых банкнот не надо. Предположим, что для получения изображений применяется линейный ПЗС-датчик, который формирует изображение банкноты по мере ее перемещения внутрь автомата. (а) Какие объектив и освещение должны использоваться? (б) Сколько элементов потребуется в линейном ПЗС-массиве?

## 2.5 ФОРМАТЫ ЦИФРОВЫХ ИЗОБРАЖЕНИЙ\*

Цифровые изображения широко используются в телекоммуникационных приложениях, базах данных и машинном зрении. Для обмена данными между различным аппаратным и программным обеспечением были разработаны стандартные форматы изображений. Одна из возможных ситуаций показана на рис. 2.10. К сожалению, в настоящее время используются десятки различных форматов. Некоторые наиболее важные из них кратко рассматриваются в данном разделе.



**Рис. 2.10.** Графические данные в виде изображений создают, получают и преобразуют многие устройства и прикладные программы. Стандартные форматы файлов изображений необходимы для эффективного совместного использования нескольких устройств и программ.

Исходное изображение можно рассматривать как обычный поток байт, представляющих пиксели изображения по очереди, строка за строкой. Такой порядок часто называется *растровым порядком* (raster order). В потоке байт могут присутствовать маркеры для разделения строк изображения. В пиксельных данных исходного изображения отсутствуют такие параметры, как тип изображения, размеры, время и способ получения. Информацию подобного характера часто записывают вручную на этикетке видеокассеты или в лабораторном журнале — но это неудобно. (В проекте с участием одного из авторов книги перед записью экспериментальных изображений на видеокассету сначала записывалось изображение штрих-кода. Компьютерная программа впоследствии могла обработать изображение штрих-кода и получить всю необходимую неграфическую информацию с интерпретацией экспериментальных данных.) Относительно недавно разработанные стандартные форматы предусматривают наличие заголовка для хранения неграфической информации, необходимой для маркировки или для декодирования пиксельных данных.

Ряд форматов были разработаны компаниями, выпускающими графическое оборудование и аппаратуру для обработки изображений. В некоторых случаях (а в некоторых нет) на форматы цифровых изображений доступны открытая документация, а для работы с ними существует открытое программное обеспечение. Приведенные далее подробные сведения могут помочь читателю, интересующемуся практическими деталями обработки компьютерных изображений. Хотя с развитием технологий технические детали быстро меняются, некоторые основные понятия, приведенные в данном разделе, выдержали испытание временем.

### 2.5.1 Заголовок файла изображения

*Заголовок файла* делает файл изображения самодостаточным для дальнейшей обработки различными программами — в нем содержатся все характеристики, необходимые для работы с изображением. В заголовке могут храниться размеры и тип изображения, дата создания, название и т. п. В заголовке также может присутствовать палитра или таблица кодировки для интерпретации значений пикселей. Полезной, но не часто доступной возможностью является раздел истории обработки для хранения комментариев о создании и обработке изображения.

### 2.5.2 Данные изображения

Некоторые форматы рассчитаны на ограниченный набор типов изображений, например, на бинарные и черно-белые. Однако наиболее распространенные форматы продолжают развиваться. Количество поддерживаемых ими свойств и типов изображений постоянно увеличивается. Различные форматы могут отличаться допустимыми предельными размерами изображений. Некоторые форматы рассчитаны на работу с последовательностями кадров. Появившиеся

сравнительно недавно мультимедиа-форматы рассчитаны на хранение данных изображений совместно с текстовой, звуковой и графической информацией.

### 2.5.3 Сжатие данных

Во многих форматах предусмотрено *сжатие* данных изображений. При этом непосредственно в файл изображения помещаются значения не всех пикселей. В зависимости от требуемого качества и метода сжатия возможно сокращение файлового размера изображения до 30 или даже 3 процентов от исходного объема. Методы сжатия делятся на *сжатие без потерь* (lossless compression) и *сжатие с потерями* (lossy compression) информации. При сжатии без потерь исходное изображение может быть точно восстановлено по данным из файла изображения. Напротив, при сжатии с потерями исходное изображение может быть восстановлено не абсолютно точно. Иногда потеря качества заметна, но не всегда.

При использовании сжатия в файле изображения также требуется хранить некоторую дополнительную информацию о типе и параметрах метода сжатия. Большинство цифровых изображений сильно отличаются от символической цифровой информации тем, что потеря или изменение нескольких бит в данных изображения для человеческого восприятия либо будет совсем незаметна, либо эффект будет несущественным. Обычно результаты изменений оказываются несущественными и применительно к компьютерной обработке изображений. Это сильно отличается от большинства других типов файлов. Например, изменение одного бита в записи о сотруднике в базе данных может привести к изменению зарплаты сотрудника на \$8192 или адреса места жительства с *A* на *B*.

Сжатие изображений — это очень интересная область, затрагивающая множество различных вопросов, от цифровой обработки сигналов до распознавания объектов. Хотя систематическое рассмотрение сжатия изображений не входит в цели данной книги, ряд методов в ней будут обсуждаться.

**13 Определение.** Методом сжатия без потерь называется такой метод сжатия изображений, для которого существует метод распаковки, позволяющий точно (с точностью до бита) восстановить исходное изображение. В противном случае метод сжатия называется методом сжатия с потерями.

### 2.5.4 Часто используемые форматы

Изображения, иллюстрирующие данную книгу, хранились в нескольких различных форматах. Некоторые изображения были предоставлены нашими коллегами или получены из баз данных изображений в форматах GIF, JPG или даже PS. Другие изображения были получены путем сканирования фотографий. Их исходным цифровым форматом был GIF или TIFF. Простые операции по обработке изображений выполнялись с помощью программы xv, более

сложные — с помощью программы `hips` или специально написанных программ на языках Си и Си++.

Далее кратко описаны некоторые наиболее часто используемые файловые форматы. Графические форматы файлов и форматы файлов изображений продолжают развиваться. Основная тенденция развития заключается в добавлении в форматы все большего количества возможностей. Следует отметить, что некоторые из приведенных ниже деталей могут измениться. Об этих изменениях читатель может узнать в справочной документации по новым версиям файловых форматов.

### 2.5.5 Групповое кодирование бинарных изображений

*Групповое кодирование* (или сжатие повторяющихся последовательностей) является эффективным методом сжатия бинарных или маркированных изображений. Этот метод не только уменьшает объем памяти при хранении изображений, но и позволяет ускорить ряд операций обработки, например, множественные операции. Групповое кодирование хорошо работает с изображениями, в строках которых присутствует большое количество одинаковых пикселей.

Рассмотрим бинарное изображение. Для каждой строки изображения можно записать *количество* нулевых и единичных значений в порядке их следования. Пример приведен на рис. 2.11. На этом рисунке групповой код В является более компактным способом представления сжатого изображения. В этом коде указываются строчные индексы для границ только единичных участков строки исходного изображения.

Столбец <i>c</i>	:	00000000001111111111222222222233333333334444444444
Строка изображения <i>r</i>	:	0123456789012345678901234567890123456789012345678
Групповой код А	:	8(0)5(1)12(0)3(1)7(0)9(1)5(0)
Групповой код В	:	(8,12) (25,27) (35,43)

**Рис. 2.11.** При групповом кодировании последовательности повторяющихся значений 0 или 1 представляются в более кратком виде. В некоторых задачах этот метод обеспечивает эффективное сжатие изображений.

Групповое кодирование будет использоваться в некоторых алгоритмах, рассматриваемых в данной книге. Этот метод также часто применяется для сжатия изображений в стандартных файловых форматах.

### 2.5.6 Формат PGM: Portable Gray Map

Семейство форматов PBM (Portable Bit Map, переносимый растровый формат) объединяет одни из простейших файловых форматов изображений. В

```

P2
# Пример небольшого изображения размерами 8 x 16 пикселей
# Максимальное значение яркости 192.
# На изображении представлено слово "Hi".
  16 8 192
64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64
64 64 128 128 64 64 64 128 128 64 64 192 192 64 64 64
64 64 128 128 64 64 64 128 128 64 64 192 192 64 64 64
64 64 128 128 128 128 128 128 128 64 64 64 64 64 64 64
64 64 128 128 128 128 128 128 128 64 64 128 128 64 64 64
64 64 128 128 64 64 64 128 128 64 64 128 128 64 64 64
64 64 128 128 64 64 64 128 128 64 64 128 128 64 64 64
64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64

```



**Рис. 2.12.** Текстовый файл (в кодировке ASCII), представляющий полутоновое изображение слова "Hi" в формате PPM. Яркость фона равна 64. Яркость пикселей буквы "H" и нижней части "i" равна 128. Верхняя часть "i" (точка) имеет яркость 192. Слева внизу показано изображение из приведенного файла, напечатанное с помощью одной из доступных утилит преобразования файловых форматов. Справа внизу показано то же самое изображение после сохранения с использованием алгоритма сжатия с потерями.

это семейство входят форматы PBM, PGM и PPM. Заголовок изображения и пиксельные данные хранятся в виде символов ASCII.

На рис. 2.12 приведено содержимое файла, представляющего полутоновое изображение размерами 8 строк на 16 столбцов с максимальным значением яркости 192. На рисунке также показаны два варианта графического представления после обработки исходного текстового файла утилитой преобразования форматов изображений. Увеличенное изображение размерами 32 × 64 пиксела в левом нижнем углу было получено дублированием пикселей исходного изображения. Изображение в правом нижнем углу сформировано путем преобразования изображения, показанного слева, в формат JPG со сжатием с потерями. В начале файла PGM хранится *магическое значение*, указывающее, в каком именно виде представлены данные изображения. В нашем примере это "P2" — значения пикселей хранятся в виде ASCII-кодов градаций яркости. Для хранения больших изображений удобнее пользоваться бинарным, а не ASCII-представлением значений пикселей («магическое значение» для бинарного представления равно "P4").

### Упражнение 2.7. Формирование файла изображения в формате PPM

Для представления цветного изображения в файле формата PPM надо указать магическое значение "P3". Для каждого пиксела требуется задавать по три зна-



чения интенсивности (R, G, B). Схема кодирования аналогична черно-белому изображению в файле "P2" (рис. 2.12). С помощью любого доступного текстового редактора сформируйте файл `bullseye.ppm` с изображением 3 концентрических круговых областей различных цветов. Значения интенсивности трех цветов должны указываться последовательно для каждого пиксела. В некоторых других файловых форматах цветовые компоненты пикселей хранятся в виде трех отдельных монохромных изображений. Просмотрите полученное изображение с помощью какой-либо утилиты просмотра графических файлов или Web-браузера.

### 2.5.7 Формат файлов изображений GIF

Формат обмена графическими данными GIF (Graphics Interchange Format) был разработан компанией CompuServe, Inc. Этот формат применяется для хранения огромного количества изображений в World Wide Web и в существующих базах данных. Формат GIF устроен относительно просто, но для кодирования цвета в нем используются только 8-битные числа, поэтому формат не рассчитан на изображения с большим количеством цветов. 256 различных цветов оказывается достаточно для многих изображений, отображаемых на компьютерных дисплеях. В GIF также предусмотрен более компактный вариант представления 16 цветных изображений. В формате GIF для сжатия без потерь применяется метод Лемпела—Зива—Уолша (LZW).

### 2.5.8 Формат файлов изображений TIFF

Теговый формат файлов изображений (Tag Image File Format) был разработан компанией Aldus Corp. TIFF является очень гибким, но то же время сложным форматом. Он используется на всех популярных компьютерных платформах и часто выбирается в качестве основного формата для сканеров. TIFF допускает хранение в одном файле нескольких изображений с цветовым разрешением от 1 до 24 бит на пиксел. В этом формате предусмотрены средства для сжатия с потерями и без потерь.

### 2.5.9 Формат JPEG для хранения фотографий

Формат JPEG (JFIF/JFI/JPG) был предложен относительно недавно Объединенной группой экспертов по обработке фотоизображений (Joint Photographic Experts Group). Основной целью при разработке этого формата ставилось нахождение практичного способа сжатия высококачественных полноцветных неподвижных изображений. Применяемый в JPEG метод кодирования ориентирован на потоковую обработку и позволяет разрабатывать аппаратные устройства для сжатия и распаковки изображений в реальном времени. Изображения могут иметь размеры до 64К × 64К пикселей с цветовым разрешением 24 бита. В одном файле допускается хранение только одного изображения. В

заголовке предусмотрено хранение уменьшенного варианта изображения и до 64К неупакованных байт. Важным преимуществом формата JPEG является независимость от системы представления цвета. (Более подробно цветовые системы рассматриваются в гл. 6.)

Для достижения высокой степени сжатия в JPEG применяется гибкий, но сложный алгоритм сжатия с потерями. Часто он позволяет сжать высококачественное изображение в отношении 20:1 без заметной потери качества. Сжатие хорошо работает на изображениях с большими областями примерно одинакового цвета и в случаях, когда для конечного потребителя не особенно важны высокочастотные изменения в областях с мелкими деталями изображений. (В JPEG также предусмотрена редко используемая возможность использовать сжатие без потерь, позволяющее достичь степени сжатия порядка 2:1 с применением алгоритма кодирования с предсказанием.) Алгоритм сжатия с потерями формата JPEG основан на применении *дискретного косинусного преобразования* (обсуждается в гл. 5), после которого выполняется *кодирование Хаффмана* (в этой книге не рассматривается). Формат JPEG не предназначен для хранения видеоданных.

---

### Упражнение 2.8

Найдите на своем компьютере пакет для просмотра файлов изображений. (Может быть, достаточно будет щелкнуть мышью на пиктограмме файла изображения.) Получите два изображения — одно с лицом человека, другое с природным ландшафтом. Оба изображения должны быть высококачественными, например,  $800 \times 600$  цветных пикселей. Подобные изображения можно получить с помощью планшетного сканера или цифровой камеры. Преобразуйте изображения в несколько различных форматов — GIF, TIFF, JPEG и др. Запишите размер сжатых файлов изображений в байтах и обратите внимание на качество изображения. При оценке качества обратите внимание на вид сцены в целом и на точность представления мелких деталей.

---

### Упражнение 2.9. Изучение формата JPEG\*

(а) Ознакомьтесь с описанием алгоритма сжатия JPEG, в котором выполняется обработка фрагментов изображения размерами  $8 \times 8$  пикселей. (б) Реализуйте и протестируйте алгоритм дискретного косинусного преобразования в варианте сжатия без потерь (за исключением возможной ошибки округления). (в) С помощью какого-либо готового графического пакета сформируйте файл изображения с применением сжатия с потерями. (д) Восстановите фрагмент исходного изображения размерами  $8 \times 8$  с использованием 64 коэффициентов, сгенерированных алгоритмом сжатия с потерями. Сравните значения пикселей со значениями в исходном фрагменте  $8 \times 8$ .

### 2.5.10 Язык PostScript

Форматы семейства BDF/PDL/EPS позволяют хранить данные изображений с использованием печатаемых символов таблицы ASCII. Эти форматы применяются при экранном отображении под управлением оконной системы X11 и в некоторых принтерах. Языки PDL (язык описания страниц) и EPS (инкапсулированный PostScript, язык, изначально разработанный компанией Adobe) часто используются для хранения графики и изображений с целью последующей вставки в другие документы. Значения пикселей кодируются 7-битными кодами ASCII, так что эти файлы можно просмотреть и отредактировать с помощью текстового редактора. Формат PostScript рассчитан на хранение полутоновых и цветных изображений с разрешением от 75 до 3000 точек на дюйм. В новых версиях предусмотрено использование алгоритма сжатия JPEG. В заголовке файлов PDL хранятся координаты описывающего прямоугольника изображения, задающие положение изображения на странице документа. Большинство изображений для этой книги были подготовлены в формате EPS.

### 2.5.11 Файловый формат MPEG для хранения видеоданных

Форматы MPEG (MPG/MPEG-1/MPEG-2) представляют собой потоковые методы сжатия видео-, аудиоданных, текста и графики. Аббревиатура MPEG является сокращением от **M**otion **P**icture **E**xperts **G**roup — Экспертная группа по обработке движущихся изображений. В состав этой интернациональной группы экспертов входят представители промышленных и правительственных организаций. Семейство стандартов MPEG быстро развивается вместе с компьютерными и телекоммуникационными технологиями.

Формат MPEG-1 в основном предназначался для мультимедиа-компьютеров и обеспечивал передачу сжатых аудиоданных со скоростью 0.25 Мбит/с и сжатых видеоданных со скоростью 1.25 Мбит/с. Подобные скорости передачи приемлемы для воспроизведения мультимедиа-данных на популярных персональных компьютерах, но недостаточны для передачи данных телевизионного качества.

В стандарте MPEG-2 предусмотрена передача данных со скоростью до 15 Мбит/с, достаточной для обеспечения телевизионного качества. В схеме сжатия сочетаются преимущества методов, учитывающих пространственную избыточность изображений (как в JPEG), и методов, использующих временную избыточность. Типичный коэффициент сжатия составляет порядка 25:1, хотя достижимы значения до 200:1.

Под *временной избыточностью* видеоданных подразумевается, что многие области кадров в видеопоследовательности при переходе от кадра к кадру сильно не меняются. Поэтому в алгоритме сжатия можно выполнять кодирование изменений и даже предсказание некоторых кадров видеопоследовательности по содержанию предыдущих и последующих кадров. (В будущих версиях MPEG

также будут предусмотрены способы кодирования распознанных объектов и использование программных методов генерации образов этих объектов.) Качество видеоданных после сжатия зависит от времени, затраченного на выполнение сжатия.

Формат Motion JPEG построен по гибридной схеме, в которой к отдельным кадрам видеопоследовательности применяется алгоритм сжатия JPEG, а временная избыточность не учитывается. Хотя сжатие и распаковка данных в формате Motion JPEG выполняется просто, коэффициент сжатия невелик. Эффективность использования памяти и коммуникационных каналов ниже, чем в формате MPEG.

Использование векторов движения для сжатия видеоданных в формате MPEG описывается в гл. 9.

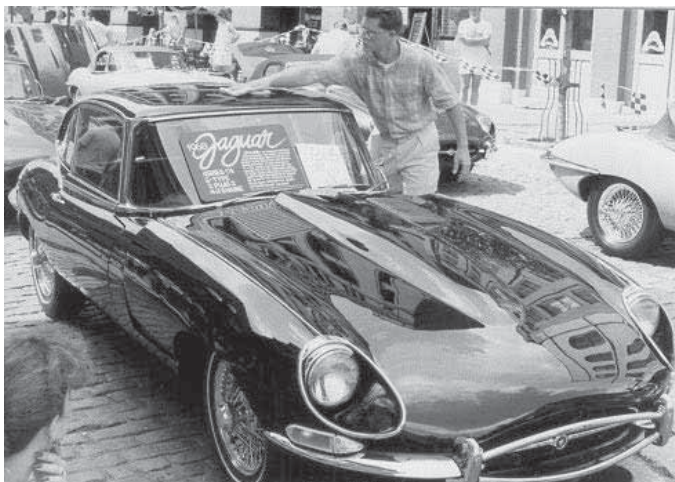
### 2.5.12 Сравнение форматов файлов изображений

В табл. 2.1 приведены данные для сравнения нескольких популярных форматов изображений по критерию файлового объема. В первом столбце указано название формата. Во втором столбце приведен размер файла с маленьким полутоновым изображением слова "Hi" (размер  $8 \times 16$  пикселей), а в третьем столбце — размер файла с более крупным цветным изображением размерами  $347 \times 489$  пикселей (Cars.TIF, рис. 2.13). Выполняя преобразования между файловыми форматами в различной последовательности, можно для одного и того же изображения получить файлы различного размера.

Например, изображение с автомобилем "Cars" TIF (рис. 2.13) было получено с помощью сканера. Объем файла составлял 509,253 байт. После преобразования в 256-цветный формат GIF был получен файл размером 138,267 байт. Затем из него был сформирован TIF-файл объемом 171,430 байт. В этом TIF-файле

**Таблица 2.1.** Размеры файлов (в байтах) с одними и теми же изображениями, сохраненными в различных форматах. Это полутоновое изображение слова "Hi" размерами  $8 \times 16$  пикселей (рис. 2.12) и цветное изображение автомобиля размерами  $347 \times 489$  (рис. 2.13).

Формат файла изображения	Кол-во байт в изобр. "Hi"	Кол-во байт в изобр. "Cars"
PGM	595	509,123
GIF	192	138,267
TIF	918	171,430
PS	1,591	345,387
HIPS	700	160,783
JPG (без потерь)	684	49,160
JPG (с потерями)	619	29,500



**Рис. 2.13.** Сложная сцена с большим количеством дальностных признаков, облегчающих зрительное восприятие глубины объектов.

пиксели представлены значениями с меньшим количеством бит, чем в первоначальном TIF-файле, полученном с помощью сканера. Но при отображении на дисплее существенной разницы между ними не обнаруживалось. Размер файла в формате JPEG составил всего треть от TIF-файла при аналогичном видимом качестве. JPEG со сжатием с потерями стал абсолютным победителем по объему файла. За этот результат приходится расплачиваться большими вычислительными затратами на распаковку изображений. В некоторых задачах для распаковки изображений в реальном времени может потребоваться специальное аппаратное обеспечение.

## 2.6 ОСОБЕННОСТИ И ПРОБЛЕМЫ ПОЛУЧЕНИЯ ИЗОБРАЖЕНИЙ ЕСТЕСТВЕННЫХ СЦЕН

Даже короткая прогулка на природе подтверждает огромное многообразие зрительно наблюдаемого окружающего мира, не перестающее служить источником вдохновения художников. Это многообразие обогащает человеческий опыт, но создает проблемы в машинном зрении (например, см. рис. 2.13). Интенсивность и цвет точки изображения сложным образом зависят от материала, геометрических свойств объектов и освещения. Важны не только свойства материала, но и ориентация объекта относительно датчика изображений, источников света и других объектов. В естественных сценах очень часто наблюдаются такие эффекты, как зеркальные блики на блестящих поверхностях, тени, взаимное отражение, прозрачные объекты.

В ряде задач при распознавании многих поверхностей и объектов цветочные признаки имеют меньшее значение, чем признаки, связанные с формой

или текстурой (эти характеристики зависят не от одного, а от совокупности пикселей). Некоторые задачи, в которых возможности управления наблюдаемыми сценами малы или отсутствуют (например, мониторинг транспортных потоков), оказываются особенно интересными и сложными.

Ряд проблем при получении изображений остаются даже в специально организованной промышленной обстановке или в телевизионных студиях. Как будет показано в гл. 6, при освещении блестящего металлического цилиндра точечным источником света интенсивность отраженного излучения может меняться в диапазоне от 1 до 100000. Большинство датчиков в таком динамическом диапазоне не работают. Солнечный или искусственный свет нагревает поверхности объектов, так что состав излучения от этих объектов с течением времени будет меняться и изображения, полученные с помощью ПЗС-камер, могут оказаться ярче (из-за чувствительности ПЗС-датчиков к инфракрасному излучению). Или, например, этот эффект может привести к тому, что после отправления самолетов на взлет на месте их стоянки на изображениях останутся тени, объясняемые наличием более холодных областей. Некоторые операции получения изображений упрощаются при использовании управляемого подсвета на основе монохромного лазера. Но и в этом случае излучение источника может полностью поглощаться некоторыми поверхностями, а на некоторых место освещения поверхности лазером будет сложно найти из-за вторичных отражений.

Во многих приложениях автоматизации ряд описанных проблем могут быть решены инженерным путем. Для исключения излучения ненужного в данной задаче диапазона датчик может быть снабжен фильтром. Например, поврежденные сдавленные темно-красные вишни лучше заметны при использовании фильтра, пропускающего только инфракрасное излучение. Движущиеся объекты, дающие размытые изображения при постоянном освещении, можно осветить *стробоскопическим источником света*, включающимся на очень короткий промежуток времени. При этом на изображении, генерируемом чувствительным датчиком, движущиеся объекты будут казаться неподвижными.

В ряде задач проведение поверхностных измерений и контроль формы объектов упрощаются с помощью *структурированного света*. Например, при освещении лопатки турбины точно чередующимися полосами красного и зеленого света многие поверхностные дефекты будут хорошо заметны на двумерном изображении в виде разрывов световых полос.

Перечисленные методы будут рассматриваться в различных разделах данной книги.

## **2.7 ОЦЕНКА ПРОСТРАНСТВЕННЫХ СВОЙСТВ ОБЪЕКТОВ ПО ДВУМЕРНЫМ ИЗОБРАЖЕНИЯМ**

При восприятии пространственной структуры трехмерного окружающего мира в зрительной системе человека учитываются дальностные признаки (или признаки глубины объектов) нескольких типов. Здесь будет приведено толь-

ко краткое качественное описание. Количественные модели многих из этих признаков построил специалист по когнитивной психологии Дж. Дж. Гибсон (J. J. Gibson). В 1980-х гг. реализация и эксперименты с этими моделями интенсивно проводились многими исследователями в области компьютерного зрения. Некоторые из них будут обсуждаться в книге далее.

При получении изображения формируются сложные взаимосвязи между трехмерной структурой физического мира и двумерной структурой изображения. Напомним, что при формировании изображения выполняется перспективное проектирование (модель перспективной проекции была приведена на рис. 2.2), и обратимся к рис. 2.13. *Промежуточное расположение*, возможно, является наиболее важным дальностным признаком: объекты, расположенные ближе к наблюдателю, частично заслоняют более далекие объекты. Распознавание перекрытий позволяет оценить относительную дальность объектов. Если человек виден в области образа стены, то очевидно, что человек находится ближе к датчику, чем стена. Человек, обнаруженный позади автомобиля, находится дальше от датчика, чем автомобиль.

Важным дальностным признаком также является относительный размер. Изображение автомобиля на расстоянии 20 м от датчика будет намного меньше, чем изображение автомобиля на расстоянии 10 м, даже если абсолютные размеры первого автомобиля больше. На большом расстоянии автомобили кажутся крошечными и медленно движущимися; но на основе опыта нам известна связь размеров и скорости автомобилей с расстоянием. Уходящие вдаль железнодорожные рельсы кажутся нам пересекающимися в некоторой далекой точке (в *точке схода*), хотя нам известно, что в действительности рельсы находятся в пространстве на постоянном расстоянии друг от друга.

Образ открытой в комнату прямоугольной двери на сетчатке глаз имеет форму трапеции, а не прямоугольника. Более далекий край двери кажется короче ближнего. Этот эффект *ракурса* при перспективном проектировании позволяет нам оценить пространственную ориентацию двери.

Связанным с восприятием дальности признаком также является *текстурный градиент*. Видимая текстура поверхностей зависит от расстояния до наблюдателя и от ориентации поверхностей. В парке мы можем заметить отдельные стебли травы или листья клена, в то время как издали видим только области зеленого цвета. *Текстурным градиентом* называется изменение текстуры изображения поверхности в зависимости от расстояния до наблюдателя вследствие перспективного проектирования при формировании изображения.

Более подробное описание упомянутых вопросов приведено в гл. 12.

---

### Упражнение 2.10. Опыт художественного восприятия

Рассмотрите естественные сцены в различных окружающих условиях и схематично изобразите наблюдаемые дальностные признаки, которые были описаны в разд. 2.7. Например, попробуйте обнаружить их в обстановке кафе или на

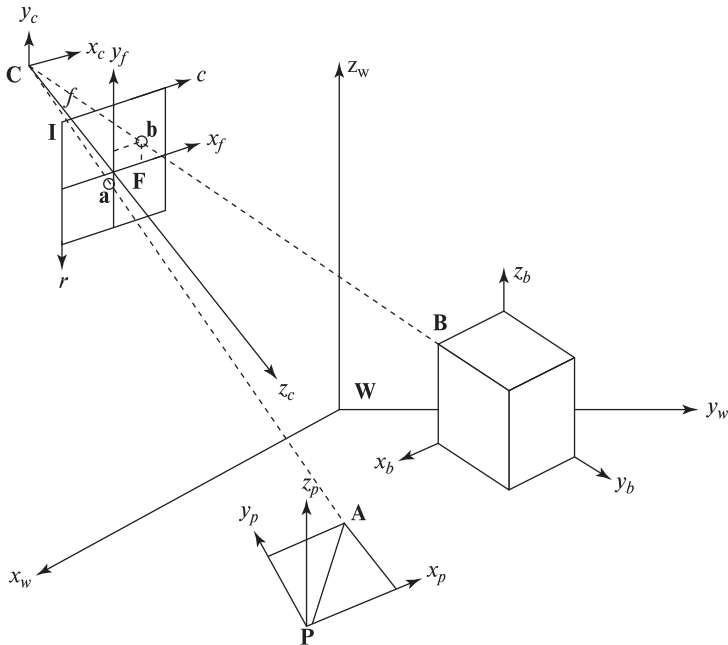


углу городской улицы, наблюдаемой с высоты нескольких этажей, или среди лесных пейзажей.

## 2.8 ПЯТЬ СИСТЕМ КООРДИНАТ

*Системы координат* необходимы и для качественного, и для количественного анализа трехмерных сцен. В общем случае в задачах анализа трехмерных сцен используются пять различных систем координат. Примерами решаемых задач являются управление роботизированным производственным модулем, снабженным зрительной системой, или формирование виртуальной трехмерной среды для общения людей.

Некоторые системы координат используются не только в робототехнике, но также оказываются важными для понимания особенностей человеческого пространственного восприятия психологами. Пять различных систем координат приведены на рис. 2.14. В действительности, показаны шесть систем координат, так как в сцене присутствуют два различных объекта — блок и пирамида, с каждым из которых связана собственная объектная система координат. В четырех из этих систем координаты являются действительными числами,



**Рис. 2.14.** Пять систем координат, которые используются при анализе трехмерных сцен: мировая система координат  $W$ , объектная  $O$  (для пирамиды  $O_p$  и  $O_b$  для блока), система координат камеры  $C$ , действительная система координат изображения  $F$  и пиксельная система координат изображения  $I$ .



задающими положение относительно непрерывных осей. В системе координат изображения координаты принимают целые значения, служащие индексами в массиве пикселей изображения. В нескольких приведенных далее примерах рассматривается сцена игры в бейсбол, наблюдаемая с помощью телевизионной камеры. Объектами этой сцены являются игроки, базы, мячи, биты и т. п.

### 2.8.1 Пиксельная система координат **I**

В пиксельном массиве каждая точка имеет целочисленные пиксельные координаты. На рис. 2.14 вершина пирамиды **A** проектируется на пиксел  $\mathbf{a} = [a_r, a_c]$ , где  $a_r$  и  $a_c$  являются целочисленными номерами строки и столбца, соответственно. Многие характеристики сцены можно определить непосредственным анализом изображения в терминах строк и столбцов пикселей.

Например, рассмотрим подъемно-транспортный робот или другой транспортный механизм, который всегда помещает блок (наподобие коробки порошка в прачечной) в одном и том же положении напротив камеры. Пометки на фронтальной поверхности блока можно найти, анализируя только изображение в виде матрицы из строк и столбцов пикселей. Располагая двумерным изображением в примере с бейсбольной игрой, например, можно определить наличие у игрока черной биты. Однако если, кроме изображения **I** нет никакой другой информации, то нам не удастся определить, какой объект в пространстве имеет наибольшие размеры или движутся ли объекты пересекающимися курсами, ведущими к столкновению.

### 2.8.2 Система координат объекта **O**

Система координат объекта (называется также объектной или модельной системой координат) используется в компьютерной графике и в компьютерном зрении для описания моделей идеальных объектов.

На рис. 2.14 показаны две объектные системы координат, одна для блока **O<sub>b</sub>**, и одна для пирамиды **O<sub>p</sub>**. Координаты угловой точки блока **B** относительно объектной системы координат равны  $[x_b, 0, z_b]$ . Эти координаты не зависят от положения блока относительно других объектов сцены (т. е. системы координат **W**).

Например, в задаче контроля формы объектная система координат может быть необходима для проверки наличия отверстия в соответствующем месте относительно других отверстий или углов объекта.

### 2.8.3 Система координат камеры **C**

Система координат камеры **C** часто применяется для описания *камероцентрического* вида. Например, чтобы определить, будет ли объект располагаться прямо напротив датчика, двигаться вдаль, и т. п. Если изображение мяча формируется в центре сетчатки вашего глаза и продолжает увеличиваться в размерах, вероятно, этот мяч попадет в вас. Человек или робот со зрительной

системой является одновременно и объектом, и датчиком (датчиками), так что объектная и датчиковая система координат могут быть почти, но не точно, одинаковыми. (Задевали ли вы когда-нибудь дверной проем, даже если казалось, что сможете пройти сквозь него, не касаясь?) В программах трехмерной компьютерной графики часто предусмотрена возможность выбора пользователем различных видов камеры для просмотра трехмерной сцены.

#### 2.8.4 Действительная система координат изображения $\mathbf{F}$

В системе координат камеры значения координат являются действительными числами и обычно измеряются в тех же единицах, что и мировые координаты — например, в мм или дюймах (в т. ч. *дальностная координата*, или, *глубина*  $z_c$ ). Пространственная точка проектируется на действительную плоскость изображения в точку с координатами  $[x_f, y_f, f]$  ( $f$  — фокусное расстояние). Координаты  $x_f$  и  $y_f$  не являются индексами пикселей в массиве изображения. Их значения зависят от размеров пикселя и пиксельного положения точки пересечения оптической оси с плоскостью изображения. На рис. 2.14 обе действительные координаты точки изображения  $\mathbf{a}$  в системе координат  $\mathbf{F}$  имеют отрицательные значения. В системе координат  $\mathbf{F}$  определена функция интенсивности, которая дискретизируется в процессе формирования цифрового изображения в пиксельном массиве  $\mathbf{I}$ .

#### 2.8.5 Мировая система координат $\mathbf{W}$

Мировая система координат  $\mathbf{W}$  нужна для описания взаимного расположения объектов в пространстве. В роботизированном модуле или среде виртуальной реальности передача координатных данных между двигателями и датчиками часто выполняется в мировых координатах. Например, датчик изображений сообщает роботу, откуда надо взять болт и куда его нужно закрутить.

Геометрические и математические взаимосвязи между перечисленными системами координат будут очень важны для последующего материала данной книги. Однако в нескольких следующих главах будут рассматриваться исключительно вопросы обработки информации пиксельного массива в предположении, что между изображением и реальной сценой существует однозначное априорно известное соответствие.

Читатель, которому необходимо работать с выражениями перспективного преобразования или выполнить оценку эффекта масштабирования при таком преобразовании, может обратиться к гл. 12 и ознакомиться с подробным описанием модели перспективной проекции при формировании изображений.

### 2.9 ДРУГИЕ ТИПЫ ДАТЧИКОВ\*

В данный раздел включены описания еще нескольких типов датчиков изображений. При первом чтении читатель может пропустить данный раздел, если

датчики описываемых типов не интересуют его и не используются в текущих прикладных задачах. Технология производства датчиков изображений быстро развивается. В будущем можно ожидать не только появления новых типов датчиков, но и улучшения характеристик существующих устройств.

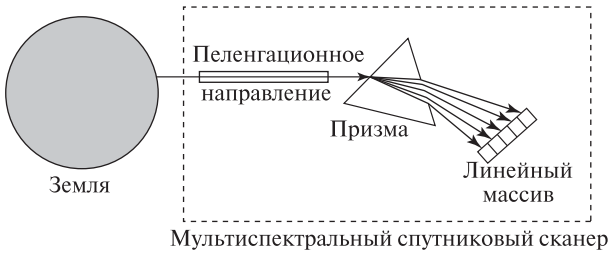
### 2.9.1 Микроденситометр\*

Для сканирования слайда или фотопленки можно воспользоваться точечным датчиком и освещением *на просвет*. Сканируемый кадр пленки помещается между источником света и датчиком. Датчик записывает значение, пропорциональное оптической плотности материала пленки в небольшой окрестности  $[r, c]$ . При сканировании необходимо очень точно механически перемещать прямоугольный кадр в поле зрения точечного датчика. Преимущество использования одного точечного датчика по сравнению с ПЗС-матрицей заключается в том, что у точечного датчика отсутствуют ошибки измерения интенсивности, присущие элементам ПЗС-матрицы по технологическим причинам. Еще одним преимуществом является возможность получения изображений с большим количеством строк и столбцов, чем с помощью ПЗС-матрицы. Однако такой прибор оказывается медленным и неудобным для промышленного применения.

Для читателя может оказаться интересным похожий способ сканирования, использовавшийся в первых исследованиях по обработке изображений. Он применялся А. Розенфельдом (Azriel Rosenfeld) в 1970-х гг. для получения большого количества изображений в лабораторных условиях для последующей компьютерной обработки. Черно-белые изображения, напечатанные на бумаге, накручивались на стальной цилиндр. Обычно сканировались отдельные изображения или наборы изображений общей площадью порядка  $23 \times 23$  см. Цилиндр закреплялся в токарном станке, который прокручивал все участки изображения перед небольшим светодиодом и датчиком, измеряющим интенсивность отраженного света от каждого участка. При каждом обороте цилиндра получалась строка из 3600 пикселей, которая сохранялась на магнитной ленте. Скорость записи на ленту была синхронизирована со скоростью вращения токарного станка. Окончательный файл на ленте представлял изображение размерами  $3600 \times 3600$  пикселей. Если файл содержал несколько изображений, то затем они разделялись программным образом.

### 2.9.2 Цветные и мультиспектральные изображения\*

Так как человеческий глаз содержит отдельные рецепторы для восприятия светового излучения в различных спектральных диапазонах, то его можно назвать *мультиспектральным* датчиком. В некоторых цветных ПЗС-камерах перед ПЗС-матрицей установлена тонкая преломляющая пленка. Эта пленка разделяет падающий световой луч на 4 луча, попадающие на 4 соседних элемента ПЗС-матрицы. Получаемое цифровое изображение можно рассматривать как набор из четырех чередующихся цветных изображений, соответ-



**Рис. 2.15.** Направленный мультиспектральный сканер на борту космического спутника. Излучение от некоторого участка земной поверхности разлагается призмой в спектр.

ствующих по-разному преломленным световым компонентам с разной длиной волны. Получение дополнительной спектральной информации достигается ценой потери пространственного разрешения.

В других конструкциях цветных ПЗС-камер используется разноцветный диск, который вращается на пути падающего света. Диск изготовлен из прозрачной пленки с равными по площади секторами, окрашенными в различные основные цвета. Скорость вращения диска синхронизирована с временными промежутками чтения информации с ПЗС-матрицы. В течение одного промежутка перед матрицей двигается только красный сектор диска, затем синий, а затем зеленый. Содержимое ПЗС-массива в течение одного поворота разноцветного диска считывается трижды. При этом получают три отдельных изображения. В данном случае повышение цветовой чувствительности приводит к уменьшению скорости ввода. Точка быстро движущегося объекта в процессе получения трех независимых изображений может отобразиться на различные пиксели плоскости изображения.

На некоторых спутниках установлены сканеры для направленного зондирования поверхности Земли. Каждый участок земной поверхности осматривается направленной оптической системой так, что в течение некоторого временного промежутка собирается все излучение с этого участка, а излучение остальных участков маскируется. На рис. 2.15 показано, что излучение от наблюдаемого участка сначала проходит через призму. Призма разлагает это излучение в спектр, который проецируется на линейный ПЗС-массив. (Излучение более коротких длин волн преломляется призмой сильнее, чем более длинных.) Подобный датчик дискретизирует интенсивность излучения одновременно в нескольких используемых спектральных диапазонах. На рис. 2.15 показан спектр, который воспринимается датчиком с чувствительностью в 5 спектральных диапазонах. В результате пиксел представляется вектором из 5 значений интенсивности  $[b_1, b_2, b_3, b_4, b_5]$ . Двумерное изображение формируется путем изменения пеленгационного направления или с помощью сканирующего зеркала. Это позволяет получить пиксели одной строки изображения. Другие строки изображения формируются в процессе орбитального движения спутника вокруг Земли.

Результирующее изображение подвержено искажениям, связанным с движением спутника. Множество всех сканированных участков образует на земной поверхности трапециевидальную область. Эту область можно восстановить по прямоугольному цифровому изображению с помощью методов деформации изображений, описанных в гл. 11. Данные об интенсивности излучения в различных спектральных диапазонах позволяют, например, классифицировать содержимое каждого участка земной поверхности — находится ли на нем вода, лес, асфальт и т. п.

### 2.9.3 Рентгеновские изображения\*

Источники и датчики рентгеновского излучения используются для получения изображений самых разнообразных объектов — частей человеческого тела, сварных швов труб, банок с яблочным пюре и т. д. Датчики фиксируют данные о распределении интенсивности рентгеновского излучения в плоскости изображения после прохождения сквозь наблюдаемые объекты. Наблюдаемые объекты располагаются между источником рентгеновского излучения и плоскостью изображения (что напоминает получение изображений с помощью микроденситометра).

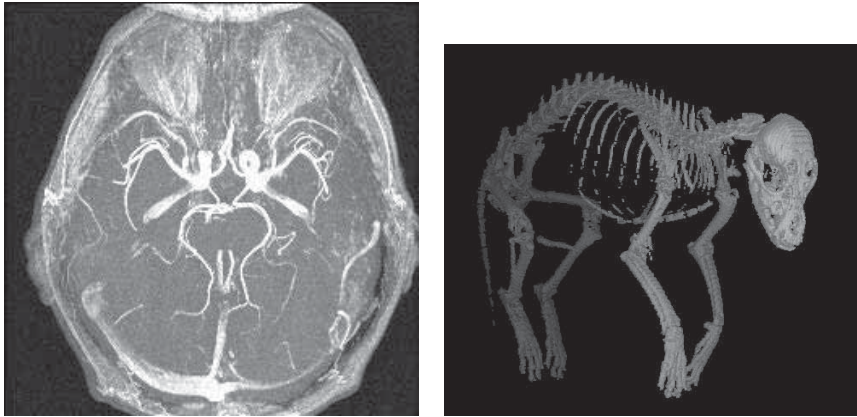
Низкая интенсивность излучения в некоторой точке изображения свидетельствует о том, что излучение источника в направлении соответствующего луча было поглощено в результате прохождения через более плотное вещество. На практике для получения двумерных рентгеновских фотографий широко используются чувствительные к рентгеновским лучам фотопленки.

Для получения трехмерных данных применяются рентгеновские томографы. В них используются специальные математические методы построения трехмерных массивов значений плотности вещества наблюдаемого объекта по результатам измерений интенсивности рентгеновских лучей, проходящих сквозь наблюдаемый объект по многим различным направлениям. Справа на рис. 2.16 представлено двумерное компьютерное изображение собаки, построенное по данным рентгеновской томографии. На этом изображении показаны воксели, соответствующие фрагментам более плотных тканей. Визуально эти воксели напоминают непрозрачные отражающие поверхности, наблюдаемые с некоторой точки. При проведении диагностики для построения двумерной проекции полученных трехмерных данных специалист может указывать любое направление.

---

#### Упражнение 2.11

В стоматологии рентгенография используется для получения изображений зубов. Как будут выглядеть на них (как яркие или темные области) плотная зубная ткань и более мягкие каверны?



**Рис. 2.16.** (Слева) изображение максимальной интенсивности, построенное путем проектирования пикселей наибольшей яркости из множества слоев человеческой головы, полученных методом магниторезонансной ангиографии. (Справа) компьютерное изображение вокселей, представляющих фрагменты плотных тканей в наборе рентгенографических изображений. Воксели выглядят как множество освещенных поверхностных элементов. (С любезного разрешения Theresa Bernardo.)

### 2.9.4 Получение изображений методом ядерного магнитного резонанса (ЯМР)\*

На основе эффекта ядерного магнитного резонанса (ЯМР) разработаны устройства для получения трехмерных изображений объектов, состоящих из различных веществ. Эти устройства — ЯМР-томографы — обычно применяются в медицине для диагностики различных заболеваний.

Получаемые данные образуют массив  $I[s, r, c]$ , где  $s$  является номером слоя (некоторого поперечного сечения объекта), а  $r$  и  $c$ , как обычно, представляют собой индексы пиксела на изображении слоя  $s$ . Каждый малый элемент объема, называемый *вокселом*, соответствует небольшой пространственной области (например, сферической области диаметром 2 мм). Для каждого воксела измеряется интенсивность сигнала ЯМР, зависящая от химических свойств вещества в соответствующей пространственной области.

В магниторезонансной ангиографии интенсивность пикселей изображений зависит от скорости вещества (потока крови) в соответствующих вокселях. Стоимость ЯМР-томографов составляет миллионы долларов, а один сеанс сканирования обходится в тысячи долларов. Но получаемые результаты оказываются очень полезными для диагностики многих заболеваний. Устройства на основе ЯМР могут использоваться для обнаружения внутренних дефектов фруктов и овощей, но применение в этих целях будет возможно только при условии уменьшения стоимости оборудования.

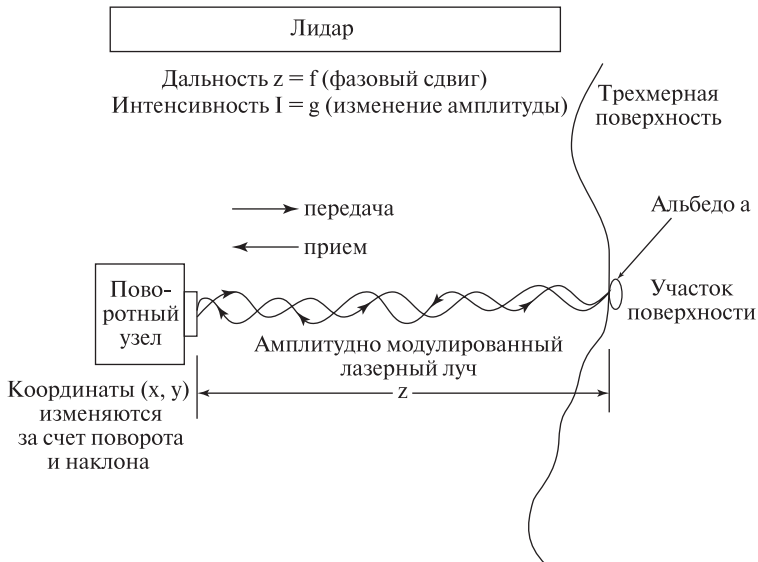
На рис. 2.16 (слева) показано цифровое изображение, построенное по трехмерным ЯМР-данным. Это изображение  $MIP[r, c]$  называется *проекцией*

максимальной интенсивности. Оно может быть построено путем выбора наиболее яркого вокселя  $I[s, \mathbf{r}, \mathbf{c}]$  среди всех слоев  $s$  при фиксированных значениях координат  $\mathbf{r}$  и  $\mathbf{c}$ . Программные алгоритмы генерации таких изображений способны выполнять проектирование изображений всех слоев вдоль любого заданного направления наблюдения. Традиционно при установлении медицинского диагноза необходимые двумерные изображения распечатываются и закрепляются на стене, причем изображений может потребоваться достаточно много. В настоящее время разработаны ряд моделей объемных дисплеев и рентгенологи постепенно начинают пользоваться ими.

### 2.9.5 Дальномеры и дальнометрические изображения\*

Существуют устройства для измерения не только интенсивности излучения элементов трехмерных поверхностей, но и расстояния до них (глубины). На дальнометрическом (дальнономном) изображении непосредственно представлены данные о форме поверхностей. На обычных изображениях, состоящих из значений интенсивности, для оценки формы поверхности необходимо выполнить обработку с применением достаточно сложных и чувствительных к ошибкам измерений методов.

На рис. 2.17 схематично показан принцип работы лазерного дальномера (лидар, LIDAR — это сокращение от Light Detection And Range). В состав этого устройства входит лазерный излучатель с амплитудной модуляцией. Лидар освещает участок трехмерной поверхности лазерным лучом и воспринимает

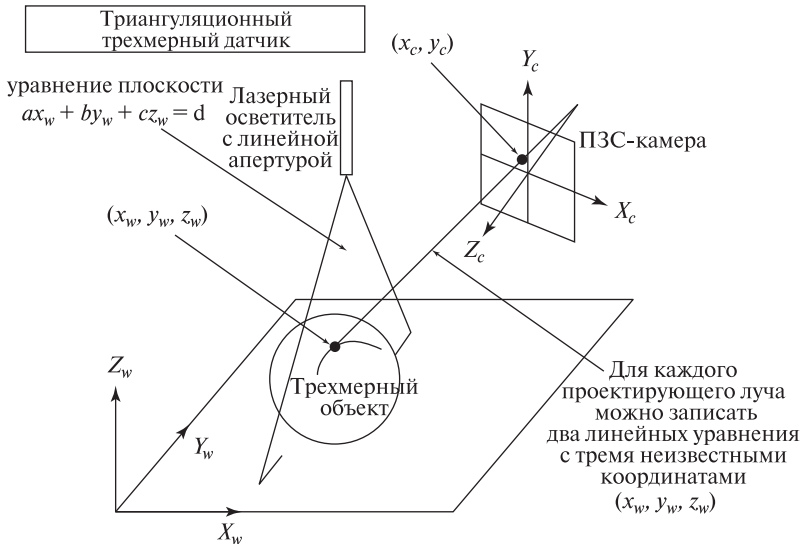


**Рис. 2.17.** Лидар — датчик, позволяющий получать пиксели с информацией и о дальности, и об интенсивности излучения участков поверхности.

отраженный сигнал. Измерение расстояния основано на сравнении фазы излученного и отраженного сигнала (фазового сдвига). Лидар измеряет расстояния в единицах длины волны сигнала, используемого для модуляции лазерного излучения. Этот метод работает только на расстояниях, соответствующих одному периоду модулирующего сигнала. На более далеких расстояниях возникает неоднозначность — участок на расстоянии  $d+n\lambda/2$  будет давать тот же отклик, что и участок на расстоянии  $d$ , где  $\lambda$  — длина волны модулирующего сигнала.

Путем сравнения интенсивности излученного и отраженного сигнала лидар может вычислить коэффициент отражения участка поверхности на длине волны лазера. Таким образом, лидар формирует два совмещенных изображения — одно дальностное и одно полутоновое. Лидар работает медленнее ПЗС-камеры, так как на вычисление фазового сдвига для каждого воспринимаемого участка требуется некоторое время задержки. Лидар также намного дороже, так как для управления лазерным лучом обычно применяются механические узлы. В ряде задач высокая цена оправдана, например, в горнодобывающей робототехнике и в космических роботах, исследующих различные тела Солнечной системы.

Для измерения трехмерных координат также применяется разновидность метода триангуляции, известного уже более 5000 лет. Источник света, показанный на рис. 2.18, имеет линейную апертуру. Свет от него распространяется в некоторой плоскости. На пересечении этой плоскости с поверхностью объекта образуется яркая линия. Эта линия должна быть видна на обрабатываемом



**Рис. 2.18.** Датчик с использованием структурированного подсвета с линейной апертурой предназначен для вычисления трехмерных координат точек триангуляционным методом.



изображении. Каждая яркая точка линии на изображении  $[x_c, y_c]$  является образом соответствующей освещенной трехмерной точки  $[x_w, y_w, z_w]$ .

Для триангуляционного датчика на рис. 2.18 известно расположение световой плоскости и проектирующего луча, проходящего через центр камеры и яркую точку изображения. Очевидно, что этот луч будет пересекаться со световой плоскостью в единственной точке. Ее координаты  $x_w, y_w, z_w$  можно определить с помощью аналитической геометрии. Одно уравнение с тремя неизвестными координатами можно записать на основе уравнения световой плоскости, и два уравнения — на основе уравнения прямой, содержащей проектирующий луч. Искомые координаты трехмерной точки вычисляются посредством решения полученной системы трех линейных уравнений. В гл. 13 описываются методы калибровки, позволяющие получить необходимые уравнения по результатам нескольких измерений на калибровочном стенде.

Вышеприведенные рассуждения упрощаются, если рассматривать проектирование на поверхность объекта отдельного светового луча, а не световой плоскости. В процессе сканирования сцены необходимо обеспечить прослеживание световой плоскостью или лучом всего видимого объема. Для этого могут применяться сканирующие зеркальные устройства. Или, наоборот, может быть предусмотрено перемещение объектов сцены сквозь освещаемый участок с помощью какой-либо конвейерной системы.

Автоматы с использованием структурированного подсвета с несколькими световыми плоскостями применяются, например, в автомобильной промышленности для контроля установки колес и дверей. Когда наблюдаемая сцена специально организована так, что в ней присутствуют известные объекты в заданных положениях, то в процессе обработки изображений может понадобиться только удостовериться в близости заданной световой полосы к некоторому идеальному положению. Поток данных от системы компьютерного зрения применяется, например, для управления производственными операциями, для контроля качества, формирования различных проверочных отчетов и т. п.

Существует много различных конструкций триангуляционных датчиков. Многие из них описаны в дополнительной литературе. При выборе датчика обычно приходится учитывать особенности решаемой прикладной задачи.

## 2.10 ЛИТЕРАТУРА

Более подробная информация о конструкции устройств для формирования изображений содержится в книге Schalkoff (1989). Учебные материалы и технические спецификации приборов с зарядовой связью легко можно найти с помощью поисковых механизмов Web. Примером подобных учебных материалов является страница [www.mrsec.wisc.edu/edetc/ccd.html](http://www.mrsec.wisc.edu/edetc/ccd.html) на сайте Университета штата Висконсин. Одной из первых статей о разработке цветных ПЗС-камер является статья [Dillon и др., 1978]. Обсуждение и моделирование многих оптических явлений рассматривается в книге [Hecht, Zajac, 1976].

В книге психолога Дж. Дж. Гибсона [J. J. Gibson, 1950] описано много фундаментальных экспериментов по тематике зрительного восприятия, которые привели к разработке ряда методов компьютерного зрения. Особенности зрительных систем животных и человека с инженерной точки зрения рассматриваются в книге [Levine, 1985]. Книга [Nalwa, 1993] начинается в обсуждения способностей и недостатков зрительной системы человека и содержит хорошее качественное описание процессов формирования изображений и перспективного преобразования. В статье [Margaret Livingstone, 1988] дают популярное рассмотрение человеческого зрительного восприятия произведений искусства. Много математических свойств перспективного преобразования описано во втором томе книги [Haralick, Shapiro, 1992]. Практические детали и общий обзор файловых форматов изображений собраны в энциклопедии [Murray, VanRyper, 1994]. Вместе с ней распространяется компакт-диск, на котором находится набор программных утилит из различных источников.

1. Dillon, P., D. Lewis, and F. Kaspar. 1978. Color imaging system using a single CCD area array. *IEEE Trans. Electron Devices*, ED-25(2):102–107.
2. Gibson, J. J. 1950. *The Perception of the Visual World*. Houghton-Mifflin, Boston.
3. Hecht, E., and A. Zajac. 1974. *Optics*. Addison-Wesley, Reading, MA.
4. Haralick, R., and L. Shapiro. 1992. *Computer and Robot Vision, Volumes I and II*. Addison-Wesley, Reading, MA.
5. Levine, M. D. 1985. *Vision in Man and Machine*. McGraw-Hill, New York.
6. Livingstone, M. 1988. Art, illusion and the visual system. *Sci. Am.* (Jan. 1988) 78–85.
7. Murray, J., and W. VanRyper. 1994. *Encyclopedia of Graphics File Formats*. O'Reilly and Associates, Inc., 103 Morris St., Suite A, Sebastopol, CA 95472. [Имеется русский перевод. Список литературы на русском языке см. в конце книги. — *Прим. перев.*]
8. Nalwa, V. 1993. *A Guided Tour of Computer Vision*. Addison-Wesley, Reading, MA.
9. Schalkoff, R. J. 1989. *Digital Image Processing and Computer Vision*. John Wiley & Sons, New York.

# Анализ бинарных изображений

В ряде приложений, например, при обработке документов и в некоторых задачах промышленного машинного зрения, бинарные изображения используются в качестве входных данных для алгоритмов обработки, разработанных с учетом особенностей решаемой задачи. Эти алгоритмы могут выполнять различные действия — от очень простых, наподобие подсчета каких-либо признаков, до гораздо более сложных действий, связанных с распознаванием, определением местоположения и контролем объектов. Знакомство с обработкой бинарных изображений перед изучением методов обработки полутоновых и цветных изображений поможет читателю быстрее получить общее представление о процессе анализа изображений.

В данной главе описываются основные операции, применяемые в бинарном машинном зрении. Во-первых, на примере простого алгоритма подсчета объектов демонстрируется возможность выполнения практически полезной задачи с помощью несложного алгоритма. Далее рассматривается алгоритм маркировки для выделения связных компонент. Этот алгоритм присваивает уникальную метку каждой связной группе пикселей. Данный алгоритм часто используется на первых стадиях в алгоритмах многоэтапной обработки изображений. Затем описывается набор операторов уточнения и наращивания. Операторы математической морфологии могут применяться для объединения и разделения компонент, для заполнения отверстий и поиска характерных признаков изображения. После изоляции связных компонент для каждой из них может быть вычислен набор важных свойств. Эти свойства используются на более высоких уровнях обработки во многих алгоритмах, например, для распознавания и слежения за движущимися объектами. В данной главе будут определены и рассмотрены многие основные свойства компонент бинарных изображений и алгоритмы для вычисления этих свойств. В заключительной части главы обсуждается проблема автоматического выбора порога бинаризации для преобразования полутоновых или цветных изображений в пригодные для обработки бинарные изображения.

### 3.1 ПИКСЕЛЫ И ОКРЕСТНОСТИ ПИКСЕЛОВ

Для формирования бинарного изображения  $\mathbf{B}$  по данным полутонового или цветного изображения  $\mathbf{I}$  можно выполнить операцию, которая выбирает некоторое подмножество пикселов изображения в качестве пикселов *переднего плана* (foreground pixels). Эти пикселы представляют интерес для решаемой задачи анализа изображений. Остальные пикселы игнорируются как *фоновые* (background pixels).

Операция отбора пикселов может быть простой (например, пороговый оператор, который отбирает пикселы со значениями из заданного яркостного диапазона или цветового подпространства) или представлять собой сложный алгоритм классификации. Пороговая бинаризация будет обсуждаться в конце данной главы, а более сложные операторы отбора будут встречаться далее в различных главах книги.

В начале данной главы будем полагать, что бинарное изображение  $\mathbf{B}$  уже сформировано. Будем рассматривать это изображение в качестве входных данных для выполнения операций анализа изображений. Понятие бинарного изображения иллюстрируется на рис. 3.1 на примере четырех бинарных изображений рукописных символов.



Рис. 3.1. Бинарные изображения рукописных символов.

Пикселы бинарного изображения  $\mathbf{B}$  принимают значения 0 или 1. Будем полагать, что значения 1 соответствуют пикселам переднего плана, а 0 — фоновым пикселам. Значение пиксела на пересечении строки  $\mathbf{r}$  и столбца  $\mathbf{c}$  пиксельного массива обозначается как  $\mathbf{B}[\mathbf{r}, \mathbf{c}]$ . Изображение размерами  $\mathbf{M} \times \mathbf{N}$  состоит из  $\mathbf{M}$  строк, пронумерованных от 0 до  $\mathbf{M} - 1$ , и  $\mathbf{N}$  столбцов, пронумерованных от 0 до  $\mathbf{N} - 1$ . Таким образом, обозначение  $\mathbf{B}[0, 0]$  соответствует значению левого верхнего пиксела изображения, а  $\mathbf{B}[\mathbf{M} - 1, \mathbf{N} - 1]$  — значению правого нижнего пиксела.

Во многих алгоритмах при обработке пикселов учитываются не только их значения, но и значения соседних пикселов. При рассмотрении соседних пикселов часто используются два определения окрестностей — *четырёхсвязная* (4-связная) и *восьмисвязная* (8-связная) окрестности. Четырёхсвязная окрестность  $\mathbf{N}_4[\mathbf{r}, \mathbf{c}]$  пиксела  $[\mathbf{r}, \mathbf{c}]$  состоит из пикселов  $[\mathbf{r} - 1, \mathbf{c}]$ ,  $[\mathbf{r} + 1, \mathbf{c}]$ ,  $[\mathbf{r}, \mathbf{c} - 1]$  и  $[\mathbf{r}, \mathbf{c} + 1]$ . Они часто называются, соответственно, северным (north), южным (south), западным (west) и восточным (east) соседями. Восьмисвязная окрестность  $\mathbf{N}_8[\mathbf{r}, \mathbf{c}]$  пиксела  $[\mathbf{r}, \mathbf{c}]$  содержит все пикселы четырёхсвязной окрест-

ности, а также диагональные соседние пиксели  $[\mathbf{r} - 1, \mathbf{c} - 1]$ ,  $[\mathbf{r} - 1, \mathbf{c} + 1]$ ,  $[\mathbf{r} + 1, \mathbf{c} - 1]$  и  $[\mathbf{r} + 1, \mathbf{c} + 1]$ . Их можно соответственно называть северо-западным (northwest), северо-восточным (northeast), юго-западным (southwest) и юго-восточным соседями (southeast). Расположение соседних пикселей в окрестностях двух типов демонстрируется на рис. 3.2.

	N	
W	*	E
	S	

(a) четырехсвязная окрестность  $N_4$

NW	N	NE
W	*	E
SW	S	SE

(b) восьмисвязная окрестность  $N_8$

**Рис. 3.2.** Две наиболее часто используемые окрестности пиксела.

Во многих алгоритмах четырехсвязные или восьмисвязные (или определенные иначе) окрестности часто называются просто *окрестностями* пикселей. В общем случае будем говорить, что пиксел  $[\mathbf{r}', \mathbf{c}']$  является *соседом* пиксела  $[\mathbf{r}, \mathbf{c}]$ , если  $[\mathbf{r}', \mathbf{c}']$  принадлежит заданной окрестности пиксела  $[\mathbf{r}, \mathbf{c}]$ .

## 3.2 ПРИМЕНЕНИЕ МАСОК К ИЗОБРАЖЕНИЯМ

В обработке изображений важным понятием является понятие *маски*, применяемой к изображению. Изначально это понятие использовалось при рассмотрении свертки — одной из операций обработки изображений. Но впоследствии маски стали использоваться и во многих других операциях обработки.

Маска — это множество координат пикселей, которым сопоставлены значения, называемые *весами*. На рис. 3.3 показаны три различные маски. Первые две маски (a) и (b) являются квадратными. У одной из них веса равны единице, а у другой веса имеют различные значения. Третья маска (c) имеет прямоугольные размеры и равные веса.

1	1	1
1	1	1
1	1	1

(a)

1	2	1
2	4	2
1	2	1

(b)

1
1
1
1
1

(c)

**Рис. 3.3.** Три возможные маски для применения к изображению.

У маски определено *начало координат*. Оно обычно совпадает с координатами одного из пикселей маски. В качестве начала координат симметричных масок (например, рис. 3.3 (a) и (b)) часто выбираются их центральные пиксели. У несимметричных масок в качестве начала может быть выбран любой пиксел. Выбор обычно зависит от способа использования маски. Например, у маски на рис. 3.3(c) в качестве начала координат может быть выбран верхний пиксел.

40	40	80	80	80
40	40	80	80	80
40	40	80	80	80
40	40	80	80	80
40	40	80	80	80

(a) Исходное полутоновое изображение

1	2	1
2	4	2
1	2	1

(b) Маска 3 × 3

640	800	1120	1280	1280
640	800	1120	1280	1280
640	800	1120	1280	1280
640	800	1120	1280	1280
640	800	1120	1280	1280

(c) Результат применения маски к изображению

40	50	70	80	80
40	50	70	80	80
40	50	70	80	80
40	50	70	80	80
40	50	70	80	80

(d) Нормировка результата путем деления на сумму весов маски (16)

(e) Исходное и результирующее изображения.  
Для удобства просмотра изображения увеличены до размеров 120 × 120**Рис. 3.4.** Применение маски весовых коэффициентов к полутоновому изображению.

В результате применения маски к входному изображению формируется выходное изображение. Размеры входного и выходного изображений одинаковы. Применение маски можно представлять себе следующим образом. Маска совмещается с входным изображением так, что центр маски совпадает с текущим обрабатываемым пикселем. Вес каждого пикселя маски умножается на значение соответствующего пикселя входного изображения. Эти произведения складываются, и полученное значение помещается в выходное изображение по координатам обрабатываемого пикселя.

На рис. 3.4 показано применение маски к полутоновому изображению. Исходное полутоновое изображение показано на рис. 3.4 (a), маска — на рис. 3.4 (b).

Когда центр маски попадает на пиксели граничных строк или столбцов, то некоторые пиксели маски выходят за границы изображения. Размеры входного и выходного изображений должны быть одинаковыми. Но к входному изображению вдоль его периметра можно добавить несколько виртуальных (воображаемых) строк и столбцов. В примере на рис. 3.4 были добавлены две виртуальные строки (одна выше и одна ниже изображения) и два виртуальных столбца (слева и справа от изображения). Значения пикселей в этих виртуальных столбцах можно произвольно выбрать нулевыми или равными какой-либо

другой константе, или, как было сделано в данном примере, значения пикселей могут быть просто скопированы из ближайшей строки (или столбца) изображения. Таким образом верхняя виртуальная строка содержит копии значений 40, 40, 80, 80, 80; виртуальный столбец слева — только значения 40; виртуальный столбец справа — значения 80; виртуальная строка внизу — значения 40, 40, 80, 80, 80. Выходное изображение (*c*), полученное применением маски (*b*), представляет собой сглаженную версию входного изображения (*a*). Однако все вычисленные значения намного больше пиксельных значений исходного изображения. Для нормировки можно разделить вычисленные значения каждого пикселя на сумму весов маски (в данном случае на 16). Полученное окончательное изображение приведено на рис. 3.4(*d*). Входное и выходное изображения показаны на рис. 3.4(*e*) в виде полутоновых изображений, увеличенных для удобства просмотра до размеров  $120 \times 120$  пикселей. При таком увеличении каждый пиксел выходного изображения выглядит как полоса шириной 24 пикселя. Это надо учитывать при визуальной оценке сглаживания, которое будет заметно на уровне полос, а не отдельных пикселей.

### 3.3 ПОДСЧЕТ ОБЪЕКТОВ НА ИЗОБРАЖЕНИИ

В гл. 1 рассматривалась прикладная задача, в которой требовалось вычислять количество отверстий на изображении объекта. Вместо количества отверстий в этой задаче можно было бы выполнять подсчет количества объектов переднего плана. Для обеих операций подходит один и тот же алгоритм, только для перехода к подсчету объектов переднего плана требуется поменять местами назначение двух множеств **E** и **I**. Для подсчета объектов шаблонами внешних углов будут маски  $2 \times 2$ , содержащие три значения 0 и одну 1. Шаблонами внутренних углов будут маски  $2 \times 2$ , содержащие три значения 1 и один 0. Эти два набора масок показаны на рис. 3.5. В алгоритме подсчета предполагалось, что каждый объект на изображении представлен в виде четырехсвязной единичной области без внутренних отверстий.

Рассмотрим применение одной из этих масок к бинарному изображению. Маска совмещается с изображением так, что ее верхний левый угол совпадает с текущим обрабатываемым пикселем бинарного изображения. В данном случае маска определяет окрестность пикселей, состоящую из текущего пикселя и трех его соседей (правого и двух нижних). Допустим, что значения всех 4 пикселей бинарного изображения, совмещенных с пикселями маски, равны значениям

0	0	0	0	1	0	0	1
0	1	1	0	0	0	0	0

(a) **E**: внешние углы

1	1	1	1	1	0	0	1
1	0	0	1	1	1	1	1

(b) **I**: внутренние углы

**Рис. 3.5.** Маски  $2 \times 2$  для подсчета количества объектов переднего плана на бинарном изображении. Значения 1 соответствуют пикселям переднего плана, значения 0 — фоновым пикселям.

соответствующих пикселей маски. Тогда будем считать, что в текущем пикселе бинарного изображения обнаружен угол, соответствующий маске. Предположим, что функция *external\_match(L, P)* последовательно перебирает все четыре маски внешних углов и возвращает *true*, если фрагмент с левым верхним пикселем  $[L, P]$  совпадает с одной из этих масок. Иначе функция возвращает *false*. Аналогично, функция *internal\_match(L, P)* возвращает *true*, если фрагмент с левым верхним пикселем  $[L, P]$  совпадает с одной из масок внутренних углов. В противном случае функция возвращает *false*. Функция подсчета объектов *count\_objects(B)* получает в качестве параметра бинарное изображение **B** и обрабатывает все пиксели изображения, за исключением последней строки и последнего столбца, в которые маску  $2 \times 2$  поместить нельзя. Функция возвращает количество объектов на бинарном изображении.

*Соглашения, принятые в описании алгоритмов.* Ниже приведена процедура подсчета объектов, записанная на псевдокоде. Далее аналогичный синтаксис будет применяться для записи всех процедур в данной книге. Отметим, что все подпрограммы называются *процедурами*, хотя некоторые из них представляют собой функции и содержат оператор возврата значения *return* (как в языке Си). Чтобы сделать процедуры по возможности краткими и простыми, в них часто будут использоваться вспомогательные процедуры (например, *external\_match* и *internal\_match*). Содержимое очевидных вспомогательных процедур,

### Вычисление количества объектов переднего плана на бинарном изображении **B**.

Объекты являются четырехсвязными и не содержат внутренних отверстий

**E** — количество внешних углов

**I** — количество внутренних углов

```

procedure count_objects(B);
{
E := 0;
I := 0;
for L := 0 to MaxRow - 1
  for P := 0 to MaxCol - 1
    {
      if external_match(L, P) then E := E + 1;
      if internal_match(L, P) then I := I + 1;
    };
return((E - I) / 4);
}

```

**Алгоритм 3.1.** Подсчет объектов переднего плана.



таких, как *external\_match* и *internal\_match*, часто опускается. Также обычно опускаются описания типов переменных, которые обычно зависят от языка программирования. Но необходимые типы и пояснение назначения основных переменных будут указываться в тексте комментариев. Чтобы уменьшить количество параметров вызова процедур, для хранения некоторых величин будут применяться глобальные константы.

В процедуре подсчета объектов константа **MaxRow** задает номер последней строки изображения, а **MaxCol** — номер последнего столбца. Номера первой строки и первого столбца считаются нулевыми (по аналогии с индексами начальных элементов массивов языка Си).

### Упражнение 3.1. Эффективность алгоритма подсчета объектов

Чему равно максимальное количество проверок, выполняемых в процедуре *count\_objects* для каждого пиксела изображения? Как наиболее эффективно реализовать процедуры *external\_match* и *internal\_match*?

### Упражнение 3.2. Обход угловых пикселей

В качестве пиксельного массива рассмотрим клетчатый лист бумаги. Закрасьте некоторую связную область клеток (пусть сначала она будет небольшой). Закрашенные клетки соответствуют пикселям переднего плана, а пустые клетки — фоновым пикселям.

Предположим, что пиксели представляют городские постройки. Требуется обойти вокруг закрашенной области в направлении часовой стрелки. Соответствуют ли места правых поворотов углам **E** или **I**? Что можно сказать о местах левых поворотов? Существует ли связь между количеством левых поворотов и количеством правых поворотов, которые необходимо сделать при полном обходе области по периметру? Если существует, то в чем состоит эта связь? В процессе полного обхода периметра приходится ли пересекать и касаться посещенных ранее пикселей? Возможны ли такие ситуации и почему?

Дополнительно рассмотрите случай двух диагонально соседних закрашенных клеток. Выполняются ли в этом случае ваши правила подсчета левых и правых поворотов? Применима ли формула подсчета объектов?

## 3.4 МАРКИРОВКА СВЯЗНЫХ КОМПОНЕНТ

Предположим, что дано бинарное изображение **V** и  $\mathbf{V}[\mathbf{r}, \mathbf{c}] = \mathbf{V}[\mathbf{r}', \mathbf{c}'] = v$ , где  $v = 0$  или  $v = 1$ . Пиксел  $[\mathbf{r}, \mathbf{c}]$  называется *связным* с пикселем  $[\mathbf{r}', \mathbf{c}']$  по значению  $v$ , если существует такая последовательность пикселей  $[\mathbf{r}, \mathbf{c}] = [\mathbf{r}_0, \mathbf{c}_0], [\mathbf{r}_1, \mathbf{c}_1], \dots, [\mathbf{r}_n, \mathbf{c}_n] = [\mathbf{r}', \mathbf{c}']$ , что  $\mathbf{V}[\mathbf{r}_i, \mathbf{c}_i] = v, i = 0, \dots, n$ , и  $[\mathbf{r}_i, \mathbf{c}_i]$  является соседом  $[\mathbf{r}_{i-1}, \mathbf{c}_{i-1}]$  для всех  $i = 1, \dots, n$ . Последовательность пикселей  $[\mathbf{r}_0, \mathbf{c}_0], \dots, [\mathbf{r}_n, \mathbf{c}_n]$  образует связный *путь* от пиксела  $[\mathbf{r}, \mathbf{c}]$  к пикселу  $[\mathbf{r}', \mathbf{c}']$ .

*Связная компонента* (connected component) со значением  $v$  — это множество  $S$  таких пикселей, которые имеют значения  $v$  и каждая пара этих пикселей является связной по значению  $v$ . На рис. 3.6 (а) показано бинарное изображение с пятью связными компонентами из единичных пикселей. Эти компоненты удовлетворяют условиям и четырехсвязности, и восьмисвязности.

**14 Определение.** Под **маркировкой связных компонент** бинарного изображения  $\mathbf{B}$  будем понимать формирование маркированного изображения  $\mathbf{LB}$ , в котором каждому пикселу присвоена метка связной компоненты, которой принадлежит данный пиксел.

Метка представляет собой некоторый идентификатор, используемый в качестве уникального имени объекта. Хотя иногда применяются символьные метки, для маркировки связных компонент часто более удобными оказываются метки в виде положительных целых чисел. На рис. 3.6 (b) показаны маркированные связные компоненты бинарного изображения рис. 3.6 (a).

Для маркировки связных компонент известно несколько различных алгоритмов. В некоторых алгоритмах предполагается, что все изображение может храниться в памяти целиком и для маркировки каждой компоненты применяется простая рекурсивная обработка, в процессе которой могут просматриваться пиксели всего изображения. Другие алгоритмы рассчитаны на обработку больших изображений, которые могут не помещаться в памяти целиком, так что изображение обрабатывается порциями по две строки. Ряд других алгоритмов предназначались для работы на параллельных компьютерах.

В данной главе будут рассмотрены два различных алгоритма маркировки: рекурсивный и построчный. В построчном алгоритме для хранения списка

1	1	0	1	1	1	0	1
1	1	0	1	0	1	0	1
1	1	1	1	0	0	0	1
0	0	0	0	0	0	0	1
1	1	1	1	0	1	0	1
0	0	0	1	0	1	0	1
1	1	0	1	0	0	0	1
1	1	0	1	0	1	1	1

(a) Бинарное изображение

1	1	0	1	1	1	0	2
1	1	0	1	0	1	0	2
1	1	1	1	0	0	0	2
0	0	0	0	0	0	0	2
3	3	3	3	0	4	0	2
0	0	0	3	0	4	0	2
5	5	0	3	0	0	0	2
5	5	0	3	0	2	2	2

(b) Маркированные связные компоненты



(c) Бинарное и маркированное изображения, увеличенные для удобства просмотра

**Рис. 3.6.** Бинарное изображение с пятью связными компонентами со значением 1.

компонент используется специальная структура данных, ориентированная на операции объединения множеств и поиска множества, которому принадлежит заданный элемент. Эту структуру будем называть структурой данных для объединения-поиска.

**Рекурсивный алгоритм маркировки.** Дано бинарное изображение  $\mathbf{B}$  размерами  $\text{MaxRow} + 1$  строк и  $\text{MaxCol} + 1$  столбцов. Требуется найти связные компоненты, состоящие из единичных пикселей, и сформировать выходное маркированное изображение  $\mathbf{LB}$ , в котором каждому пикселу присваивается метка его связной компоненты. Способ обработки основан на методе из книги по искусственному интеллекту. Сначала у пикселей бинарного изображения меняется знак [Tanimoto, 1990]. После смены знака все единичные пиксели получают значения  $-1$ . Это позволяет отличать необработанные пиксели ( $-1$ ) от пикселей компоненты с меткой  $1$ . Для операции смены знака будем пользоваться функцией с именем *negate*. Входным параметром этой функции является бинарное изображение  $\mathbf{B}$ , а выходным — инвертированное изображение  $\mathbf{LB}$ , которое впоследствии будет преобразовано в маркированное изображение.

Процесс поиска связных компонент сводится к следующим операциям. Сначала на  $\mathbf{LB}$  выполняется поиск пиксела со значением  $-1$ . Этому пикселу присваивается новая метка и вызывается процедура *search* для поиска всех соседей со значениями  $-1$ . Для каждого из найденных соседей производится рекурсивный вызов *search*. Получение соседних пикселей выполняется с помощью вспомогательной функции *neighbors(L, P)*. Ей передаются координаты пиксела в виде двух индексов  $\mathbf{L}$  и  $\mathbf{P}$ . Эта функция возвращает множество координат всех соседей указанного пиксела (для алгоритма не важно, четырех- или восьмисвязных). Возвращаемые координаты соответствуют только допустимым положениям на бинарном изображении. Координаты соседей перечисляются в растровом порядке (см. рис. 3.7). Реализация рекурсивного алгоритма маркировки связных компонент состоит из 6 процедур, в том числе *negate*, *print*, и *neighbors*, написание которых предоставляется читателю в качестве упражнения.

На рис. 3.8 демонстрируется применение рекурсивного алгоритма маркировки связных компонент к первой (верхней слева) компоненте бинарного изображения с рис. 3.6.

**Алгоритм построчной маркировки.** Этот алгоритм, описанный в статье [Rosenfeld, Pfaltz, 1966], часто называется классическим, так как он основан на классическом алгоритме поиска связных компонент на графах.

	1	
2	*	3
	4	

(a) четырехсвязная окрестность

1	2	3
4	*	5
6	7	8

(b) восьмисвязная окрестность

**Рис. 3.7.** Растровый порядок возврата координат соседних пикселей.

**Поиск связных компонент на бинарном изображении.****B** — входное бинарное изображение**LB** — выходное маркированное изображение

```

procedure recursive_connected_components(B, LB);
{
LB := negate(B);
label := 0;
find_components(LB, label);
print(LB);
}

procedure find_components(LB, label);
{
for L := 0 to MaxRow
  for P := 0 to MaxCol
    if LB[L,P] == -1 then
      {
        label := label + 1;
        search(LB, label, L, P);
      }
}

procedure search(LB, label, L, P);
{
LB[L,P] := label;
Nset := neighbors(L, P);
for each [L', P'] in Nset
  {
    if LB[L', P'] == -1
    then search(LB, label, L', P');
  }
}

```

**Алгоритм 3.2.** Рекурсивный алгоритм маркировки.

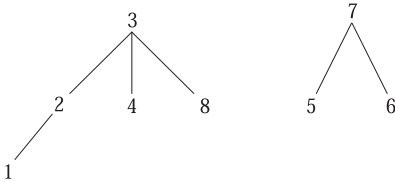
Алгоритм обрабатывает изображение за два прохода. На первом проходе определяются классы эквивалентности и присваиваются временные метки. На втором проходе каждая временная метка заменяется меткой соответствующего класса эквивалентности. Между первым и вторым проходами записанное множество отношений эквивалентности, сохраненное в виде бинарной таблицы, обрабатывается с целью определения классов эквивалентности. Алгоритм *объединения-поиска*, который динамически строит классы эквивалентности

Шаг 1.	<b>-1</b>	-1	0	-1	-1	-1
	-1	-1	0	-1	0	0
	-1	-1	-1	-1	0	0
Шаг 2.	1	<b>-1</b>	0	-1	-1	-1
	-1	-1	0	-1	0	0
	-1	-1	-1	-1	0	0
Шаг 3.	1	1	0	-1	-1	-1
	<b>-1</b>	-1	0	-1	0	0
	-1	-1	-1	-1	0	0
Шаг 4.	1	1	0	-1	-1	-1
	1	<b>-1</b>	0	-1	0	0
	-1	-1	-1	-1	0	0
Шаг 5.	1	1	0	-1	-1	-1
	1	1	0	-1	0	0
	<b>-1</b>	-1	-1	-1	0	0

**Рис. 3.8.** Первые пять шагов рекурсивного алгоритма маркировки в процессе обработки первой компоненты бинарного изображения на рис. 3.6. Показан фрагмент (частично) маркированного изображения **ЛВ**. Значение текущего пиксела, обрабатываемого в процедуре `search`, выделено полужирным шрифтом. Для обработки на следующем шаге выбирается первый еще необработанный соседний пиксел текущего пиксела, с учетом упорядочения соседей согласно рис. 3.7.

#### PARENT

1	2	3	4	5	6	7	8
2	3	0	3	7	7	0	3



**Рис. 3.9.** Структура данных для объединения-поиска, построенная для двух множеств меток. Первое множество содержит метки {1, 2, 3, 4, 8}, а второе — метки {5, 6, 7}. Для каждой целочисленной метки  $i$  значение **PARENT**[ $i$ ] является меткой родительского узла для узла  $i$ . Значение **PARENT**[ $i$ ] равно 0, если  $i$  является корневым узлом, у которого родительского узла нет.

после обнаружения связанных отношением эквивалентности объектов, стал широко использоваться в различных прикладных задачах. Структура данных, ориентированная на объединение и поиск, позволяет эффективно формировать классы эквивалентности, представленные в форме древовидных структур данных, и выполнять операции над ними. Добавление этой структуры данных в алгоритм маркировки оказалось полезным усовершенствованием классического алгоритма поиска связных компонент на графах.

*Структура данных для объединения-поиска (union-find data structure).* Эта структура данных предназначена для хранения набора непересекающихся множеств и для эффективной реализации операций *объединения* (слияние двух множеств в одно) и *поиска* (определение, какому множеству принадлежит заданный элемент). Каждое множество хранится в форме древовидной струк-

туры, в каждом узле которой хранится метка и ссылка на один родительский узел. В качестве базовой структуры данных используется массив **PARENT**, индексы которого образуют множество возможных меток, а значения элементов являются метками родительских узлов. Нулевой индекс родительского узла соответствует корневому узлу дерева. На рис. 3.9 показана древовидная структура для двух множеств меток  $\{1,2,3,4,8\}$  и  $\{5,6,7\}$ . Метка 3 является родительским узлом и меткой для первого множества, а метка 7 — родительским узлом и меткой второго множества. Значения в массиве **PARENT** говорят о том, что узлы 3 и 7 не имеют родительских узлов, метка 2 является родительской меткой для метки 1, метка 3 — для меток 2, 4 и 8 и т. п. Отметим, что нулевой элемент массива не используется, так как метка 0 соответствует фону, а значение 0 в элементах массива означает, что у соответствующих узлов нет родительского узла.

Процедура *find* принимает в качестве параметров метку **X** и массив с данными о родительских узлах **PARENT**. Эта процедура прослеживает родительские указатели вверх к корню дерева и находит метку корневого узла дерева, которому принадлежит **X**. Процедуре *union* передаются две метки **X** и **Y** и массив **PARENT**. Эта процедура модифицирует структуру данных (если необходимо) для слияния множества, содержащего метку **X**, с множеством, содержащим метку **Y**. Процедура начинает обработку с меток **X** и **Y** и обходит узлы дерева вверх до нахождения корневых узлов двух множеств. Если корневые узлы не совпадают, то узел одной из меток назначается родительским узлом для узла второй метки. Рассматриваемая здесь процедура *union* произвольно назначает узел **X** родительским узлом для **Y**. В процессе обработки можно отслеживать размеры множеств и присоединять меньшее множество к корню большего множества. Таким образом предотвращается чрезмерное увеличение глубины дерева.

### Поиск родительской метки для заданной метки множества.

**X** — метка множества

**PARENT** — массив, представляющий структуру данных для объединения-поиска

```
procedure find(X, PARENT);  
{  
  j := X;  
  while PARENT[j] <> 0  
    j := PARENT[j];  
  return(j);  
}
```

**Алгоритм 3.3.** Процедура Find для определения принадлежности метки множеству.

### Построение объединения двух множеств.

**X** — метка, принадлежащая первому множеству.

**Y** — метка, принадлежащая второму множеству.

**PARENT** — массив, представляющий структуру данных для объединения-поиска.

```

procedure union(X, Y, PARENT);
{
j := X;
k := Y;
while PARENT[j] <> 0
  j := PARENT[j];
while PARENT[k] <> 0
  k := PARENT[k];
if j <> k then PARENT[k] := j;
}

```

**Алгоритм 3.4.** Процедура Union для объединения двух множеств.

*Классический алгоритм маркировки связанных компонент на основе структуры данных для объединения-поиска.* Эта структура данных обеспечивает более эффективную реализацию классического алгоритма маркировки связанных компонент. На первом проходе алгоритм пытается распространить метки с целью присвоить их правым и нижним соседям уже помеченных пикселей. Когда возникает ситуация, в которой две различные метки могут распространиться на один и тот же пиксел, то для распространения выбирается метка с меньшим значением. Каждый такой класс эквивалентности заносится в структуру данных объединения-поиска. После первого прохода каждый класс эквивалентности будет полностью определен и снабжен уникальной меткой, соответствующей корневому узлу дерева в структуре объединения-поиска. На втором проходе выполняется преобразование этой структуры данных путем присвоения каждому пикселу выходного изображения метки его класса эквивалентности.

В реализации процедуры маркировки используются две вспомогательные функции: *prior\_neighbors* и *labels*. Функция *prior\_neighbors* возвращает множество соседних единичных пикселей выше и слева от заданного пиксела. Эта функция может быть реализована как в четырехсвязном (тогда она возвращает северного и западного соседей), так и в восьмисвязном варианте (в этом случае возвращаются северо-западный, северный, северо-восточный и западный соседи). Функция *labels* возвращает множество меток, которые уже были присвоены заданному множеству пикселей.

1	1	0	2	2	2	0	3
1	1	0	2	0	2	0	3
1	1	1	1	0	0	0	3
0	0	0	0	0	0	0	3
4	4	4	4	0	5	0	3
0	0	0	4	0	5	0	3
6	6	0	4	0	0	0	3
6	6	0	4	0	7	7	3

(a) Маркированное изображение после прохода 1

1	1	0	1	1	1	0	3
1	1	0	1	0	1	0	3
1	1	1	1	0	0	0	3
0	0	0	0	0	0	0	3
4	4	4	4	0	5	0	3
0	0	0	4	0	5	0	3
6	6	0	4	0	0	0	3
6	6	0	4	0	3	3	3

(c) Маркированное изображение после прохода 2

## PARENT

1	2	3	4	5	6	7
0	1	0	0	0	0	3

(b) Структура данных для объединения-поиска, представляющая обнаруженные классы эквивалентности

**Рис. 3.10.** Применение классического алгоритма на основе структуры данных объединения-поиска к бинарному изображению на рис. 3.6.

На рис. 3.10 демонстрируется применение классического алгоритма на основе структуры данных для объединения-поиска к бинарному изображению с рис. 3.6. На рис. 3.10 (a) показаны метки каждого пиксела после первого прохода. На рис. 3.10 (b) приведена структура данных для объединения-поиска, согласно которой на первом проходе были определены классы эквивалентности  $\{\{1, 2\}, \{3, 7\}, 4, 5, 6\}$ . На рис. 3.10 (c) показаны окончательные результаты маркировки изображения после второго прохода. Связные компоненты представляют области изображения, для которых впоследствии могут быть вычислены характерные признаки формы и интенсивности. Некоторые из этих свойств будут рассматриваться в разделе 3.5.

*Применение группового кодирования для маркировки связных компонент.* Как отмечалось в гл. 2, при *групповом кодировании* (run-length encoding) бинарного изображения формируется список непрерывных горизонтальных цепочек единичных значений. Для каждой цепочки запоминается местоположение первого пиксела и длина цепочки или положение последнего пиксела цепочки. На рис. 3.11 показан пример структуры данных, применяемой при групповом кодировании. Каждая цепочка на изображении записывается в виде координат первого и последнего пиксела цепочки. Координаты (**ROW**, **START\_COL**) определяют местоположение первого пиксела цепочки, а (**ROW**, **END\_COL**) — последнего. Поле **LABEL** предназначено для хранения метки связной компоненты, которой принадлежит данная цепочка. Это поле инициализируется нулевым значением. На первом проходе в поле **LABEL** помещаются временные значения. После второго прохода это поле будет содержать окончательную метку цепочки. Затем сформированную структуру данных можно использовать для записи значений меток в соответствующие пиксела выходного изображения.



### Инициализация структур данных для классического алгоритма маркировки связных компонент.

```

procedure initialize();
“Инициализация глобальной переменной label и массива
PARENT.”
{
“Инициализация label.”
label := 0;
“Инициализация структуры данных для объединения-поиска.”
for i := 1 to MaxLab
    PARENT[i] := 0;
}

```

**Алгоритм 3.5.** Инициализация данных в классическом алгоритме маркировки связных компонент.

	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
<b>0</b>	1	1	0	1	1
<b>1</b>	1	1	0	0	1
<b>2</b>	1	1	1	0	1
<b>3</b>	0	0	0	0	0
<b>4</b>	0	1	1	1	1

(a)

	<b>ROW_START</b>	<b>ROW_END</b>
<b>0</b>	1	2
<b>1</b>	3	4
<b>2</b>	5	6
<b>3</b>	0	0
<b>4</b>	7	7

(b)

	<b>ROW</b>	<b>START_COL</b>	<b>END_COL</b>	<b>LABEL</b>
<b>1</b>	0	0	1	0
<b>2</b>	0	3	4	0
<b>3</b>	1	0	1	0
<b>4</b>	1	4	4	0
<b>5</b>	2	0	2	0
<b>6</b>	2	4	4	0
<b>7</b>	4	1	4	0

(c)

**Рис. 3.11.** Бинарное изображение в исходном виде (a) и после группового кодирования (b) и (c). Для каждой цепочки единичных значений записывается номер строки (**ROW**) и столбца начального и конечного пикселей цепочки (**START\_COL** и **END\_COL**). Для каждой строки изображения **ROW\_START** указывает на первую, а **ROW\_END** — на последнюю цепочку в данной строке. Поле **LABEL** предназначено для хранения метки компоненты, которой принадлежит цепочка. Поле инициализируется нулевым значением.

### 3.5 МОРФОЛОГИЯ БИНАРНЫХ ИЗОБРАЖЕНИЙ

Термин *морфология* относится к описанию свойств формы и структуры каких-либо объектов. В компьютерном зрении этот термин часто относится к описанию свойств формы областей на изображении. Операции *математической морфологии* изначально были определены в виде операций над множествами. Они также оказались полезны в задачах обработки множеств двумерных точек. В данном разделе будут определены операции бинарной морфологии и продемонстрировано применение этих операций для обработки областей, обнаруженных в результате маркировки связных компонент.

#### 3.5.1 Структурирующие элементы

Входными данными для операций бинарной морфологии являются бинарное изображение  $\mathbf{B}$  и *структурирующий элемент* (structuring element)  $\mathbf{S}$ , представляющий собой другое, обычно меньшее, бинарное изображение. Структурирующий элемент можно считать описанием области некоторой формы. Эта область может иметь любые размеры и форму, которую можно представить в виде бинарного изображения. Существует набор наиболее часто используемых структурирующих элементов, например, прямоугольник заданных размеров [BOX( $l, w$ )] или диск заданного диаметра [DISK( $d$ )]. В некоторых пакетах для обработки изображений пользователю предлагаются библиотеки подобных элементарных структурирующих элементов. Примеры нескольких распространенных и нескольких нестандартных структурирующих элементов показаны на рис. 3.12.

#### Упражнение 3.3. Сравнение алгоритмов маркировки

Предположим, что на бинарном изображении есть одна область переднего плана в виде прямоугольника размерами  $1000 \times 1000$ . Сколько раз рекурсивный алгоритм будет просматривать (для чтения или записи) каждый пиксел? Сколько раз будет обрабатываться каждый пиксел классическим алгоритмом маркировки?

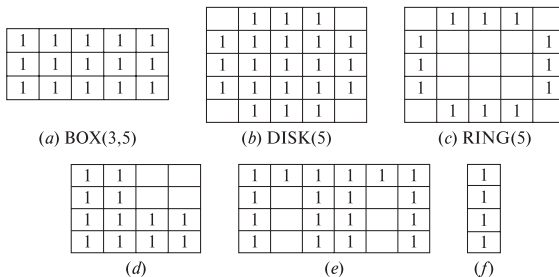


Рис. 3.12. Примеры структурирующих элементов (пустые клетки соответствуют нулевым значениям).

### Поиск связных компонент на бинарном изображении.

**B** — исходное бинарное изображение.

**LB** — выходное маркированное изображение.

```

procedure classical_with_union-find(B, LB);
{
  “Инициализация структур данных.”
  initialize();
  “На проходе 1 каждой строке L изображения присваиваются
  начальные значения меток.”
  for L := 0 to MaxRow
  {
    “Инициализация всех меток в строке L нулевыми значениями”
    for P := 0 to MaxCol
      LB[L,P] := 0;
    “Обработка строки L.”
    for P := 0 to MaxCol
      if B[L,P] == 1 then
        {
          A := prior_neighbors(L,P);
          if isempty(A)
            then {M := label; label := label + 1};
          else M := min(labels(A));
          LB[L,P] := M;
          for X in labels(A) and X <> M
            union(M, X, PARENT);
        }
      }
    }
  “На проходе 2 метки, обнаруженные на проходе 1, заменяются
  метками соответствующих классов эквивалентности.”
  for L := 0 to MaxRow
    for P := 0 to MaxCol
      if B[L, P] == 1
        then LB[L,P] := find(LB[L,P], PARENT);
  };

```

**Алгоритм 3.6.** Классический алгоритм маркировки связных компонент с использованием структуры данных для объединения-поиска.

### Упражнение 3.4. Замена меток

Так как эквивалентные метки объединяются в один класс эквивалентности, то некоторые начальные метки, обнаруженные на проходе 1, на проходе 2

теряются. В результате в окончательном множестве метки могут быть пронумерованы с многочисленными пропусками номеров. Напишите процедуру для замены меток, которая будет преобразовывать метки компонент так, чтобы компоненты были пронумерованы последовательно возрастающими числами от 1 и до количества компонент на изображении.

### Упражнение 3.5. Групповое кодирование

Спроектируйте и реализуйте построчный алгоритм маркировки, использующий бинарное изображение не в исходном виде, а после сжатия методом группового кодирования. Для хранения меток цепочек должно использоваться поле **LABEL** в структуре данных для объединения-поиска.

Структурирующие элементы можно представлять себе как зонды, применяемые для взятия данных с бинарного изображения. Один из пикселей структурирующего элемента выбирается в качестве *начала координат*. В симметричных структурирующих элементах в этом качестве обычно выбирается центральный пиксел, хотя в принципе может быть выбран любой. Используя начало координат как точку привязки, структурирующий элемент может быть перенесен в любой пиксел изображения и использован для добавления к области бинарного изображения формы элемента или для проверки, попадает или нет форма элемента в данную область изображения. Например, для контроля размера отверстия можно было бы проверить, что маленький диск полностью помещается в заданную область, а диск большего размера — не полностью.

### 3.5.2 Основные операции

Основными операциями бинарной морфологии являются *наращивание* (dilation), *эрозия* (erosion), *замыкание* (closing) и *размыкание* (opening). В этих названиях отражено, что операция наращивания увеличивает область изображения, а эрозия делает ее меньше. Операция замыкания позволяет замкнуть внутренние отверстия области и устранить *заливы* (bay) вдоль границы области. Операция размыкания помогает избавиться от маленьких фрагментов, выступающих наружу области вблизи ее границы. Далее будут представлены математические определения морфологических операций.

**15 Определение.** Операция **переноса** (translation)  $X_t$  множества пикселей  $X$  на вектор  $t$  задается в виде

$$X_t = \{x + t \mid x \in X\} \quad (3.1)$$

Следовательно, перенос множества единичных пикселей на бинарном изображении сдвигает все пиксели множества на заданное расстояние. Вектор переноса  $t$  может задаваться в виде упорядоченной пары  $(\Delta r, \Delta c)$ , где  $\Delta r$  — компонент вектора переноса в направлении строк, а  $\Delta c$  — компонент вектора переноса в направлении столбцов изображения.

**16 Определение.** **Наращивание** бинарного изображения  $\mathbf{B}$  структурирующим элементом  $\mathbf{S}$  обозначается  $\mathbf{B} \oplus \mathbf{S}$  и задается следующим образом

$$B \oplus S = \bigcup_{b \in B} S_b \quad (3.2)$$

В данном выражении оператор объединения можно считать оператором, применяемым в окрестности пикселей. Структурирующий элемент  $\mathbf{S}$  применяется ко всем пикселям бинарного изображения. Каждый раз, когда начало координат структурирующего элемента совмещается с единичным бинарным пикселем, ко всему структурирующему элементу применяется перенос и последующее логическое сложение (логическое ИЛИ) с соответствующими пикселями бинарного изображения. Результаты логического сложения записываются в выходное бинарное изображение, которое изначально инициализируется нулевыми значениями. На рис. 3.13 (a) показано бинарное изображение, а на рис. 3.13 (c) демонстрируется результат наращивания прямоугольным структурирующим элементом  $3 \times 3$ , который приведен на рис. 3.13 (b).

Поясним математическое определение на примере обработки первого единичного пикселя бинарного изображения  $\mathbf{B}$ . Координаты этого пикселя равны  $[1,0]$ , т. е. он располагается на пересечении строки 1 и столбца 0 изображения  $\mathbf{B}$ . После переноса  $\mathbf{S}_{(1,0)}$  начало координат структурирующего элемента  $S$  (его центр) совпадает с пикселем  $[1,0]$ . В результате логического сложения в выходное изображение (сначала заполненное нулями) будут помещены единичные пиксели с координатами  $[0,0]$ ,  $[0,1]$ ,  $[1,0]$ ,  $[1,1]$ ,  $[2,0]$  и  $[2,1]$  и с виртуальными координатами  $[0,-1]$ ,  $[1,-1]$  и  $[2,-1]$ , которые могут быть проигнорированы. Для следующего пикселя  $\mathbf{B}$  с координатами  $[1,1]$  выполняется перенос  $\mathbf{S}_{(1,1)}$  и последующее логическое сложение дает в выходном изображении единичные пиксели с координатами  $[0,0]$ ,  $[0,1]$ ,  $[0,2]$ ,  $[1,0]$ ,  $[1,1]$ ,  $[1,2]$ ,  $[2,0]$ ,  $[2,1]$ ,  $[2,2]$ . Аналогичные операции продолжаются до тех пор, пока копии структурирующего элемента не будут логически сложены в позиции каждого единичного пикселя входного изображения. Результирующее выходное изображение показано на рис. 3.13 (c).

**17 Определение.** **Эрозия** бинарного изображения  $\mathbf{B}$  структурирующим элементом  $\mathbf{S}$  обозначается  $\mathbf{B} \ominus \mathbf{S}$  и задается выражением

$$B \ominus S = \{ b \mid b + s \in B \forall s \in S \} \quad (3.3)$$

При выполнении операции эрозии структурирующий элемент тоже «проходит» по всем пикселям входного изображения. Если в некоторой позиции каждый единичный пиксел структурирующего элемента совпадает с единичным пикселем бинарного изображения, то выполняется логическое сложение центрального пикселя структурирующего элемента с соответствующим пикселем выходного изображения. На рис. 3.13 (d) показан результат эрозии бинарного изображения рис. 3.13 (a) прямоугольным структурирующим элементом  $3 \times 3$ .

1	1	1	1	1	1	1	
			1	1	1	1	
			1	1	1	1	
		1	1	1	1	1	
			1	1	1	1	
		1	1				

(a) Бинарное изображение  $\mathbf{B}$ 

1	1	1
1	<b>1</b>	1
1	1	1

(b) Структурирующий элемент  $\mathbf{S}$ 

1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
	1	1	1	1	1	1	1
	1	1	1	1	1	1	1
	1	1	1	1	1	1	1
	1	1	1	1	1	1	1
	1	1	1	1	1	1	1
	1	1	1	1			

(c) Нарращивание  $\mathbf{B} \oplus \mathbf{S}$ 

			1	1			
			1	1			
			1	1			

(d) Эрозия  $\mathbf{B} \ominus \mathbf{S}$ 

	1	1	1	1	1	1	
		1	1	1	1	1	
			1	1	1	1	
			1	1	1	1	
			1	1	1	1	
			1	1			

(e) Замыкание  $\mathbf{B} \bullet \mathbf{S}$ 

		1	1	1	1		
		1	1	1	1		
		1	1	1	1		
		1	1	1	1		
		1	1	1	1		

(f) Размыкание  $\mathbf{B} \circ \mathbf{S}$ 

**Рис. 3.13.** Основные операции бинарной морфологии. Пиксели переднего плана имеют единичные значения. Фоновые нулевые пиксели показаны в виде пустых клеток.

Нарращивание и эрозия представляют собой две самые простые операции математической морфологии. Композиция этих примитивных операций образует еще две часто используемые операции: замыкание и размыкание.

**18 Определение.** Замыкание бинарного изображения  $\mathbf{B}$  структурирующим элементом  $\mathbf{S}$  обозначается  $\mathbf{B} \bullet \mathbf{S}$  и задается выражением

$$\mathbf{B} \bullet \mathbf{S} = (\mathbf{B} \oplus \mathbf{S}) \ominus \mathbf{S} \quad (3.4)$$

**19 Определение.** Размыкание бинарного изображения  $\mathbf{B}$  структурирующим элементом  $\mathbf{S}$  обозначается  $\mathbf{B} \circ \mathbf{S}$  и задается выражением

$$\mathbf{B} \circ \mathbf{S} = (\mathbf{B} \ominus \mathbf{S}) \oplus \mathbf{S} \quad (3.5)$$

На рис. 3.13 (e) показан результат замыкания, а на рис. 3.13 (f) — результат размыкания бинарного изображения рис. 3.13 (a) прямоугольным структурирующим элементом  $3 \times 3$ .

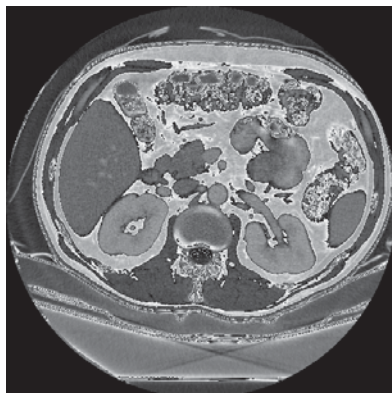
### Упражнение 3.6. Применение элементарных операций бинарной морфологии

С помощью камеры получено изображение **I**, на котором представлены монеты достоинством 1, 10 и 25 центов. Монеты расположены на белом фоне и не касаются друг друга. Бинарное изображение **B** было успешно сформировано с помощью пороговой бинаризации. Единичные пиксели соответствуют образам монет, а нулевые — белому фону. Диаметры монет  $D_P$ ,  $D_D$  и  $D_Q$  известны. С помощью операций математической морфологии (наращивание, эрозия, размыкание и замыкание) и логических операций (И, ИЛИ, НЕ и РАЗНОСТЬ — разность множеств) покажите, как получить три бинарных выходных изображения **P**, **D** и **Q** трех различных монет. На изображении **P** должен присутствовать только образ монеты достоинством 1 цент, на изображении **D** — только монеты достоинством 10 центов, а на изображении **Q** — монеты достоинством 25 центов.

### 3.5.3 Некоторые приложения бинарной морфологии

Операции замыкания и размыкания полезны в задачах обработки изображений, в которых после пороговой бинаризации (или в результате какой-либо другой операции предварительной обработки) формируется бинарное изображение с небольшими отверстиями в образах связанных компонент, или пары компонент, которые должны быть отдельными, оказываются соединенными небольшой полоской пикселей переднего плана. На рис. 3.14 (a) представлено 16-битное полутоновое медицинское изображение размерами  $512 \times 512$  пикселей. На рис. 3.14 (b) приведен результат пороговой бинаризации исходного изображения для выделения пикселей с яркостями больше 1070 градаций. На рис. 3.14 (c) представлено бинарное изображение после выполнения операции размыкания для разделения образов отдельных органов и операции замыкания для устранения малых отверстий. Для размыкания применяется структурирующий элемент  $\text{DISK}(13)$ , а для замыкания — элемент  $\text{DISK}(2)$ .

Методы бинарной морфологии могут применяться для выполнения некоторых весьма специфических операций в инспекционных промышленных системах машинного зрения. В статье [Sternberg, 1985] рассмотрен пример контроля качества часовых шестеренок на предмет отсутствия или дефекта зубьев. На рис. 3.15 (a) показано бинарное изображение контролируемых шестеренок. Образ шестеренки содержит четыре внутренних отверстия. На внешней границе шестеренки видно большое количество зубцов, которые на изображении заметны по отдельности. Для обработки подобных изображений шестеренок применялись несколько структурирующих элементов специального назначе-



(a) Полутоновое медицинское изображение G



(b) Бинарное изображение B



(c) Результат применения морфологических операций

**Рис. 3.14.** Применение бинарной морфологии для обработки медицинских изображений. 16-битное полутоновое изображение размерами  $512 \times 512$  пикселей (a) после применения пороговой бинаризации (с порогом 1070) было преобразовано в бинарное изображение, показанное на рис. (b). Изображение (c) получено после выполнения размыкания структурирующим элементом DISK(13) и последующего замыкания элементом DISK(2).



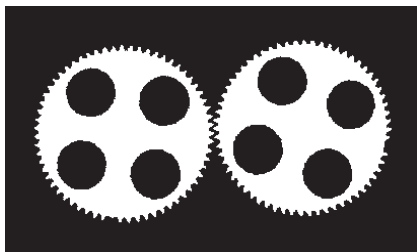
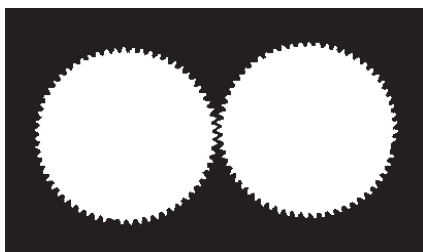
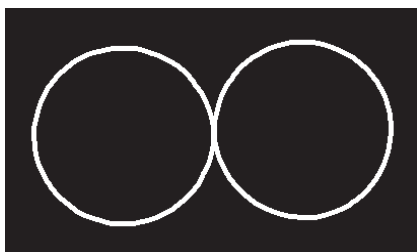
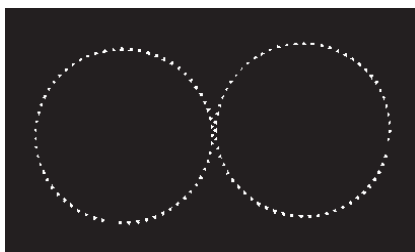
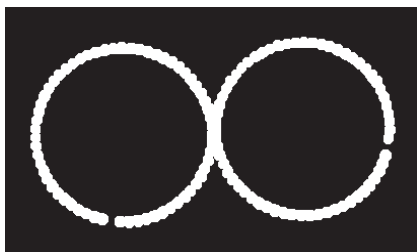
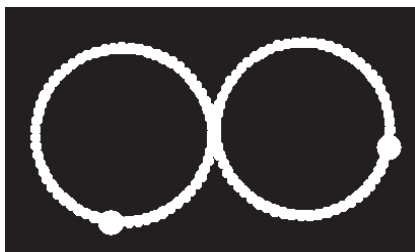
(a) Исходное изображение  $B$ (b)  $B1 = B \ominus \text{hole\_ring}$ (c)  $B2 = B1 \oplus \text{hole\_mask}$ (d)  $B3 = B \text{ ИЛИ } B2$ (e)  $B7$  (см. в тексте)(f)  $B8 = B \text{ И } B7$ (g)  $B9 = B8 \oplus \text{tip\_spacing}$ (h)  $\text{RESULT} = ((B7 - B9) \oplus \text{defect\_cue}) \text{ ИЛИ } B9$ 

Рис. 3.15. Процедура инспекционного контроля часовых шестеренок. (С любезного разрешения Stanley R. Sternberg и Academic Press.)

ния. Размеры и форма этих элементов были определены с учетом физических свойств шестеренок. В алгоритме инспекции часовых шестеренок применялись следующие структурирующие элементы:

- **hole\_ring**: кольцо пикселей, диаметр которого немного превышает диаметр четырех внутренних отверстий шестеренок. Этот элемент помещался в области внутренних отверстий и применялся для отметки нескольких пикселей в качестве предполагаемых центров отверстий.
- **hole\_mask**: восьмиугольник, площадь которого немного превышает площадь внутренних отверстий шестеренок.
- **gear\_body**: дисковый структурирующий элемент с радиусом, равным внешнему радиусу шестеренки, уменьшенному на высоту зубца.
- **sampling\_ring\_spacer**: дисковый структурирующий элемент, применяемый для небольшого смещения выборочного кольца по направлению от центра шестеренки.
- **sampling\_ring\_width**: дисковый структурирующий элемент, используемый для наращивания выборочного кольца в направлении вершин зубцов.
- **tip\_spacing**: дисковый структурирующий элемент, диаметр которого соответствует промежутку между зубцами шестеренки.
- **defect\_cue**: дисковый структурирующий элемент, предназначенный для наращивания в областях дефектов, чтобы они были явно заметны для пользователя.

Процедура инспекции часовых шестеренок поясняется на рис. 3.15. Исходное бинарное изображение показано на рис. 3.15 (a). На рис. 3.15 (b) представлен результат эрозии исходного изображения структурирующим элементом **hole\_ring**. В полученном изображении единичные пиксели образуют небольшие кластеры в окрестности центров внутренних отверстий шестеренок. Эти пиксели соответствуют небольшому количеству положений, в которых элемент **hole\_ring** полностью совпадает с содержимым образа объекта. На рис. 3.15 (c) представлен результат наращивания предыдущего изображения элементом **hole\_mask**. В результате формируется изображение с восьмиугольниками, соответствующими внутренним отверстиям шестеренок. Изображение на рис. 3.15 (d) получено логическим сложением образов восьмиугольников с исходным бинарным изображением. В результате внутренние отверстия на образах шестеренок оказываются заполненными единичными пикселями.

На следующем шаге формируется выборочное кольцо для контроля зубцов шестеренки. Для его построения берется изображение с рис. 3.15 (d) и подвергается размыканию элементом **gear\_body** с целью отделения области зубцов. Затем выполняется наращивание элементом **sampling\_ring\_spacer** для смещения кольца к основанию зубцов. Получается промежуточное изображение. Оно наращивается элементом **sampling\_ring\_width** для переноса изображения кольца к вершинам зубцов. Полученное изображение вычитается

из промежуточного и дает изображение кольца, которое полностью закрывает область зубцов шестеренки. Это выборочное кольцо показано на рис. 3.15 (e).

После построения выборочного кольца, соответствующего области зубцов, оно логически умножается на исходное изображение. В результате получается изображение, содержащее только зубцы (рис. 3.15 (f)). Дефектные промежутки между зубцами (т. е. отсутствующие зубцы) заметны, но никак не отмечены. Нарращивание изображения зубцов элементом **tip\_spacing** дает изображение кольца с разрывами в области отсутствующих зубцов (рис. 3.15 (g)). Вычитание этого кольца из выборочного кольца оставляет только области дефектов. Они наращиваются элементом **defect\_cue** и демонстрируются пользователю на экране в виде крупных пятен в области дефектов.

### Упражнение 3.7. Выбор структурирующих элементов

В работе [Sternberg, 1985] был применен кольцевой структурирующий элемент для обнаружения центров внутренних отверстий шестеренок. Как можно обнаружить центры отверстий, если разрешается пользоваться только дисковыми и прямоугольными структурирующими элементами?

### Упражнение 3.8. Приложение морфологической обработки

Предположим, что после пороговой бинаризации спутниковой фотографии земной поверхности пиксели, соответствующие водным пространствам, получают единичные значения. Мосты на реках представлены в виде тонких линий из нулевых пикселей, которые пересекают единичные области рек. (а) Предложите способ устранения пикселей, соответствующих мостам, чтобы они не разрывали области водной поверхности.

(б) Предложите способ обнаружения узких образов мостов в качестве отдельных объектов.

Бинарная морфология может также применяться для извлечения примитивных характерных признаков объекта для последующего использования этих признаков в процессе распознавания объекта. Например, в качестве примитивных признаков при распознавании формы плоского двумерного объекта могут быть выбраны его углы. Если образ объекта с резкими углами подвергнуть размыканию дисковым структурирующим элементом, то углы будут



(a) Исходное изображение

(b) Размыкание

(c) Углы

Рис. 3.16. Выделение областей примитивной формы с помощью бинарной морфологии.

«отрезаны», как показано на рис. 3.16. После вычитания полученного изображения из исходного бинарного изображения будет сформировано изображение, содержащее только углы объекта. Оно может использоваться в алгоритме распознавания формы объекта. В системе распознавания формы объектов может быть предусмотрено морфологическое выделение признаков для быстрого обнаружения примитивов, полезных при распознавании объектов.

### 3.5.4 Условное наращивание

Одним из типичных применений бинарной морфологии является выделение на бинарном изображении компонент, у которых форма и размеры удовлетворяют заданным ограничениям. Во многих подобных задачах возможно построение структурирующего элемента, который после применения к бинарному изображению удаляет не удовлетворяющие ограничениям компоненты и оставляет несколько единичных пикселей, соответствующих удовлетворяющим ограничениям компонентам. Но для последующей обработки могут потребоваться компоненты целиком, а не только их фрагменты, оставшиеся после эрозии. Для решения этой проблемы была введена операция *условного наращивания* (conditional dilation).

**20 Определение.** Обозначим исходное бинарное изображение  $\mathbf{B}$ , обработанное бинарное изображение  $\mathbf{C}$  и структурирующий элемент  $\mathbf{S}$ . Пусть  $\mathbf{C}_0 = \mathbf{C}$  и  $\mathbf{C}_n = (\mathbf{C}_{n-1} \oplus \mathbf{S}) \cap \mathbf{B}$ . **Условным наращиванием** изображения  $\mathbf{C}$  элементом  $\mathbf{S}$  по отношению к  $\mathbf{B}$  называется операция

$$\mathbf{C} \oplus |_{\mathbf{B}} \mathbf{S} = \mathbf{C}_m \quad (3.6)$$

где индекс  $m$  равен наименьшему значению, при котором  $\mathbf{C}_m = \mathbf{C}_{m-1}$ .

Приведенное определение сформулировано для конечных дискретных множеств точек на цифровых изображениях. Оно говорит, что множество  $\mathbf{C} = \mathbf{C}_0$  циклически наращивается структурирующим элементом  $\mathbf{S}$ , и на каждом шаге результат уменьшается до подмножества пикселей, которые имеют единичные значения на исходном изображении  $\mathbf{B}$ . Операция условного наращивания поясняется на рис. 3.17. На этом рисунке бинарное изображение  $\mathbf{B}$  было подвергнуто эрозии элементом  $\mathbf{V}$  для выделения компонент, содержащих вертикальные фрагменты высотой 3 пикселя. На полученном изображении  $\mathbf{C}$  есть две таких компоненты. Чтобы выделить эти компоненты целиком, изображение  $\mathbf{C}$  условно наращивается элементом  $\mathbf{D}$  относительно исходного изображения  $\mathbf{B}$ .

## 3.6 СВОЙСТВА ОБЛАСТЕЙ

После того как на бинарном изображении было обнаружено множество областей, может потребоваться определить их свойства. Эти свойства часто используются в качестве входных данных в процедурах более высокого уровня, например, для распознавания объектов или для решения инспекционных задач. В большинстве пакетов обработки изображений реализованы операции для

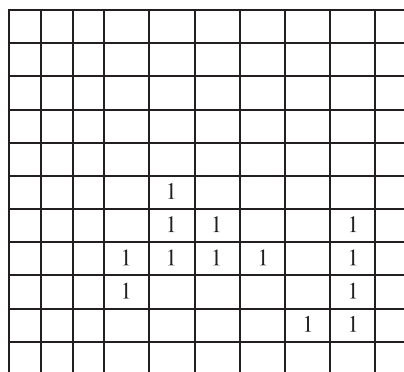
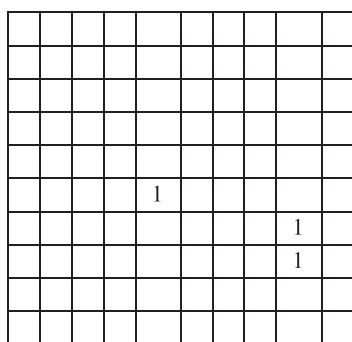
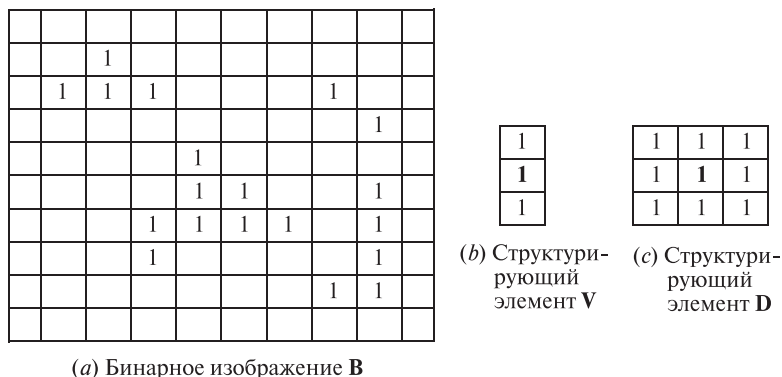


Рис. 3.17. Операция условного наращивания.

получения набора свойств каждой области бинарного изображения. Наиболее распространенными свойствами являются геометрические свойства (например, площадь области, центр тяжести, экстремальные точки), свойства формы (например, характеристики округлости и продолговатости), свойства интенсивности (средняя интенсивность области и различные текстурные статистики). В данном разделе будут даны определения нескольких часто используемых геометрических свойств и свойств, связанных с формой областей. Будут рассмотрены способы использования этих свойств в задачах принятия решений. Свойства, связанные с интенсивностью на полутоновых изображениях, будут обсуждаться в гл. 7.

В последующем изложении будем обозначать множество пикселей области буквой  $R$ . Пиксели считаются квадратными. Ниже приведено определение двух простейших геометрических свойств области — площади  $A$  и координат центра тяжести  $(\bar{r}, \bar{c})$ .

**Площадь:**

$$A = \sum_{(r,c) \in R} 1 \quad (3.7)$$

Из (3.7) следует, что площадь равна количеству пикселей, принадлежащих области  $R$ .

**Центр тяжести (centroid):**

$$\bar{r} = \frac{1}{A} \sum_{(r,c) \in R} r \quad (3.8)$$

$$\bar{c} = \frac{1}{A} \sum_{(r,c) \in R} c \quad (3.9)$$

Координаты центра тяжести  $(\bar{r}, \bar{c})$  равны *средним* значениям координат пикселей области  $R$ . Отметим, что, хотя каждая пара  $[\mathbf{r}, \mathbf{c}] \in R$  состоит из двух целых чисел, координаты  $(\bar{r}, \bar{c})$  в общем случае не являются целочисленными. Во многих задачах достаточно вычислять координаты центра тяжести с точностью до одного знака после запятой.

---

### Упражнение 3.9. Использование свойства «площадь»

В рассмотренном ранее примере инспекционного контроля шестеренок для обработки изображений применялись только морфологические и логические операции. Эти операции очень быстро выполняются на специально разработанных вычислительных устройствах, а на компьютерах общего назначения выполняются значительно медленнее. Как можно было бы обнаружить дефектные большие промежутки между зубцами шестеренки с применением наименьшего количества морфологических операций?

Длина *периметра*  $P$  представляет собой еще одно глобальное свойство области. В качестве простого определения периметра области без внутренних отверстий можно рассматривать множество граничных пикселей области. Пиксел области  $R$  называется граничным, если один из его соседей не принадлежит  $R$ . Если принадлежность пикселей области или фону определяется по восьмисвязности, то полученное множество пикселей периметра будет четырехсвязным. Наоборот, если принадлежность пикселя области или фону определяется по четырехсвязности, то множество пикселей периметра оказывается восьмисвязным. Эти замечания приводят к следующим определениям четырехсвязного периметра  $P_4$  и восьмисвязного периметра  $P_8$  области  $R$ .

**периметр:**

$$P_4 = \{(r, c) \in R | N_4(r, c) - R \neq \emptyset\}$$

$$P_8 = \{(r, c) \in R | N_8(r, c) - R \neq \emptyset\}$$

**Упражнение 3.10.** Восстановление области по периметру

Опишите алгоритм для генерации бинарного изображения области без отверстий по ее периметру.

**Упражнение 3.11.** Вычисление площади по периметру области

Разработайте алгоритм для вычисления площади области без отверстий по ее периметру. Можно ли решить эту задачу без восстановления бинарного изображения области?

Для вычисления длины  $|P|$  периметра  $P$  надо упорядочить пиксели  $P$  в последовательности  $P = \langle (r_o, c_o), \dots, (r_{K-1}, c_{K-1}) \rangle$ , в которой пиксели каждой последовательной пары являются соседними (в том числе первый и последний пиксели). Тогда *длина периметра*  $|P|$  определяется как **длина периметра**:

$$|P| = |\{k|(r_{k+1}, c_{k+1}) \in N_4(r_k, c_k)\}| + \sqrt{2}|\{k|(r_{k+1}, c_{k+1}) \in N_8(r_k, c_k) - N_4(r_k, c_k)\}| \quad (3.10)$$

где  $k+1$  вычисляется по модулю  $K$  ( $K$  равно длине последовательности пикселей). Таким образом, два вертикально или горизонтально смежных пиксела приведут к увеличению длины периметра на 1, а два диагонально смежных пиксела — к увеличению длины периметра примерно на 1.4.

С учетом определений площади  $A$  и периметра  $P$  можно ввести известное свойство округлости (circularity) области как квадрата длины периметра, деленного на площадь области.

**округлость (1):**

$$C_1 = \frac{|P|^2}{A} \quad (3.11)$$

К сожалению, применительно к цифровым образам фигур величина  $|P|^2/A$  принимает наименьшее значение не для окружностей (как в случае непрерывных образов плоских фигур), а для восьмиугольников или ромбов (в зависимости от того, как вычислялся периметр — как количество четырехсвязных граничных пикселей или как длина границы с приращением на 1 для вертикального или горизонтального перемещения и на  $\sqrt{2}$  для диагонального перемещения). Для решения этой проблемы в 1974 г. Харалик (Haralick) предложил второй вариант определения свойства округлости.

**округлость (2):**

$$C_2 = \frac{\mu_R}{\sigma_R} \quad (3.12)$$

где  $\mu_R$  и  $\sigma_R$  равны среднему значению и среднеквадратическому отклонению расстояния граничных пикселей от центра тяжести области. Формулы для вычисления этих величин приведены далее.

среднее радиальное расстояние:

$$\mu_R = \frac{1}{K} \sum_{k=0}^{K-1} \|(r_k, c_k) - (\bar{r}, \bar{c})\| \tag{3.13}$$

среднеквадратическое отклонение радиального расстояния:

$$\sigma_R = \left( \frac{1}{K} \sum_{k=0}^{K-1} [ \|(r_k, c_k) - (\bar{r}, \bar{c})\| - \mu_R ]^2 \right)^{1/2} \tag{3.14}$$

где множество пикселей  $(r_k, c_k)$ ,  $k = 0, \dots, K - 1$  принадлежит периметру  $P$  рассматриваемой области. Функция округлости  $C_2$  монотонно увеличивается по мере приближения цифрового образа фигуры к окружности и ведет себя похожим образом и для цифровых, и для непрерывных образов фигур.

На рис. 3.18 показаны некоторые из перечисленных основных свойств областей маркированного изображения, которое содержит три области в форме эллипса, прямоугольника и квадрата  $3 \times 3$ .

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	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	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0
2	2	2	2	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0
2	2	2	2	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0
2	2	2	2	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
2	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	2	2	2	0	0	3	3	3	0	0	0	0	0	0	0	0	0	0	0
2	2	2	2	0	0	3	3	3	0	0	0	0	0	0	0	0	0	0	0
2	2	2	2	0	0	3	3	3	0	0	0	0	0	0	0	0	0	0	0
2	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(a) Изображение после маркировки связных компонент

номер области	площадь области	центр. строка	центр. столбец	длина периметра	округлость 1	округлость 2	ср. рад. расст.	рад. откл.
1	44	6	11.5	21.2	10.2	15.4	3.33	.05
2	48	9	1.5	28	16.3	2.5	3.80	2.28
3	9	13	7	8	7.1	5.8	1.2	0.04

(b) Свойства трех областей

Рис. 3.18. Основные свойства областей изображения.



### Упражнение 3.12. Использование свойств

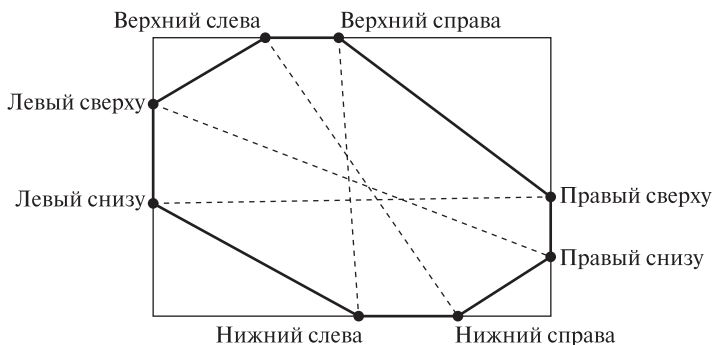
В наборе двумерных фигур есть треугольники, прямоугольники, восьмиугольники, окружности и эллипсы. Предложите способ распознавания этих фигур. Можете пользоваться операциями математической морфологии и рассмотренными выше свойствами областей.

#### описывающий прямоугольник и экстремальные точки:

Часто оказывается полезна информация о приблизительном местоположении области на изображении. С приблизительным местоположением области связано понятие *описывающего прямоугольника* (bounding box). Это прямоугольник, окружающий область и касающийся ее крайних точек сверху, снизу, слева и справа. Как показано на рис. 3.19, у области может быть 8 различных экстремальных пикселей: верхний справа, правый сверху, правый снизу, нижний справа, нижний слева, левый снизу, левый сверху и верхний слева. Каждая экстремальная точка имеет координату с предельным значением по строке или по столбцу. Под «предельным» подразумевается минимальное или максимальное значение координаты среди всех пикселей рассматриваемой области. Все экстремальные точки лежат на сторонах описывающего прямоугольника области.

Экстремальные точки образуют пары противоположащих точек: верхняя слева и нижняя справа; верхняя справа и нижняя слева; правая сверху и левая снизу; правая снизу и левая сверху. Каждая пара противоположащих экстремальных точек определяет ось. В качестве полезных свойств оси могут быть выбраны ее длина и направление.

Так как экстремальные точки формируются в результате оцифровки и пространственной дискретизации, то обычная формула евклидова расстояния будет давать немного заниженные значения расстояний. Например, рассмот-



**Рис. 3.19.** Восемь экстремальных точек области и нормально ориентированный описывающий прямоугольник этой области. Пунктирными отрезками показаны пары противоположащих экстремальных точек, образующие оси экстремальных точек фигуры.

рите случай вычисления расстояния между двумя горизонтально смежными пикселями. Расстояние от левого края левого пикселя до правого края правого пикселя равно 2, но расстояние между центрами пикселей равно 1. При вычислении расстояний на цифровых изображениях можно учесть поправку к евклидову расстоянию. Это приращение зависит от направления оси  $\theta$  и равно

$$Q(\theta) = \begin{cases} \frac{1}{|\cos \theta|} & : |\theta| < 45^\circ \\ \frac{1}{|\sin \theta|} & : |\theta| > 45^\circ \end{cases} \quad (3.15)$$

С учетом этой поправки, длина экстремальной оси между экстремальными точками  $[\mathbf{r}_1, \mathbf{c}_1]$  и  $[\mathbf{r}_2, \mathbf{c}_2]$  равна

**длина экстремальной оси:**

$$D = \sqrt{(r_2 - r_1)^2 + (c_2 - c_1)^2} + Q(\theta) \quad (3.16)$$

Для описания формы области часто применяются моменты второго порядка. Они обозначаются  $\mu_{rr}$ ,  $\mu_{rc}$  и  $\mu_{cc}$  определяются формулами:

**центральный момент второго порядка по строке:**

$$\mu_{rr} = \frac{1}{A} \sum_{(r,c) \in R} (r - \bar{r})^2 \quad (3.17)$$

**смешанный центральный момент второго порядка:**

$$\mu_{rc} = \frac{1}{A} \sum_{(r,c) \in R} (r - \bar{r})(c - \bar{c}) \quad (3.18)$$

**центральный момент второго порядка по столбцу:**

$$\mu_{cc} = \frac{1}{A} \sum_{(r,c) \in R} (c - \bar{c})^2 \quad (3.19)$$

Центральный момент  $\mu_{rr}$  может служить в качестве меры отклонения координат пикселей от координаты центра тяжести в направлении строк, а центральный момент  $\mu_{cc}$  — в направлении столбцов изображения. Центральный момент  $\mu_{rc}$  учитывает отклонение координат пикселя от координат центра тяжести по обоим координатным направлениям. Эти величины часто используются в качестве простых характеристик формы. Они инвариантны относительно двумерных преобразований переноса и масштабирования.

Центральные моменты второго порядка определены и имеют смысл для областей любой формы (так же, как и ковариационная матрица, которая определена и имеет смысл для любого двумерного распределения вероятности). Центральным моментам второго порядка можно дать геометрическую интерпретацию на примере эллиптической области.

Если область  $R$  имеет форму эллипса с центром в начале координат, то  $R$  можно представить в виде

$$R = \{(r, c) \mid dr^2 + 2erc + fc^2 \leq 1\} \quad (3.20)$$

Между коэффициентами уравнения эллипса  $d$ ,  $e$  и  $f$  и моментами второго порядка  $\mu_{rr}$ ,  $\mu_{rc}$  и  $\mu_{cc}$  существует связь. Она записывается в форме выражения

$$\begin{pmatrix} d & e \\ e & f \end{pmatrix} = \frac{1}{4(\mu_{rr}\mu_{cc} - \mu_{rc}^2)} \begin{pmatrix} \mu_{cc} - \mu_{rc} \\ -\mu_{rc} & \mu_{rr} \end{pmatrix} \quad (3.21)$$

Коэффициенты  $d$ ,  $e$  и  $f$  определяют длину большой и малой оси и ориентацию эллипса. Поэтому из приведенного выражения можно заключить, что моменты второго порядка  $\mu_{rr}$ ,  $\mu_{rc}$  и  $\mu_{cc}$  также задают эти характеристики эллипса. На изображениях форму эллипса часто принимают образы объектов круглой формы. Эллипсы также могут использоваться в качестве грубого приближенного представления других продолговатых объектов.

**Длина и направление осей эллипса\*** Для определения длины и ориентации большой и малой осей по центральным моментам второго порядка рассмотрим четыре различных случая.

1.  $\mu_{rc} = 0$  и  $\mu_{rr} > \mu_{cc}$

Большая ось направлена под углом  $-90^\circ$  против часовой стрелки относительно столбцов изображения. Ее длина равна  $4\mu_{rr}^{1/2}$ . Малая ось длиной  $4\mu_{cc}^{1/2}$  направлена параллельно столбцам изображения.

2.  $\mu_{rc} = 0$  и  $\mu_{rr} \leq \mu_{cc}$

Большая ось длиной  $4\mu_{cc}^{1/2}$  направлена параллельно столбцам изображения. Малая ось длиной  $4\mu_{rr}^{1/2}$  направлена под углом  $-90^\circ$  против часовой стрелки относительно столбцов.

3.  $\mu_{rc} \neq 0$  и  $\mu_{rr} \leq \mu_{cc}$

Большая ось направлена под углом

$$\operatorname{tg}^{-1} \left\{ \frac{-2\mu_{rc}}{\mu_{rr} - \mu_{cc} + [(\mu_{rr} - \mu_{cc})^2 + 4\mu_{rc}^2]^{1/2}} \right\}$$

против часовой стрелки относительно столбцов изображения. Ее длина равна

$$\left\{ 8 \left( \mu_{rr} + \mu_{cc} + [(\mu_{rr} - \mu_{cc})^2 + 4\mu_{rc}^2]^{1/2} \right) \right\}^{1/2}$$

Малая ось направлена под углом  $90^\circ$  против часовой стрелки относительно большой оси и имеет длину

$$\left[ 8 \left\{ \mu_{rr} + \mu_{cc} - [(\mu_{rr} - \mu_{cc})^2 + 4\mu_{rc}^2]^{1/2} \right\} \right]^{1/2}$$

4.  $\mu_{rc} \neq 0$  и  $\mu_{rr} > \mu_{cc}$ 

Большая ось направлена под углом

$$\operatorname{tg}^{-1} \frac{\left[ \left\{ \mu_{cc} + \mu_{rr} + \left[ (\mu_{cc} - \mu_{rr})^2 + 4\mu_{rc}^2 \right]^{1/2} \right\} \right]^{1/2}}{-2\mu_{rc}}$$

против часовой стрелки относительно столбцов изображения. Длина большой оси равна

$$\left[ 8 \left\{ \mu_{rr} + \mu_{cc} + \left[ (\mu_{rr} - \mu_{cc})^2 + 4\mu_{rc}^2 \right]^{1/2} \right\} \right]^{1/2}$$

Малая ось направлена под углом  $90^\circ$  против часовой стрелки относительно большой оси и имеет длину

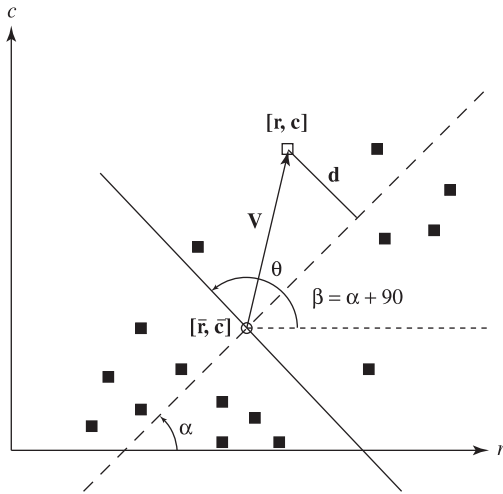
$$\left[ 8 \left\{ \mu_{rr} + \mu_{cc} - \left[ (\mu_{rr} - \mu_{cc})^2 + 4\mu_{rc}^2 \right]^{1/2} \right\} \right]^{1/2}$$

**Главные оси\*** У некоторых областей на изображениях (и у объектов) можно интуитивно выделить характерную «наилучшую» ось. Например, это карандаш или молоток, или символы 'I', '/' и '-'. Главная ось объекта на плоском изображении может быть найдена как ось, относительно которой пиксели области имеют наименьший момент второго порядка. По аналогии с механикой, эта ось является осью, относительно которой момент инерции объекта минимален. Если бы можно было привести пиксели области во вращение, то для вращения вокруг этой оси потребовались бы минимальные затраты энергии по сравнению с другими осями. Отметим, что момент инерции кругового диска относительно всех осей имеет одинаковое значение.

Известно, что ось, соответствующая минимальному моменту инерции (главная ось), проходит через центр тяжести  $(\bar{r}, \bar{c})$  множества пикселей области (эти пиксели можно рассматривать как материальные точки единичной массы). Далее будем пользоваться этим фактом. Во-первых, вычислим момент второго порядка множества точек относительно произвольной оси. Затем определим направление главной оси, которой соответствует наименьшее значение момента второго порядка.

Множество моментов относительно заданного набора осей может служить хорошим набором характерных признаков при распознавании объектов. Этот прием будет рассматриваться в следующей главе. Например, момент второго порядка символа 'I' относительно вертикальной оси, проходящей через его центр тяжести, очень мал, а для символов '/' и '-' имеет большее значение.

На рис. 3.20 показано множество пикселей и ось, направленная под углом  $\alpha$  относительно строк изображения. Перпендикуляр к этой оси направлен под углом  $\beta = \alpha + 90$  относительно строк изображения. Для вычисления момента второго порядка множества точек относительно оси необходимо вычислить



**Рис. 3.20.** Момент относительно оси вычисляется путем сложения квадратов расстояний между каждым пикселем области и осью.

сумму квадратов расстояний  $d$  для всех пикселей. Вычисленное значение нормируется путем деления на количество пикселей области. Нормировка нужна, чтобы получить характерный признак, не сильно зависящий от количества пикселей в рассматриваемой области. Отметим, что поскольку вычисляется сумма  $d^2$ , то изменение углов  $\alpha$  и  $\beta$  на  $\pm \pi$  не влияет на значение момента. Способ вычисления момента второго порядка приведен в виде формулы 3.22. Символ  $\circ$  означает скалярное произведение векторов. Скалярное произведение используется для вычисления проекции вектора  $\vec{V}$  на единичный вектор, направленный под углом  $\beta$ . Длина этой проекции равна  $d$ . С помощью трех параметров  $\bar{r}$ ,  $\bar{c}$  и  $\alpha$  может быть задана любая ось на плоскости.

**момент второго порядка относительно оси:\***

$$\begin{aligned}
 \mu_{\bar{r}, \bar{c}, \alpha} &= \frac{1}{A} \sum_{(r,c) \in R} d^2 \\
 &= \frac{1}{A} \sum_{(r,c) \in R} (\vec{V} \circ (\cos \beta, \sin \beta))^2 \\
 &= \frac{1}{A} \sum_{(r,c) \in R} ((r - \bar{r}) \cos \beta + (c - \bar{c}) \sin \beta)^2
 \end{aligned} \tag{3.22}$$

где  $\beta = \alpha + \pi/2$ .

---

**Упражнение 3.13.** Программа для вычисления характерных признаков множеств пикселей

Напишите программный модуль или класс на Си++ для хранения двумерных точек и выполнения перечисленных далее действий. Для хранения точек можно использовать структуру данных в виде множества с повторяющимися элементами (bag).

- Инициализация структуры данных для хранения двумерных точек  $[\mathbf{r}, \mathbf{c}]$ .
- Добавление точки  $[\mathbf{r}, \mathbf{c}]$  в структуру данных.
- Вычисление центра тяжести точек.
- Вычисление центральных моментов второго порядка.
- Вычисление описывающего прямоугольника.
- Вычисление главных осей и моментов второго порядка относительно этих осей.

---

**Упражнение 3.14.** Программа для вычисления характерных признаков изображений

После разработки модуля для вычисления характерных признаков в предыдущем упражнении дополните его функциями для вычисления моментов второго порядка относительно горизонтальной, вертикальной и диагональной осей, проходящих через центр тяжести множества точек. Таким образом, будет обеспечиваться вычисление 5 различных моментов второго порядка. Сформируйте набор тестовых бинарных изображений размерами  $20 \times 20$ , представляющих цифры от 0 до 9 (можете воспользоваться подходящими готовыми изображениями). Напишите программу для обработки изображений цифр и вычисления для них пяти моментов второго порядка. Выясните, достаточно ли пяти различных моментов для построения алгоритма распознавания цифр.

Приведенная выше формула позволяет вычислить набор из нескольких моментов для использования их в качестве характеристик формы множества точек. Например, моменты относительно вертикальной, горизонтальной и диагональной осей полезны в задаче классификации алфавитных символов стандартного написания (без поворотов). Оси, которым соответствуют наименьшее (и наибольшее) значения момента инерции, представляют собой характеристики множества точек, инвариантные относительно преобразований переноса и поворота. Ось, соответствующая минимальному моменту инерции, можно определить поиском минимального значения  $\mu_{\bar{r}, \bar{c}, \alpha}$ . Учитывая, что главная ось проходит через центр тяжести, при поиске минимума можно продифференцировать эту формулу только по  $\alpha$  и найти значение угла  $\hat{\alpha}$ , минимизирующее значение  $\mu_{\bar{r}, \bar{c}, \alpha}$ .

главная ось, соответствующая минимальному моменту второго порядка:\*

$$\begin{aligned} \operatorname{tg} 2\hat{\alpha} &= \frac{2 \sum (r - \bar{r})(c - \bar{c})}{\sum (r - \bar{r})(r - \bar{r}) - \sum (c - \bar{c})(c - \bar{c})} \\ &= \frac{\frac{1}{A} 2 \sum (r - \bar{r})(c - \bar{c})}{\frac{1}{A} \sum (r - \bar{r})(r - \bar{r}) - \frac{1}{A} \sum (c - \bar{c})(c - \bar{c})} \\ &= \frac{2 \mu_{rc}}{\mu_{rr} - \mu_{cc}} \end{aligned} \quad (3.23)$$

Угол  $\alpha$  принимает два экстремальных значения, одно минимальное и одно максимальное. Они отличаются на 90 градусов. Способ различения двух этих значений уже рассматривался выше при обсуждении направления осей эллипса. В действительности формула (3.23) позволяет найти эллипс, который можно считать приближенным представлением множества точек. Отметим, что в случае симметричных объектов, например, квадратов или окружностей, при вычислении по формуле (3.23) может возникнуть ошибка деления на 0. В таком случае также можно применить анализ, аналогичный анализу значений моментов для эллипса.

**Упражнение 3.15.** Вычисление предельных значений момента инерции

Продифференцируйте выражение 3.22 и покажите, каким образом для определения направления главных осей получается выражение 3.23.

**Упражнение 3.16.** Проверка прохождения главных осей через центр тяжести

Проверьте, что ось, соответствующая наименьшему моменту инерции, проходит через центр тяжести. Выполните это самостоятельно или обратитесь к справочникам по статистике или механике, указанным в дополнительной литературе или каким-либо другим.

### 3.7 ГРАФЫ СМЕЖНОСТИ ОБЛАСТЕЙ БИНАРНОГО ИЗОБРАЖЕНИЯ

В задачах анализа изображений могут учитываться не только свойства отдельных областей, но и взаимосвязь между группами областей. *Смежность областей* (region adjacency) представляет собой один из простейших, но в то же время полезный тип взаимосвязей. Две области называются смежными, если некоторый пиксел одной области является соседним для пиксела другой области. На бинарных изображениях существуют области всего двух типов: области переднего плана и фоновые области. Все области переднего плана являются смежными с фоновыми областями, но не смежными друг с другом.

Если фон образует одну связную область, то на этом вычисление свойств смежности областей заканчивается.

Предположим, что области переднего плана содержат отверстия. Область каждого отверстия относится к фону. Выполнив маркировку связных компонент для пикселей переднего плана, получим маркированное изображение, на котором все области переднего плана имеют числовые метки, а пикселям всех фоновых областей присвоены нулевые значения. Аналогичным образом можно выполнить маркировку связных компонент применительно к фоновым пикселям. Тогда всем фоновым областям также будут присвоены метки. Вероятно, одна из этих областей с наибольшей площадью будет начинаться в левом верхнем углу изображения. Этой области можно присвоить специальную метку, например, 0. Остальные фоновые области меньшей площади будут соответствовать отверстиям в областях переднего плана.

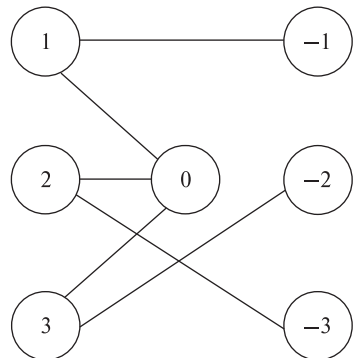
По двум маркированным изображениям с помеченными областями переднего плана и фона можно определить, какие фоновые области являются смежными для каждой области переднего плана (или наоборот). Структура данных для хранения информации о смежности пар областей называется *графом смежности областей* (region adjacency graph, RAG). Эта структура может использоваться для представления смежности областей переднего плана и фона на бинарных изображениях или в общем случае для представления смежности областей в процессе сегментации полутоновых или цветных изображений.

**21 Определение.** Граф, в котором каждый узел соответствует некоторой области изображения, а ребра соединяют пары узлов, представляющие смежные области, называется **графом смежности областей**.

Пример графа смежности для областей переднего плана и фона на бинарном изображении показан на рис. 3.21. Области переднего плана промар-

0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	0	2	2	0
0	1	-1	-1	-1	1	0	2	2	0
0	1	1	1	1	1	0	2	2	0
0	0	0	0	0	0	0	2	2	0
0	3	3	3	0	2	2	2	2	0
0	3	-2	3	0	2	-3	-3	2	0
0	3	-2	3	0	2	-3	-3	2	0
0	3	3	3	0	2	2	2	2	0
0	0	0	0	0	0	0	0	0	0

(a) Маркированное изображение с помеченными областями переднего плана и фона



(b) Граф смежности областей

**Рис. 3.21.** Маркированное изображение и граф смежности областей этого изображения.



кированы, как обычно, положительными числами. Метки фоновых областей начинаются с 0 для области наибольшей площади, начинающейся в левом верхнем углу изображения, и уменьшаются в область отрицательных чисел для областей, соответствующих отверстиям в областях переднего плана.

Алгоритм для построения графа смежности областей довольно прост. Он обрабатывает изображение порциями по две строки. Алгоритм обнаруживает горизонтальную и вертикальную смежность и, если применяется восьмисвязность, также диагональную смежность точек с различными метками. При обнаружении смежности ранее несмежных областей в граф добавляются новые ребра. При оценке эффективности этого алгоритма возникают два вопроса. Во-первых, размер структуры данных. На изображении могут присутствовать десятки тысяч меток. В таком случае может быть невозможно или нерационально с точки зрения использования виртуальной памяти, хранить всю графовую структуру данных в памяти целиком. Второй вопрос касается временных затрат. При поточечной обработке изображения одно и то же отношение смежности (т. е. для одних и тех же пар областей) будет обнаруживаться многократно. Добавление ребра в структуру данных желательно выполнять один раз для каждой пары смежных областей. Эти вопросы подробнее рассматриваются в упражнении 3.17.

---

### Упражнение 3.17. Эффективное построение графа смежности

Разработайте структуру данных для хранения информации о смежности в процессе построения графа смежности областей. Спроектируйте алгоритм для построения графа по произвольному маркированному изображению с минимальным количеством обращений к структуре данных. Обсудите, как можно было бы хранить граф смежности в окончательном виде в долговременной памяти (на диске) и как можно было бы выполнять обработку в случаях, когда граф смежности слишком велик и не помещается в памяти целиком в процессе построения.

## 3.8 ПОРОГОВАЯ БИНАРИЗАЦИЯ ПОЛУТОНОВЫХ ИЗОБРАЖЕНИЙ

Бинарные изображения можно получить из полутоновых изображений посредством пороговой бинаризации. При выполнении этой операции часть пикселей выбирается в качестве пикселей переднего плана, представляющих объекты интереса, а остальные — в качестве фоновых пикселей. Зная распределение значений яркости на данном изображении, некоторые значения можно выбрать в качестве порогов, разделяющих пиксели на группы. В простейшем случае выбирается одно пороговое значение  $t$ . Все пиксели с яркостью больше или равной  $t$  становятся пикселями переднего плана, а остальные — фоновыми. Эта операция называется *верхней пороговой бинаризацией* (threshold above). Существуют и другие разновидности пороговой бинаризации. При выполнении *нижней пороговой бинаризации* (threshold below) пикселями переднего плана

становятся пиксели с яркостью меньше или равной  $t$ . При *пороговой бинаризации по диапазону* (threshold inside) указываются два пороговых значения — верхнее и нижнее. Пиксели с яркостью в диапазоне между пороговыми значениями выбираются в качестве пикселей переднего плана. Обратный критерий выбора пикселей переднего плана применяется при выполнении *внешней пороговой бинаризации* (threshold outside). Главным вопросом, возникающим при выполнении всех этих простейших разновидностей операции бинаризации, является выбор пороговых значений.

### 3.8.1 Выбор порога бинаризации по гистограмме

Во многих интерактивных пакетах обработки изображений предусматривается возможность указания порогов бинаризации пользователем. Но в задачах автоматического анализа изображений было бы желательно вычислять пороговые значения автоматически. В качестве исходных данных для выбора порога бинаризации можно рассмотреть *гистограмму* распределения значений яркости на полутоновом изображении. Для краткости будем называть ее гистограммой изображения.

**22 Определение.** Гистограмма  $h$  полутонового изображения  $I$  задается выражением:

$$h(m) = |\{(r, c) \mid I(r, c) = m\}|,$$

где  $m$  соответствует интервалам значений яркости.

На рис. 3.22 показано изображение сдавленной вишни и гистограмма этого изображения. На гистограмме заметны две моды, соответствующие сдавленной и нормальной частям вишни.

Вычислить гистограмму можно с помощью простой процедуры обработки с использованием массива. Обозначим через  $\mathbf{H}$  массив с индексами от 0 до  $\text{MaxVal}$ , где 0 соответствует наименьшему возможному значению яркости,



(a) Изображение сдавленной вишни



(b) Гистограмма изображения вишни

**Рис. 3.22.** Бимодальная гистограмма изображения слегка сдавленной вишни. Одна мода соответствует сдавленной области вишни, а вторая — нормальной области. (С любезного разрешения Patchrawat Uthaisombut.)

а **MaxVal** — наибольшему. Пусть **I** — это двумерный массив изображения с индексами строк от 0 до **MaxRow** и индексами столбцов от 0 до **MaxCol** (как в предыдущих разделах). Возможная реализация процедуры расчета гистограммы приведена ниже:

**Вычисление гистограммы **H** для полутонового изображения **I**.**

```

procedure histogram(I,H);
{
  “Инициализация карманов гистограммы нулевыми значениями.”
  for i := 0 to MaxVal
    H[i] := 0;
  “Накопительное вычисление значений.”
  for L := 0 to MaxRow
    for P := 0 to MaxCol
      {
        grayval := I[r,c];
        H[grayval] := H[grayval] + 1;
      } ;
}

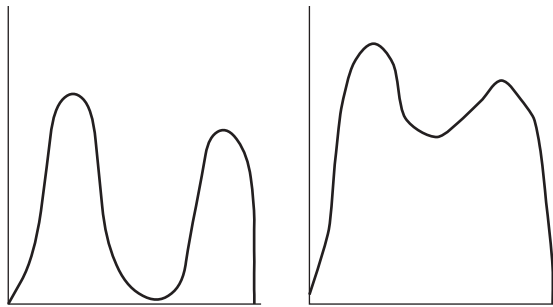
```

**Алгоритм 3.7.** Построение гистограммы изображения.

В приведенной процедуре построения гистограммы предполагается, что каждому возможному значению яркости на изображении соответствует один карман гистограммы. Иногда вместо этого желательно сгруппировать несколько значений яркости в один карман, например, для отображения гистограммы изображения с большим количеством градаций яркости. Для этой цели процедуру расчета гистограммы легко модифицировать, чтобы вычислять номер кармана как функцию значения яркости. Если *binsize* — это количество значений яркости, соответствующих одному карману гистограммы, то округленное до целого значение  $\text{grayval} / \text{binsize}$  будет равно индексу соответствующего кармана.

Можно реализовать процедуры для автоматического обнаружения пиков и впадин на заданной гистограмме. В простейшем случае для этого выполняется поиск одиночного порогового значения, разделяющего пиксели изображения на светлые и темные. Если распределения темных и светлых пикселей на гистограмме отделены друг от друга значительным промежутком, то такая гистограмма оказывается бимодальной. Одна мода соответствует темным пикселям, а другая — светлым.

При малом перекрытии (или при отсутствии перекрытия) распределений темных и светлых пикселей в качестве порога можно выбрать любое значение



(a) Бимодальная гистограмма с отдельными модами (b) Бимодальная гистограмма с перекрывающимися модами

**Рис. 3.23.** Примеры гистограмм двух изображений. (a) На гистограмме присутствуют две легко разделяемые моды. (b) Гистограмма содержит перекрывающиеся моды, которые сложнее разделить, чтобы найти подходящий порог бинаризации.

в области впадины между двумя модами гистограммы (рис. 3.23 (a)). Однако при сильном перекрытии распределений светлых и темных пикселей выбор порога усложняется, так как впадина на гистограмме исчезает, и два распределения начинают сливаться в одно (рис. 3.23 (b)).

### 3.8.2 Автоматический выбор порога бинаризации: метод Оцу\*

Для автоматического выбора порога бинаризации было разработано несколько различных методов. В данном разделе будет описан метод Оцу (Otsu). Выбор порога в этом методе основан на минимизации внутригрупповой дисперсии двух групп пикселей, разделяемых оператором пороговой бинаризации. При рассмотрении метода будем считать, что гистограмма задана в виде дискретной функции распределения вероятности  $P$ . Значения  $P(0), \dots, P(I)$  представляют собой наблюдаемые вероятности значений яркости  $0, \dots, I$ ;  $P(i) = |\{(r, c) \mid \text{Image}(r, c) = i\}| / |R \times C|$ , где  $R \times C$  равно площади изображения. Если гистограмма бимодальная, то задача выбора порога бинаризации заключается в определении наилучшего порога  $t$ , отделяющего две моды гистограммы друг от друга. Каждому пороговому значению  $t$  соответствует дисперсия группы значений, меньших или равных  $t$ , и дисперсия группы значений, больших  $t$ . Оцу предложил определение наилучшего порога бинаризации как такого значения, для которого взвешенная сумма *внутригрупповых дисперсий* (within-group variances) минимальна. Веса равны суммарным вероятностям соответствующих групп.

Для пояснения критерия выбора порога на основе внутригрупповой дисперсии рассмотрим ситуацию, связанную с формированием учебных групп в лыжной школе. Обучающимся был предложен предварительный тест для

оценки лыжных навыков. По результатам тестов была построена гистограмма. Гистограмма оказалась бимодальной. На ней выделяются распределения результатов опытных и начинающих лыжников. Содержание занятий для опытных лыжников окажется сложным для новичков, а занятия для начинающих будут скучными для более опытных. Чтобы решить эту проблему, инструктор по результатам теста может разделить обучающихся на две непересекающиеся группы с примерно одинаковыми результатами. Вопрос лишь в том, какое значение результата выбрать в качестве критерия для разделения групп. В идеале группы начинающих и опытных лыжников должны показать результаты с унимодальным нормальным распределением. На общей гистограмме будут заметны два нормальных распределения, каждое в окрестности собственного среднего значения. Это могло бы означать, что две группы являются однородными и явно отделены друг от друга.

Мерой однородности групп служит дисперсия. Чем меньше разброс значений внутри группы, тем меньше значение дисперсии. Возможным способом выбора критерия для разделения групп может быть выбор такого значения, при котором взвешенная сумма внутригрупповых дисперсий будет минимальна. В этом критерии предполагается существование однородных групп (состоящих из примерно одинаковых значений). Второй способ заключается в выборе значения, при котором квадрат разности между средними значениями полученных групп принимает максимальное значение. Эта разность связана с межгрупповой дисперсией (between-group variance).

Оба критерия разделения приводят к аналогичным результатам, так как сумма внутригрупповых и межгрупповой дисперсии постоянна.

Обозначим через  $\sigma_W^2$  взвешенную сумму внутригрупповых дисперсий. Эта сумма называется *общей внутригрупповой дисперсией*. Через  $\sigma_1^2(t)$  обозначим дисперсию первой группы, состоящей из значений, меньших или равных  $t$ .  $\sigma_2^2(t)$  будет дисперсией второй группы, которая состоит из значений, больших  $t$ . Обозначим через  $q_1(t)$  суммарную вероятность первой группы и через  $q_2(t)$  — суммарную вероятность второй группы. Средние значения первой и второй групп обозначим  $\mu_1(t)$  и  $\mu_2(t)$ . Тогда общая внутригрупповая дисперсия  $\sigma_W^2$  определяется как

$$\sigma_W^2(t) = q_1(t) \sigma_1^2(t) + q_2(t) \sigma_2^2(t) \quad (3.24)$$

где

$$\begin{aligned} q_1(t) &= \sum_{i=1}^t P(i) \\ q_2(t) &= \sum_{i=t+1}^I P(i) \\ \mu_1(t) &= \sum_{i=1}^t i P(i) / q_1(t) \end{aligned} \quad (3.25)$$

$$\mu_2(t) = \sum_{i=t+1}^I i P(i)/q_2(t) \quad (3.26)$$

$$\sigma_1^2(t) = \sum_{i=1}^t [i - \mu_1(t)]^2 P(i)/q_1(t)$$

$$\sigma_2^2(t) = \sum_{i=t+1}^I [i - \mu_2(t)]^2 P(i)/q_2(t) \quad (3.27)$$

Наилучшее значение порога можно определить простым перебором всех возможных значений  $t$  для поиска значения  $t$ , минимизирующего  $\sigma_W^2(t)$ . Во многих ситуациях область перебора можно уменьшить до промежутка между двумя модами гистограммы. Однако затраты на обнаружение мод сопоставимы с поиском значения, разделяющего распределения в окрестности этих мод.

Между внутригрупповой дисперсией  $\sigma_W^2(t)$  и полной дисперсией  $\sigma^2$  существует взаимосвязь, которая не зависит от порогового значения. Полная дисперсия равна

$$\sigma^2 = \sum_{i=1}^I (i - \mu)^2 P(i), \quad \text{где} \quad \mu = \sum_{i=1}^I i P(i)$$

Связь полной и внутригрупповой дисперсий позволяет сократить вычислительные затраты на определение наилучшего порога. Переписывая выражение для  $\sigma^2$ , получим

$$\begin{aligned} \sigma^2 &= \sum_{i=1}^t [i - \mu_1(t) + \mu_1(t) - \mu]^2 P(i) + \sum_{i=t+1}^I [i - \mu_2(t) + \mu_2(t) - \mu]^2 P(i) \\ &= \sum_{i=1}^t \{ [i - \mu_1(t)]^2 + 2[i - \mu_1(t)][\mu_1(t) - \mu] + [\mu_1(t) - \mu]^2 \} P(i) \\ &\quad + \sum_{i=t+1}^I \{ [i - \mu_2(t)]^2 + 2[i - \mu_2(t)][\mu_2(t) - \mu] + [\mu_2(t) - \mu]^2 \} P(i) \end{aligned}$$

Но

$$\sum_{i=1}^t [i - \mu_1(t)][\mu_1(t) - \mu] P(i) = 0$$

$$\sum_{i=t+1}^I [i - \mu_2(t)][\mu_2(t) - \mu] P(i) = 0$$

Так как

$$\begin{aligned}
 q_1(t) &= \sum_{i=1}^t P(i) \quad \text{и} \quad q_2(t) = \sum_{i=t+1}^I P(i) \\
 \sigma^2 &= \sum_{i=1}^t [i - \mu_1(t)]^2 P(i) + [\mu_1(t) - \mu]^2 q_1(t) \\
 &\quad + \sum_{i=t+1}^I [i - \mu_2(t)]^2 P(i) + [\mu_2(t) - \mu]^2 q_2(t) \\
 &= [q_1(t) \sigma_1^2(t) + q_2(t) \sigma_2^2(t)] \\
 &\quad + \{q_1(t) [\mu_1(t) - \mu]^2 + q_2(t) [\mu_2(t) - \mu]^2\} \quad (3.28)
 \end{aligned}$$

Первое слагаемое в скобках равно внутригрупповой дисперсии  $\sigma_W^2$ . Это просто сумма взвешенных дисперсий каждой группы. Второе слагаемое в скобках называется межгрупповой дисперсией  $\sigma_B^2$ . Это сумма взвешенных квадратов расстояний между средними значениями каждой группы и средним значением всего распределения. Выражение межгрупповой дисперсии можно еще упростить. Отметим, что среднее значение всего распределения  $\mu$  можно записать в виде

$$\mu = q_1(t) \mu_1(t) + q_2(t) \mu_2(t) \quad (3.29)$$

С помощью (3.29) исключим  $\mu$  из (3.28) и подставим  $1 - q_1(t)$  вместо  $q_2(t)$ . После простых преобразований получим

$$\sigma^2 = \sigma_W^2(t) + q_1(t)[1 - q_1(t)] [\mu_1(t) - \mu_2(t)]^2$$

Так как полная дисперсия  $\sigma^2$  не зависит от  $t$ , то значением  $t$ , минимизирующим  $\sigma_W^2(t)$ , будет такое значение  $t$ , при котором достигается максимум межгрупповой дисперсии  $\sigma_B^2(t)$ ,

$$\sigma_B^2(t) = q_1(t) [1 - q_1(t)] [\mu_1(t) - \mu_2(t)]^2 \quad (3.30)$$

Для определения  $t$ , максимизирующего  $\sigma_B^2(t)$ , все необходимые величины должны быть вычислены согласно (3.25) и (3.27). Вычисления этих величин не обязательно независимо выполнять для каждого значения  $t$ . Существует взаимосвязь между значениями этих величин для  $t$  и для следующего значения  $t + 1$ . Непосредственно из (3.25) получаем рекуррентное соотношение

$$q_1(t + 1) = q_1(t) + P(t + 1) \quad (3.31)$$

с начальным значением  $q_1(1) = P(1)$ .

Из (3.26) получаем рекуррентное соотношение

$$\mu_1(t + 1) = \frac{q_1(t) \mu_1(t) + (t + 1)P(t + 1)}{q_1(t + 1)} \quad (3.32)$$

с начальным значением  $\mu_1(0) = 0$ . В результате, из (3.29) получаем

$$\mu_2(t+1) = \frac{\mu - q_1(t+1) \mu_1(t+1)}{1 - q_1(t+1)} \quad (3.33)$$

При проектировании алгоритмов автоматического определения порога бинаризации обычно делаются некоторые предположения относительно формы распределения значений яркости пикселей изображения. Поэтому такие алгоритмы хорошо работают на изображениях, которые удовлетворяют принятым в алгоритмах предположениям. В методе Оцу (Otsu) распределение значений яркости считается бимодальным. Если изображение примерно удовлетворяет этому ограничению, то алгоритм даст хорошие результаты. Если же гистограмма изображения далека от бимодальной, то результаты могут оказаться практически бесполезными. На рис. 3.24 демонстрируется применение метода Оцу к полутоновому изображению нескольких игрушечных блоков, показанному на рис. 3.24 (a). Алгоритм Оцу вычислил пороговое значение 93 при диапазоне значений яркости от 0 до 255. Пикселы со значениями меньше и больше порога показаны, соответственно, на рис. 3.24 (b) и (c). При таком пороге выделенными оказываются только очень темные области изображения.



**Рис. 3.24.** Полутоновое изображение и пикселы со значениями яркости меньше и больше порогового значения 93 (эти пикселы выделены белым цветом). Пороговое значение было определено методом Оцу. (Исходное изображение получено с любезного разрешения John Illingworth, Ata Etamadi.)

Если яркость изображения сильно зависит от положения пикселей, например, изображение в целом светлее в левом верхнем углу и темнее в правом нижнем, то больший смысл может иметь использование локальных порогов бинаризации для различных фрагментов изображения. Этот подход иногда называется *динамической пороговой бинаризацией* (dynamic thresholding). В некоторых задачах форма и размеры искоемых объектов известны заранее. В таком случае возможно применение *пороговой бинаризации* на основе *априорной информации* (knowledge-based thresholding). При этом считается, что местоположение областей интереса заранее известно, и с учетом этого выбираются пороговые значения, дающие наилучшие результаты бинаризации. В заключение следует отметить, что некоторые изображения оказываются непригодными



для бинаризации и для обнаружения объектов на них необходимо применять другие методы.

---

### Упражнение 3.18. Автоматическое определение порога бинаризации

Напишите программу для автоматического определения порога бинаризации методом Оцу. Проверьте работу программы на нескольких сканированных изображениях различных сцен.

## 3.9 ЛИТЕРАТУРА

Для маркировки связных компонент разработано большое количество различных алгоритмов, многие из которых адаптированы для специфических задач. В книге [Tanimoto, 1990] считается, что изображение помещается в памяти и может быть обработано простым рекурсивным алгоритмом, который просматривает компоненты по очереди, но при этом способен перемещаться в пределах всего изображения. Другие алгоритмы были разработаны в расчете на большие изображения, которые могут не поместиться в памяти целиком. Такие алгоритмы могут обрабатывать по две строки изображения на каждом шаге. В статье [Rosenfeld, Pfaltz, 1966] описан двухпроходный алгоритм, в котором используется глобальная таблица эквивалентности. Этот алгоритм иногда называется классическим алгоритмом маркировки связных компонент. В работе [Lumia, Shapiro, Zuniga, 1983] предложен другой двухпроходный алгоритм, в котором применяется локальная таблица эквивалентности, позволяющая избежать проблем со страничной адресацией виртуальной памяти. В работе [Danielsson, Tanimoto, 1983] рассмотрен алгоритм для параллельных компьютеров. В алгоритмах, хранящих сведения о классах эквивалентности, может применяться структура данных для объединения-поиска [Tarjan, 1975]. Эта структура данных предназначена для эффективной реализации операций объединения множеств и установления принадлежности элемента множеству.

Первое систематическое изложение теории математической морфологии представлено в книге [Serra, 1982]. В статье [Sternberg, 1985] рассматривается параллельная конвейерная архитектура для быстрого аппаратного выполнения морфологических операций и ее применение для обработки медицинских изображений и в ряде задач промышленного машинного зрения. Эти операции бинарной морфологии были расширены на случай обработки полутоновых изображений. Эти операции стали стандартными операциями фильтрации изображений. Статья [Haralick, Sternberg, Zhuang, 1987] является учебной статьей по бинарной и полутоновой морфологии, в которой продемонстрировано значение этих операций широкому кругу специалистов в области компьютерного зрения. В работе [Shapiro, MacDonald, Sternberg, 1987] показано, что морфологическое обнаружение характерных признаков может применяться для распознавания объектов.

Автоматическое определение порога бинаризации рассматривается во многих статьях. Метод, описанный в данной главе, был разработан Оцу [Otsu, 1979]. Ряд других методов представлен в работах [Kittler, Illingworth, 1986] и [Cho, Haralick, Yi, 1989]. В статье [Sahoo и др., 1988] дан общий обзор методов пороговой бинаризации.

1. Tanimoto, S. L. 1990. *The Elements of Artificial Intelligence Using Common LISP*. W. H. Freeman and Company, New York.
2. Rosenfeld, A., and J. L. Pfaltz. 1966. Sequential operations in digital picture processing. *J. Assoc. Comput. Machinery*, v. 13:471–494.
3. Lumia, R., G. Shapiro, and O. Zuniga. 1983. A new connected components algorithm for virtual memory computers. *Comput. Vision, Graphics, and Image Proc.*, v. 22: 287–300.
4. Danielsson, P.-E., and S. L. Tanimoto. 1983. Time complexity for serial and parallel propagation in images. In *Architecture and Algorithms for Digital Image Processing*, A. Oosterlinck and P.-E. Danielsson, eds. *Proc. SPIE*, v. 435:60–67.
5. Tarjan, R. E. 1975. Efficiency of a good but not linear set union algorithm. *J. Assoc. Comput. Machinery*, v. 22:215–225.
6. Serra, J. 1982. *Image Analysis and Mathematical Morphology*. Academic Press, New York.
7. Sternberg, S. R. 1985. An overview of image algebra and related architectures. *Integrated Technology for Parallel Image Processing*. Academic Press, London, 79–100.
8. Sternberg, S. R. 1986. Grayscale morphology. *Comput. Vision, Graphics, and Image Proc.*, v. 35:333–355.
9. Haralick, R. M., S. R. Sternberg, and X. Zhuang. 1987. Image analysis using mathematical morphology. *IEEE Trans. Pattern Analysis and Machine Intelligence*, v. PMI-9:523–550.
10. Shapiro, L. G., R. S. MacDonald, and S. R. Sternberg. 1987. Ordered structural shape matching with primitive extraction by mathematical morphology. *Pattern Recog.*, v. 20(1)75–90.
11. Haralick, R. M. 1974. A measure of circularity of digital figures. *IEEE Trans. Syst., Man, and Cybern.*, v. SMC-4:394–396.
12. Otsu, N. 1979. A threshold selection method from gray-level histograms. *IEEE Trans. Syst., Man and Cybern.*, v. SMC-9:62–66.
13. Kittler, J., and J. Illingworth. 1986. Minimum error thresholding. *Pattern Recog.*, v. 19:41–47.
14. Cho, S., R. M. Haralick, and S. Yi. 1989. Improvement of Kittler and Illingworth's minimum error thresholding. *Pattern Recog.*, v. 22:609–617.
15. Sahoo, P. K., and others. 1988. A survey of thresholding techniques. *Comput. Vision, Graphics, and Image Proc.*, v. 41:233–260.

# Основные понятия распознавания образов

В данной главе приведен краткий обзор методов распознавания объектов. Эти методы применяются для распознавания объектов на изображениях, но также могут использоваться применительно к данным, полученным иными способами. В качестве основного подхода выбран подход, согласно которому распознаваемый образ может быть представлен в виде вектора измерений. Рассматриваются несколько примеров, среди которых центральное место занимает задача распознавания символов. Читатель также будет ознакомлен с некоторыми простыми методами, позволяющими научить машину распознавать объекты посредством анализа последовательности эталонных образцов. После изучения первых четырех глав книги читатель получит представление об устройстве ряда законченных систем компьютерного зрения и сможет проводить эксперименты по разработке набора алгоритмов, достаточных для решения некоторых простых, но тем не менее реальных задач.

## 4.1 ЗАДАЧИ РАСПОЗНАВАНИЯ ОБРАЗОВ

Во многих практических задачах необходимо принять некоторое решение относительно содержания изображения или классифицировать присутствующий на изображении объект. Например, пользователь портативного компьютера может в качестве входных данных ввести рукописные символы. В данном случае, если рассматривать только английские символы, то возможно  $m = 128$  различных ASCII-символов, и каждый рукописный объект требуется в результате классификации отнести к одному из  $m$  классов. Обратимся к рис. 4.1. Классификация объекта — является ли он символом 'A', '8' и т. п. — может основываться на характерных признаках оптического изображения этого объекта или, возможно, на анализе значений давления ручки в процессе рисования, которые также можно представить в виде изображения. Процесс классификации может окончиться неудачей вследствие некорректного написания символа или потому, что человек написал неизвестный системе символ.

00000000000000000000	00000000000000000000
00000000010000000000	00000000011110000000
00000000011000000000	000000001100001100000
00000000010100000000	0000000110000000110000
000000001100110000000	000001000000000010000
000000001000010000000	000011000000000011000
000000010000010000000	000010000000000001000
000001100000001000000	0000100000000000011000
000001000000001000000	0000110000000000010000
000001000000001100000	0000010000000000110000
000010000000001000000	000001110000000100000
000011001111111100000	000000111001111000000
000011111110000010000	000000001111000000000
000110000000000011000	000000110001110000000
000100000000000001000	000001100000001100000
000100000000000001100	000011000000000110000
001100000000000000100	000110000000000011000
001100000000000000110	001100000000000001000
001000000000000000010	001000000000000001100
010000000000000000010	000100000000000011000
011000000000000000010	000110000000000010000
010000000000000000000	0000100000000000110000
000000000000000000000	000011100000011100000
000000000000000000000	00000011111110000000
000000000000000000000	000000000000000000000

**Рис. 4.1.** Бинарные изображения символов 'A' и '8'.

Обычно при разработке системы предусматривается класс выбросов, в который помещаются подобные объекты. Изображения, помещенные в класс выбросов, позднее могут быть повторно обработаны на более высоком уровне, могут привести к формированию нового класса, или могут быть просто записаны в исходном виде для последующего отображения.

Представьте автоматический банкомат, в котором камера используется для идентификации личности клиента. Изображение лица проверяемого человека сравнивается с сохраненным изображением или с изображениями, связанными со счетом этого человека и хранящимися либо в компьютерной сети, либо на самой банковской карте.

**23 Определение.** Процесс сопоставления экземпляра объекта с одним прототипом (моделью) объекта или с описанием класса называется **верификацией**.

В другой задаче, упоминавшейся в упражнениях в гл. 1, система распознавания в продуктовом магазине должна классифицировать фрукты и овощи, помещенные на контрольные весы. Классы будут представлять элементы множества всех распознаваемых разновидностей продукции, например яблоки Ида, яблоки Фуджи, шпинат, огурцы, грибы, и т. п. Для каждого класса хранится название и цена за килограмм<sup>1</sup>.

Для слова «распознавание» близким по значению является слово «узнавание». Система распознавания должна обладать памятью для запоминания

<sup>1</sup> Такая система, названная *Veggie Vision*, была разработана фирмой IBM. См. гл. 16.

объектов, которые она будет распознавать. Представление объектов в памяти может быть встроенным, например, модель прыгающей лягушки; или может быть сформировано по большому количеству эталонных образцов, что напоминает изучение алфавита в начальной школе; или это представление может быть запрограммировано в терминах специфических признаков изображения, что напоминает, например, обучение ребенка отличать пожарные машины от автобусов. Вопросы распознавания и обучения образам являются глубокими темами, представляющими интерес для когнитивной психологии, распознавания образов и компьютерного зрения. В данной главе выбран практический подход и описаны методы, которые успешно применялись в прикладных задачах. В конце главы приведен ряд ссылок на теоретическую литературу.

## 4.2 ОБЩАЯ МОДЕЛЬ КЛАССИФИКАЦИИ

Рассмотрим элементы общей модели **классификации**. Этот материал носит скорее практический, а не теоретический, характер и построен так, чтобы системы распознавания образов можно было проектировать и реализовывать с использованием независимо разработанных аппаратных и программных модулей.

### 4.2.1 Классы

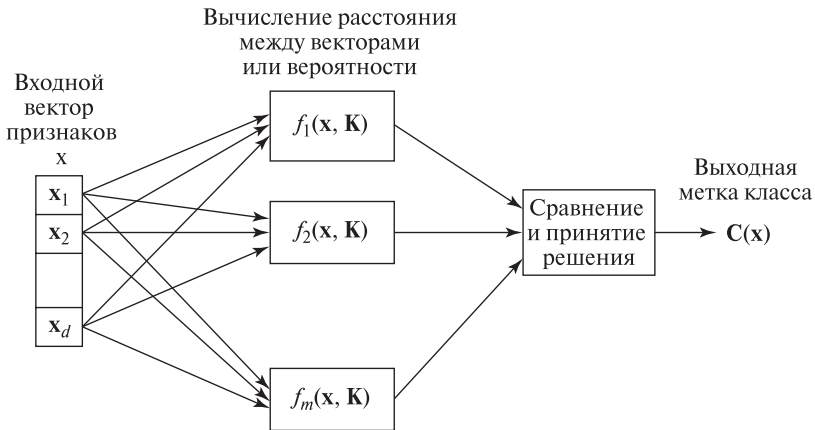
Имеется множество из  $t$  известных классов объектов. Классы представлены или в форме некоторого описания, или в виде множества примеров объектов (эталонных образцов) каждого класса. Например, при классификации символов для каждого символа имеется или описание его внешнего вида, или набор образцов. В общем случае также предусматривается особый класс выбросов для объектов, которые не удалось отнести ни к одному из известных классов.

**24 Определение. Идеальный класс** — это множество объектов, у которых есть некоторые важные общие свойства. На практике, это класс, принадлежность объектов к которому обозначается с помощью какой-либо **метки класса**. **Классификация** — это процесс назначения меток объектам согласно некоторому описанию свойств объектов. **Классификатор** — это устройство или алгоритм, получающий в качестве входных данных описание объекта и выдающий в качестве результата метку класса.

**25 Определение. Класс выбросов** — это общий класс всех объектов, которые не удалось отнести ни к одному из известных системе классов.

### 4.2.2 Датчик/преобразователь

Должно быть доступно некоторое устройство для восприятия реального физического объекта, которое в качестве выходных данных генерирует описание



**Рис. 4.2.** Схема системы классификации: классифицирующие функции  $f(x, K)$  выполняют вычислительные операции над входным вектором признаков  $x$  с использованием некоторых сведений  $K$ , полученных на этапе обучения. Результаты вычислений передаются на окончательный этап обработки, на котором определяется класс объекта.

(обычно цифровое) этого объекта, пригодное для машинной обработки. Обычно датчик выбирается среди готовых устройств, которые производятся для применения в широком круге задач. Например, для классификации фруктов в супермаркете сначала можно было бы попробовать воспользоваться цветной камерой общего назначения. Эта камера генерирует представление объекта в виде изображения, из которого можно извлечь цветовые, геометрические и текстурные признаки. Для распознавания символов по данным о нажатии пера нам потребовался бы массив датчиков давления.

Данная книга посвящена компьютерному зрению, поэтому наибольший интерес представляют датчики, которые генерируют двумерные массивы воспринимаемых данных. Однако предмет распознавания образов имеет общий характер и методы из этой области применимы, например, как для речевого распознавания произносимых вслух телефонных номеров, так и для распознавания номеров, написанных на бумаге.

### 4.2.3 Экстрактор характерных признаков

Экстрактор характерных признаков среди исходных данных, полученных от датчика, обнаруживает информацию, полезную для классификации. Экстрактор признаков обычно реализуется программно. Программное обеспечение относительно легко адаптировать к аппаратному обеспечению датчика. Программная реализация в процессе исследований и разработки упрощает достижение выходных данных, наиболее удобных для классификации. Многие часто используемые признаки изображения были описаны в гл. 3.

#### 4.2.4 Классификатор

Классификатор использует признаки, найденные среди исходных данных датчика, для присвоения объекту метки одного из  $m$  известных классов  $C_1, C_2, \dots, C_{m-1}, C_m = C_r$ , где  $C_r$  обозначает класс выбросов.

На рис. 4.2 показана блок-схема системы классификации. Входными данными, представляющими объект для классификации, служит  $d$ -мерный вектор признаков  $\mathbf{x}$ . В системе для каждого возможного класса имеется по одному блоку, в котором хранятся некоторые сведения  $\mathbf{K}$  об этом классе и некоторые характеристики обработки. Результаты  $m$  вычислительных операций передаются на окончательный этап классификации, на котором принимается решение о классе объекта. Эта схема достаточно общая и может служить моделью обсуждаемых ниже трех различных типов классификации: (а) классификация по ближайшему среднему значению; (б) классификация по максимальной *апостериорной* вероятности и (в) классификация с помощью нейронной сети прямого распространения.

#### 4.2.5 Построение системы классификации

Каждая компонента системы может быть реализована различными способами. Датчики изображения были рассмотрены в гл. 2. В гл. 3 обсуждались методы вычисления различных признаков на бинарных изображениях объектов. Вычисление цветовых и текстурных признаков описывается в гл. 6 и 7. Снова обратимся к задаче распознавания символов. Символы, написанные в окне  $30 \times 20$  пикселей, будут занимать 600 пикселей. При выделении признаков надо обработать эти 600 пикселей, и выдать, например, от 10 до 30 признаков, на основе которых будут приниматься решения классификации. Этот пример будет уточняться далее.

Еще одно распространенное название для экстрактора признаков — *препроцессор* (preprocessor). Между датчиком и классификатором может выполняться некоторая фильтрация или удаление шума, эти операции могут входить в функции препроцессора. Методы удаления шума частично рассматривались в гл. 3, более подробно они будут изучаться в гл. 5. Разделение обработки на выделение признаков и классификацию часто носит произвольный характер и скорее объясняется инженерными решениями, а не какими-либо внутренними свойствами приложения. В конце данной главы будут рассмотрены нейронные сети, которые выполняют классификацию непосредственно по входному изображению за один шаг обработки.

#### 4.2.6 Оценка ошибок системы

*Коэффициент ошибок* (error rate) системы классификации — это одна из численных мер, позволяющих охарактеризовать, насколько хорошо система решает свою задачу. Другими характеристиками являются скорость (количество объектов, обрабатываемых за единичный промежуток времени) и стоимость

(учитывающая цену аппаратного, программного обеспечения и процесса разработки). Производительность системы зависит и от количества ошибок, и от количества выбросов. Классификация всех входных данных в класс выбросов означает, что система не делает ошибок, но она бесполезна.

- 26 Определение. Ошибка классификации** происходит каждый раз, когда классификатор относит входной объект к классу  $C_i$ , хотя верным классом должен быть класс  $C_j$ , причем  $i \neq j$  и  $C_i \neq C_r$  ( $C_r$  — класс выбросов).
- 27 Определение. Эмпирический коэффициент ошибок** системы классификации равен количеству ошибок, сделанных на независимых тестовых данных, деленному на количество предпринятых попыток классификации.
- 28 Определение. Эмпирический коэффициент выбросов** системы классификации равен количеству выбросов, сделанных на независимых тестовых данных, деленному на количество предпринятых попыток классификации.
- 29 Определение. Независимые тестовые данные** — это эталонные объекты известных классов, включая объекты из *класса выбросов*, которые не использовались при разработке алгоритмов выделения признаков и классификации.

Приведенные выше определения можно использовать на практике для тестирования производительности системы классификации. Следует очень внимательно следить за тем, чтобы эталонные образцы, применяемые на этапе проектирования, и образцы для тестирования были представительными образцами объектов, которые система будет обрабатывать в будущем. Кроме того, образцы для тестирования системы должны быть независимы от образцов, используемых для ее проектирования. Иногда предполагается, что исходные данные подчиняются некоторому теоретическому распределению. С учетом этого можно быстрее вычислить теоретическую вероятность ошибки для оценки производительности системы, а не определять эмпирический коэффициент ошибок по результатам тестирования. Этот подход будет обсуждаться ниже.

Предположим, что модуль распознавания рукописных символов для карманного компьютера корректно распознает 95% введенных пользователем символов. Так как пользователь может потом редактировать входной документ, то 5% ошибок является приемлемым коэффициентом ошибок. Интересно, что подобная система может обучить пользователя или, наоборот, быть обучена пользователем так, что производительность улучшится. Например, пользователь запомнил, что надо более аккуратно писать верхнюю часть символа '8', чтобы система не путала его с '6'. Для банковской системы, которая считывает рукописные цифры с депозитного бланка, коэффициент ошибок 5% может быть недопустимым.



### 4.2.7 Ложные срабатывания и ложные пропуски

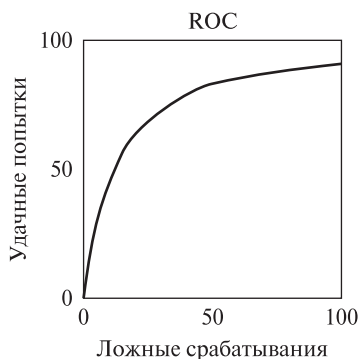
Особой разновидностью ошибок являются *проблемы двух классов* (two-class problems), возникающие, когда два класса имеют противоположные смысловые значения: (а) хороший объект и плохой объект, (б) объект присутствует на изображении и объект отсутствует, или (в) у человека обнаружено заболевание D и не обнаружено. В таких случаях ошибки имеют особое значение и несимметричны.

Случай (в) наиболее иллюстративен. Если система ошибочно говорит, что у человека есть заболевание D, то ошибка называется *ложным срабатыванием* или *позитивной ошибкой*. Если же система ошибочно говорит, что у человека нет заболевания D, то ошибка называется *ложным пропуском* или *негативной ошибкой*. В случае ложного срабатывания вероятно, что человек подвергнется дополнительному анализу или перенесет ненужные медицинские процедуры. В случае ложного пропуска диагноз будет потерян и человек не будет вылечен, что может привести к самым неприятным последствиям. Цена этих ошибок существенно отличается. Поэтому может иметь смысл предусмотреть принятие решений с тенденцией минимизации ложных пропусков даже за счет роста числа ложных срабатываний.

Случай (а) менее драматический, например, это задача об отборе мятых вишен. Ложное срабатывание означает, что вишня отправляется в пирог, а не в розничную продажу.

Ложные срабатывания в примере (б) могут означать лишние траты энергии на включение света, когда фактически движения в сцене нет, а система решила, что по дороге проехал автомобиль — на самом деле его не было. Ложные пропуски в случае (б) также имеют интересные следствия.

На рис. 4.3 показана типичная *кривая функционирования приемника*, которая представляет взаимосвязь числа ложных срабатываний и числа правильных обнаружений. Для увеличения количества корректно распознаваемых объектов обычно приходится идти на увеличение количества пропущенных объектов, которые будут отнесены к выбросам.



**Рис. 4.3.** Кривая функционирования приемника (receiver operating curve, ROC) показывает отношение количества корректных и неправильных попыток распознавания. Обычно количество ложных срабатываний возрастает при попытке системы обнаружить большее количество известных объектов. Уменьшение производительности системы часто может быть достигнуто при малых ложных срабатываниях, но корректное обнаружение почти всех объектов приводит к тому, что значительная часть неизвестных объектов будет неправильно отнесена к каким-либо известным классам.

### 4.3 ВЗАИМОСВЯЗЬ ТОЧНОСТИ СИСТЕМЫ И ПОЛНОТЫ ВЫБОРКИ

В задаче поиска документов (document retrieval, DR) или изображений требуется на основе признаков, содержащихся в пользовательском запросе, найти объекты интереса класса  $C_1$ , возможно, вместе с не слишком большим количеством неинтересующих объектов класса  $C_2$ . Например, пользователь может заказать поиск изображений солнечных закатов или лошадей. Производительность системы, обслуживающей такие запросы, характеризуется *точностью* и *полнотой выборки*.

**30 Определение.** Точность DR-системы равна количеству найденных релевантных документов (истинные элементы класса  $C_1$ ), деленных на общее число найденных документов (истинные элементы класса  $C_1$  плюс ложные срабатывания, в действительности принадлежащие  $C_2$ ).

**31 Определение.** Полнота выборки DR-системы равна количеству релевантных документов, найденных системой, деленному на общее количество релевантных документов в базе данных.

Например, предположим, что в базе данных хранится 200 изображений солнечных закатов, которые интересуют пользователя и которые пользователь хочет получить в результате запроса. Допустим, система находит 150 из 200 релевантных изображений и 100 других изображений, ненужных пользователю. Точность этой операции поиска (классификации) составляет  $150/250 = 60\%$ , а полнота выборки  $150/200 = 75\%$ . Система может достичь 100% полноты выборки, если вернет все изображения базы данных, но при этом ее точность будет чрезвычайно низкой. С другой стороны, если классификация тщательно настроена на низкий коэффициент ложных срабатываний, то точность будет высокой, но полнота выборки может быть низкой. Поиск в базе данных изображений будет более подробно рассматриваться в гл. 8.

### 4.4 ПРИЗНАКИ, ИСПОЛЬЗУЕМЫЕ ДЛЯ ОПИСАНИЯ ОБЪЕКТОВ

И для теории, и для практики важным является вопрос о том, *какое представление или способ кодирования объекта следует использовать в процессе распознавания*. Или, по-другому, какие признаки важны для распознавания? Вернемся к задаче распознавания рукописных символов. Допустим, что отдельные символы можно изолировать друг от друга с помощью алгоритма маркировки связных компонент или предъявляя пользователю требование, что символы следует писать в явно различимых прямоугольных областях. Будем считать, что для каждого символа доступны следующие признаки, вычисленные с помощью методов из гл. 3.

- *Площадь* символа (количество черных пикселей).
- *Высота* и *ширина* описывающего прямоугольника.
- *Количество отверстий* внутри символа.

Таблица 4.1. Пример признаков для простого набора символов.

(Класс) Символ	Значение площади	Значение высоты	Значение ширины	Коли- чество отверстий	Коли- чество штрихов	Центр тяжести (cx,cy)	Главная ось	Мини- мальное значение момента инерции
'A'	среднее	большое	3/4	1	3	1/2, 2/3	90	среднее
'B'	среднее	большое	3/4	2	1	1/3, 1/2	90	большое
'8'	среднее	большое	2/3	2	0	1/2, 1/2	90	среднее
'0'	среднее	большое	2/3	1	0	1/2, 1/2	90	большое
'1'	малое	большое	1/4	0	1	1/2, 1/2	90	малое
'W'	большое	большое	1	0	4	1/2, 2/3	90	большое
'X'	большое	большое	3/4	0	2	1/2, 1/2	?	большое
'*'	среднее	малое	1/2	0	0	1/2, 1/2	?	большое
':'	малое	малое	2/3	0	1	1/2, 1/2	0	малое
'/'	малое	большое	2/3	0	1	1/2, 1/2	60	малое

- *Количество штрихов*, образующих символ.
- *Центр тяжести* множества пикселей символа.
- *Направление главной оси*, проходящей через пиксели символа, относительно которой момент инерции принимает наименьшее значение.
- *Вторые моменты* пикселей относительно главных осей (осей с наибольшим и наименьшим моментами инерции).

На основе интуитивных соображений и анализа эталонных символов можно составить таблицу свойств символов в терминах перечисленных признаков. Таблицу можно уточнить, учитывая свойства большого количества образцов каждого символа. После этого можно сформировать короткую процедуру принятия решения для классификации символов или, как минимум, сформировать набор прототипов, которыми можно будет пользоваться для сравнения.

В табл. 4.1 показаны 8 признаков 10 различных символов. Сейчас будем полагать, что признаки были вычислены без ошибок. Последовательная процедура принятия решений для классификации экземпляров этих 10 классов приведена в виде алгоритма 4.1. Эта структура для выполнения классификации называется *деревом решений*. Решения, показанные в виде таблицы, легко реализовать в форме компьютерной программы, имеющей доступ к значениям признаков. В каждой точке процесса решения используется только малый набор признаков, с помощью которых осуществляется переход к некоторым другим точкам процесса решения. В нашем примере в каждой точке решения используется только один признак. Процесс ветвления моделирует редукцию

**Классификация символа по трем характерным признакам.**

**Входные данные:** вектор признаков вида [количество отверстий #holes, количество штрихов #strokes, минимальный момент инерции относительно главной оси ]

**Выходные данные:** метка класса символов

case of #holes

0: символ 1, W, X, \*, -, или /

case of момент относительно главной оси

малый: символ 1, - или /

case of направление главной оси

0: символ -

60: символ /

90: символ 1

большой: символ W или X или \*

case of #strokes

0: символ \*

2: символ X

4: символ W

1: символ A или O

case of #strokes

0: символ o

3: символ A

2: символ B или 8

case of #strokes

0: символ 8

1: символ B

**Алгоритм 4.1.** Простая процедура принятия решений для классификации на множестве из 8 возможных символов.

на множестве возможных решений, производимую по мере успешного учета все большего количества признаков.

Описанный пример интуитивно понятен, но было бы наивно полагать, что схематично обозначенная здесь процедура принятия решений близка к процедуре, которая сможет корректно работать в реальной системе распознавания рукописных символов. Например, надежное описание и вычисление количества штрихов оказывается очень сложной задачей (хотя в гл. 10 будет приведен работоспособный метод). Более того, требуется применение методов из гл. 3 и 5 для устранения некоторой изменчивости данных перед выделением признаков. В управляемой промышленной среде можно применить описанную простую процедуру классификации и затем настроить численные параметры в соответствии с измерениями, проведенными на эталонных изображениях. По результатам измерений можно ожидать изменчивость признаков внутри одного класса и перекрытие некоторых признаков между классами. Ряд методов для применения в условиях изменчивости и перекрытия признаков будет рассмотрен далее.

## 4.5 ПРЕДСТАВЛЕНИЕ ОБЪЕКТОВ В ВИДЕ ВЕКТОРОВ ПРИЗНАКОВ

Сравнение объектов можно производить на основе их представления в виде векторов измерений. Допустим, что каждый объект представлен в точности  $d$  результатами измерений. У каждого объекта  $A$   $i$ -я координата такого вектора признаков имеет один и тот же смысл. Например, первая координата может быть площадью объекта, вторая — моментом второго порядка по строке  $\mu_{rr}$ , определенным в гл. 3, третья координата — это протяженность объекта и т. п. Данные измерений удобно представлять в виде вещественных чисел. Сходство (близость) векторов признаков двух объектов может быть описано с помощью евклидова расстояния между векторами (4.1). Как показано на рис. 4.4 и пояснено в следующем параграфе, иногда евклидово расстояние между наблюдаемым вектором и хранимым в памяти прототипом класса позволяет сформировать практически полезную классифицирующую функцию.

**32 Определение.** Евклидово расстояние между двумя  $d$ -мерными векторами признаков  $\mathbf{x}_1$  и  $\mathbf{x}_2$  равно

$$\|\mathbf{x}_1 - \mathbf{x}_2\| = \sqrt{\sum_{i=1,d} (\mathbf{x}_1[i] - \mathbf{x}_2[i])^2} \quad (4.1)$$

## 4.6 РЕАЛИЗАЦИЯ КЛАССИФИКАТОРА

Теперь вернемся к классическому подходу к распознаванию образов, в котором неизвестный объект для классификации представляется в виде вектора элементарных признаков. Система распознавания на основе векторов признаков может быть разработана различными способами — эти векторы могут быть известны системе в результате обучения или предсказаны на основе каких-либо моделей. Далее будут рассмотрены два альтернативных метода использования



**Рис. 4.4.** Два компактных класса: классификация по ближайшему среднему значению характеризуется малым коэффициентом ошибок.

базы данных эталонных образцов, сформированных в результате обучения системы. Предположим, что есть  $m$  классов объектов, не считая класса выбросов. Для  $i$ -го класса имеется  $n_i$  эталонных векторов. В нашем примере распознавания символов (алгоритм 4.1) есть  $m = 10$  классов символов; возможно, мы могли бы сформировать  $n_i = 100$  образцов для каждого класса. В этом примере векторы признаков имеют размерность  $d = 8$ .

#### 4.6.1 Классификация по ближайшему среднему значению

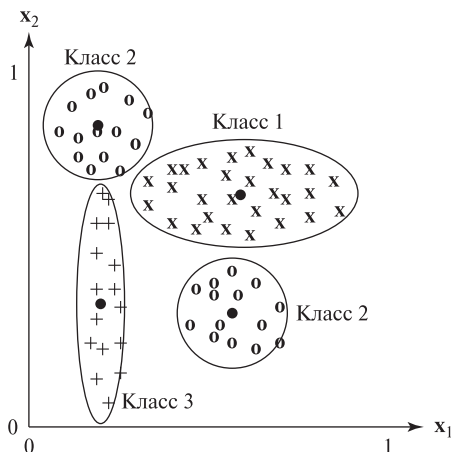
Простой алгоритм классификации заключается в группировке эталонных данных каждого класса с использованием вектора математического ожидания (среднего значения, центра тяжести) класса,  $\bar{\mathbf{x}}_i = 1/n_i \sum_{j=1, n_i} \mathbf{x}_{i,j}$ , где  $\mathbf{x}_{i,j}$  —  $j$ -й эталонный вектор признаков для класса  $i$ . Неизвестный объект с вектором признаков  $\mathbf{x}$  относится к классу  $i$ , если он существенно ближе к вектору математического ожидания класса  $i$ , чем к векторам математических ожиданий других классов. Вектор  $\mathbf{x}$  можно отнести к классу выбросов, если он не лежит достаточно близко к одному из векторов математических ожиданий. Этот метод классификации прост и быстро работает, он подходит для задач, в которых точки каждого класса располагаются компактно и далеко от других классов. Простой пример с двумя классами с двумерными векторами признаков показан на рис. 4.4. Точки одного класса обозначены символами  $\mathbf{x}$ , а точки другого — символами  $\mathbf{o}$ . Так как в каждом классе есть точки, расположенные на одинаковом расстоянии от центров обоих классов, то коэффициент ошибок классификации будет ненулевым. При данной структуре классов коэффициент ошибок будет небольшим, если только структура эталонных образцов хорошо представляет структуру распознаваемых системой объектов. Теперь у нас есть конкретный пример интерпретации функциональных блоков на рис. 4.2:  $i$ -й функциональный блок вычисляет расстояние между неизвестным входным вектором  $\mathbf{x}$  и вектором математического ожидания эталонных образцов со-

ответствующего класса. Эталонные образцы представляют собой априорные сведения  $\mathbf{K}$  о каждом классе.

#### Упражнение 4.1. Классификация монет (А)

Возьмите по 10 экземпляров металлических монет 5–6 разновидностей. Микрометром измерьте диаметр и толщину каждой из 60 монет с точностью порядка 0,25 мм и постройте точечный график шести классов по аналогии с рис. 4.4. (Всегда измеряйте толщину в одном и том же месте, например, вблизи центра или около края монет.) Оцените коэффициент ошибок классификатора на основе близости к центру классов.

Трудности могут возникнуть, если классы эталонных образцов обладают сложной структурой. На рис. 4.5 показан случай, в котором образцы классов хорошо разделены, но их структура такова, что классификация по близости к центру даст плохие результаты — и этому есть несколько причин. Прежде всего, второй класс (о) является *многомодальным*: его образцы лежат в двух отдельных компактных кластерах, которые плохо описываются одним средним значением — эта точка лежит между двумя модами. Несколько образцов первого класса (х) ближе к центру второго класса, а не первого. Изучая образцы, мы можем обнаружить наличие двух мод во втором классе и разделить второй класс на два подкласса, описываемых различными средними значениями. Это легко сделать с помощью двумерного точечного графика, как на рис. 4.5, но может быть очень сложно понять структуру образцов при количестве измерений  $d$  намного больше 2. Вторая проблема возникает из-за протяженности первого и третьего классов. Очевидно, образцы третьего класса с большими координатами  $x_2$  ближе к среднему значению второго класса, а не первого. Аналогично, образцы первого класса (х) с малыми координатами  $x_1$  будут ближе к среднему значению одного из подклассов второго класса, даже если разбить его по двум модам. Описанная проблема в некоторых случаях



**Рис. 4.5.** Три класса со сложной структурой: классификация по ближайшему среднему значению в данном случае работает плохо.

может быть устранена путем изменения способа расчета расстояния для учета различного распределения образцов по разным измерениям.

«Модифицированное» расстояние между неизвестным вектором признаков  $\mathbf{x}$  и вектором математического ожидания класса  $\mathbf{x}_c$  можно вычислить путем масштабирования компонентов евклидова расстояния на *среднеквадратическое отклонение*  $\sigma_i$  (характеристику «разброса» значений класса) класса  $c$  вдоль каждого координатного направления  $i$ . Среднеквадратическое отклонение равно квадратному корню из дисперсии.

**33 Определение.** Шкалированное евклидово расстояние между вектором  $\mathbf{x}$  и вектором математического ожидания класса  $\mathbf{x}_c$  равно:

$$\|\mathbf{x} - \mathbf{x}_c\| = \sqrt{\sum_{i=1,d} ((\mathbf{x}[i] - \mathbf{x}_c[i])/\sigma_i)^2} \quad (4.2)$$

Шкалирование почти всегда требуется при использовании различных единиц измерения по разным координатным осям. Например, рассмотрим классификацию автомобилей по признакам  $\mathbf{x}[1]$  = длина в метрах и  $\mathbf{x}[2]$  = вес в килограммах. Без шкалирования второй признак значительно снизит влияние различия по длине автомобилей.

В примере на рис. 4.5, когда есть два отдельных вектора математических ожиданий для двух мод второго класса, такое раздельное, зависящее от класса, шкалирование признаков  $x_1$  и  $x_2$  даст хорошие результаты классификации. Однако большинство задач не настолько простые. Если эллипсы, представляющие распределение образцов классов, не выровнены по координатным осям, как на рис. 4.5, то необходимо преобразование координат для корректного вычисления расстояния от неизвестного вектора до вектора математического ожидания класса. Этот вопрос обсуждается далее в связи со схемой байесовской классификации. Более сложная проблема возникает, когда множество образцов имеет криволинейную границу в  $d$ -мерном пространстве.

#### 4.6.2 Классификация по расстоянию до ближайших соседей

Более гибкий, но одновременно более длительный метод классификации заключается в отнесении неизвестного вектора признаков  $\mathbf{x}$  к тому классу, к отдельному эталонному образцу которого этот вектор наиболее близок. Это *правило ближайшего соседа*. Классификация по ближайшему соседу может быть эффективна, даже если классы имеют сложную структуру в  $d$ -мерном пространстве и когда классы перекрываются. При таком подходе не требуется предположений о моделях распределения векторов признаков в пространстве; алгоритм пользуется только известными эталонными образцами. Метод лобового решения (алгоритм 4.2) основан на вычислении расстояния от  $\mathbf{x}$  до всех образцов в базе данных и нахождении минимального расстояния. Преимущество такого подхода в том, что новые маркированные образцы можно в любой момент добавлять в базу данных. Существуют структуры данных, позволяющие сократить количество вычисляемых расстояний. Примерами таких



структур являются древовидные и сеточные структуры данных, они описаны в работах, перечисленных в списке литературы в конце главы.

Наилучшее классификационное решение может быть сделано на основе проверки в базе данных  $k$  ближайших векторов признаков. Значения  $k > 1$  обеспечивают наилучшую выборку распределения векторов в  $d$ -мерном пространстве: это особенно полезно в областях перекрытия классов. Было показано, что в пределе при увеличении количества образцов до бесконечности коэффициент ошибок даже для  $k = 1$  превышает оптимальный коэффициент ошибок не более чем в 2 раза. Теоретически, можно было бы добиться лучших результатов при  $k > 1$ ; однако эффективное использование больших значений  $k$  зависит от того, имеется ли соответствующее количество образцов в каждой рассматриваемой области пространства, чтобы предотвратить поиск среди слишком далеких от  $\mathbf{x}$  образцов. В проблеме двух классов с использованием  $k = 3$  мы могли бы классифицировать  $\mathbf{x}$  как элемент класса, имеющего 2 или 3 образца вблизи  $\mathbf{x}$ . Если имеется больше двух классов, то возникает больше возможных комбинаций и принять решение оказывается сложнее. Приведенный ниже алгоритм 4.2 классифицирует входной вектор в класс выбросов, если среди ближайших  $k$  образцов нельзя выделить большинство образцов одного класса. Этот алгоритм не предполагает какой-либо структуры на множестве эталонных образцов; без такой структуры алгоритм замедляется при увеличении количества образцов  $n$  и  $k$ . Среди работ в списке литературы в конце главы можно найти алгоритмы с применением эффективных структур данных для хранения эталонных образцов.

## 4.7 СТРУКТУРНЫЕ МЕТОДЫ РАСПОЗНАВАНИЯ

Для распознавания может оказаться недостаточно простых численных или символических признаков объекта. Например, рассмотрим два символа на рис. 4.6. У них одинаковые описывающие прямоугольники, одинаковое количество отверстий и штрихов, одинаковые центры тяжести, равные вторые моменты по направлениям строк и столбцов, и главные оси символов совпадают с точностью до 0.1 радиан. У каждого из этих символов по два *залива*, которые выглядят как проникновение фона внутрь символа. У каждого залива есть *крышка*, воображаемый отрезок, замыкающий залив. Главный отличительный признак двух этих символов заключается во взаимосвязи, пространственном отношении между двумя заливами. У символа слева крышка верхнего залива расположена справа от крышки нижнего залива. У символа справа крышка верхнего залива расположена слева от крышки нижнего залива. Эти сведения о взаимосвязи между примитивными признаками можно использовать на более высоком уровне обработки и, в принципе, таким образом формируются более мощные признаки для распознавания. На основе подобных соображений были разработаны методы *структурного распознавания образов*.

При статистическом распознавании символов сущности представляются с помощью векторов признаков, которые обычно являются векторами элементарных значений, таких, как числа или логические значения (ИСТИНА или

**Вычисление  $k$  ближайших соседей для вектора  $x$  и возврат метки класса, которому принадлежит большинство ближайших соседей.**

**Входные данные:**

$S$  — множество  $n$  образцов  $s_i$  с метками классов, где  $s_i.x$  есть вектор признаков, а  $s_i.c$  является целочисленной меткой класса.

$x$  — неизвестный входной вектор признаков для классификации.

$A$  — массив для хранения  $k$  образцов, отсортированных по расстоянию  $d$ .

**Выходные данные:**

метка классов в виде целого числа в диапазоне  $[1, m]$

```

procedure K_Nearest_Neighbors( $x, S$ )
{
  очистка массива  $A$ ;
  for каждый образец  $s_i$  из  $S$ 
  {
     $d$  = евклидово расстояние между  $s_i$  и  $x$ ;
    if  $A$  содержит меньше  $k$  элементов then добавить  $(d, s_i)$  в  $A$ ;
    else if  $d$  меньше максимального значения из  $A$ 
      then {
        удалить максимальный элемент из  $A$ ;
        добавить  $(d, s_i)$  в  $A$ ;
      }
  } ;
  на этом этапе считается, что  $A$  содержит  $k$  образцов из  $S$ ,
  ближайших к  $x$ ;
  if большинство меток  $s_i.c$  из  $A$  принадлежат классу  $c_0$ 
    then отнести  $x$  к классу  $c_0$ ;
    else отнести  $x$  к классу выбросов;
  return(метка класса  $x$ );
}

```

**Алгоритм 4.2.** Классификация по расстоянию до ближайших соседей.

ЛОЖЬ). Эти значения являются мерой некоторых глобальных особенностей сущностей, таких, как площадь или моменты инерции. Наш пример с распознаванием символов более сложен за счет того, что в нем измеряется количество отверстий и штрихов в каждом символе. Это подразумевает наличие процедуры поиска отверстий и некоторой разновидности алгоритма сегментации для разбиения символа на штрихи.

0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	1	1	1	0
0	1	0	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	1	1	0
0	1	0	0	0	1	1	1	0	0	0
0	0	1	1	1	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	1	1	1	1	1	1	1	1	1	0
0	0	1	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0
0	0	1	1	1	1	0	0	0	1	0
0	0	0	0	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0	0

Рис. 4.6. Два символа с одинаковыми глобальными признаками, но с различной структурой.

В структурном распознавании символов сущность представляется в виде совокупности элементарных частей, их атрибутов и отношений наряду с глобальными признаками сущности. На рис. 4.7 показаны три отдельные буквы 'А', имеющие примерно одинаковую структуру. Каждая буква может быть разделена на 4 основных штриха: два горизонтальных и два вертикальных или скошенных. У каждого символа есть отверстие (*озеро*) в верхней части, а также залив внизу; озеро и залив разделены горизонтальным штрихом.

В случае бинарных отношений между примитивами структурное описание сущности можно представить в виде графа. Допустим, что между штрихами, заливами и озерами возможны следующие отношения, полезные при распознавании символов:

- **CON**: указывает на соединение двух штрихов;
- **ADJ**: указывает, что область штриха соприкасается с областью озера или залива;
- **ABOVE**: указывает, что одно отверстие (озеро или залив) лежит выше другого.

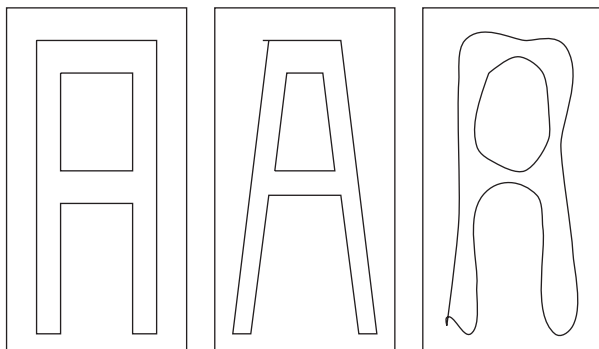
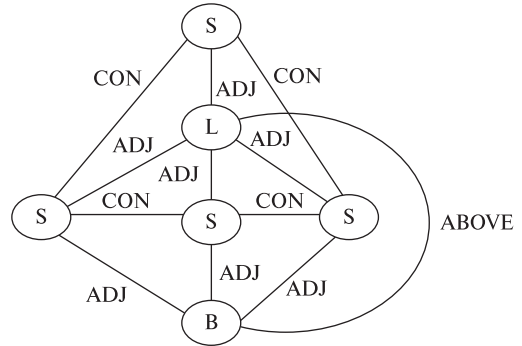


Рис. 4.7. Три символа 'А' с похожей структурой.

На рис. 4.8 показано представление структурного описания символа 'A' в виде графа, в котором используются перечисленные выше три бинарных отношения. Отношения более высокого уровня, например, тернарные или даже кватернарные, целесообразно определять и использовать, если они позволяют обеспечить какие-то более сильные ограничения. Например, можно построить тернарное отношение между озером, нижележащим горизонтальным штрихом и заливом, лежащим ниже штриха.



**Рис. 4.8.** Граф, представляющий символ 'A'. Обозначения 'S', 'L' и 'B' соответствуют *штриху* (stroke), *озеру* (lake) и *заливу* (bay).

Структурное распознавание образов часто реализуется с помощью алгоритмов *сопоставления графов*, которые будут рассматриваться в гл. 11. Однако отношение между двумя примитивами само может рассматриваться как элементарный признак, входить в вектор признаков и учитываться в процедуре статистического принятия решения. Один из простейших подобных способов состоит в том, что для данного образа можно подсчитать количество заданных отношений между двумя признаками разных типов (например, залив внизу горизонтального штриха). Затем целочисленное значение счетчика используется в качестве признака для распознавания образа.

Структурные методы полезны для распознавания сложных образов, состоящих из многих образов более низкого уровня. Они также имеют преимущества при высокоуровневом понимании сцены, особенно при наличии нескольких объектов. С общей точки зрения, структурное распознавание образов вместе с другими методами данной главы охватывает почти все компьютерное зрение: при таком подходе оставшиеся главы книги описывают различные методы для выделения признаков и частей образов по двумерным и трехмерным изображениям объектов и сцен.

## 4.8 МАТРИЦА НЕТОЧНОСТЕЙ

**34 Определение.** Матрица неточностей обычно используется для представления результатов экспериментов по классификации. Пример приведен на рис. 4.9. Элемент в  $i$ -й строке и  $j$ -м столбце равен количеству попыток

		Класс $j$ , идентифицированный системой распознавания											
		'0'	'1'	'2'	'3'	'4'	'5'	'6'	'7'	'8'	'9'	'R'	
Истинный класс объекта	$i$	'0'	97	0	0	0	0	0	1	0	0	1	1
	'1'	0	98	0	0	1	0	0	1	0	0	0	
	'2'	0	0	96	1	0	1	0	1	0	0	1	
	'3'	0	0	2	95	0	1	0	0	1	0	1	
	'4'	0	0	0	0	98	0	0	0	0	2	0	
	'5'	0	0	0	1	0	97	0	0	0	0	2	
	'6'	1	0	0	0	0	1	98	0	0	0	0	
	'7'	0	0	1	0	0	0	0	98	0	0	1	
	'8'	0	0	0	1	0	0	1	0	96	1	1	
	'9'	1	0	0	0	3	0	0	0	1	95	0	

**Рис. 4.9.** Гипотетическая матрица неточностей при распознавании цифры. Символ 'R' обозначает класс выбросов.

классификации, в которых объект, принадлежащий классу  $i$ , был ошибочно классифицирован как объект класса  $j$ .

Диагональные элементы матрицы неточностей с индексами  $i = j$  обозначают успешные результаты. При идеальных результатах классификации все недиагональные элементы равны нулю. Большие значения недиагональных элементов указывают на неверный выбор классов и требуют дополнительного анализа процедур выделения признаков и/или процедуры классификации. В процессе тестирования системы матрица неточностей показывает типы и коэффициенты ошибок, которые следует ожидать при последующей эксплуатации системы. В примере, показанном на рис. 4.9, семь из 1000 входных векторов были отнесены системой к классу выбросов. Три входных вектора класса 9 были неправильно помечены как 4, а два вектора класса 4 — некорректно классифицированы как 9. В целом, 25 входных векторов были классифицированы ошибочно. Полагая, что тестовые данные независимы от данных для обучения системы, можно вычислить эмпирический коэффициент выбросов  $7/1000 = 0.007$  и (общий) коэффициент ошибок  $25/1000 = 0.025$ . Однако коэффициент ошибок для элементов класса 9 равен  $5/100 = 0.05$ .

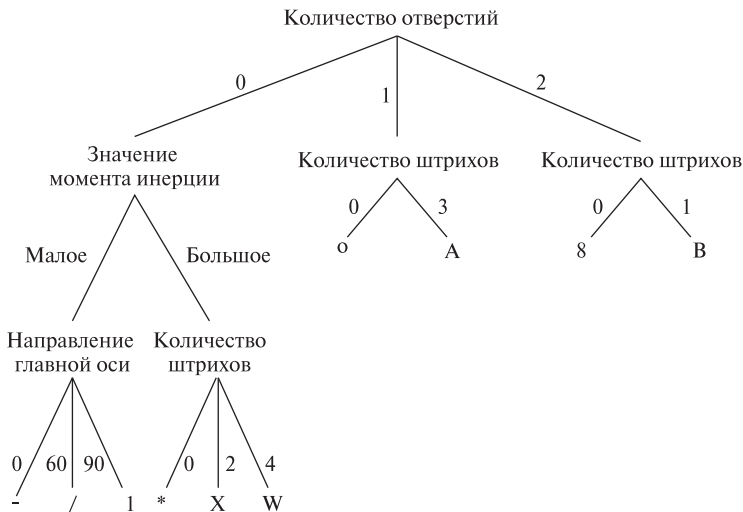
## 4.9 ДЕРЕВЬЯ РЕШЕНИЙ

В сложных задачах распознавания образов, в которых требуется учитывать большое количество различных возможных признаков, полное сравнение неизвестного вектора признаков со многими векторами признаков эталонных образцов может занимать слишком много времени. Иногда это недопустимо, например, в случае медицинской диагностики, когда измерение признаков часто требует дорогих и, возможно, болезненных лабораторных анализов. Использование дерева решений позволяет чередовать этапы выделения признаков и классификации. Дерево решений — это компактная структура, ис-

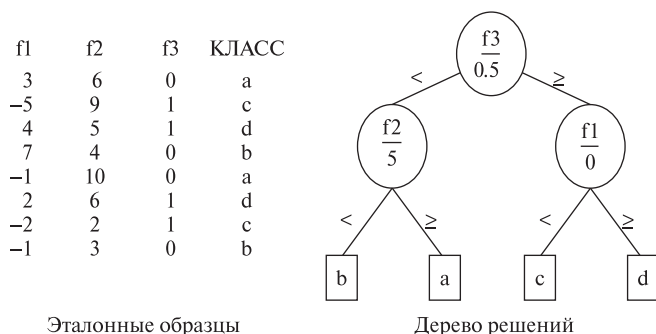
пользующая один признак (или несколько) в каждый момент времени для разбиения пространства поиска всех возможных образов. Простая процедура принятия решения в алгоритме 4.1 представляет собой поток управления для реализации дерева решения согласно рис. 4.10. Узлы этого дерева представляют различные признаки вектора признаков. Каждый узел с ветвями имеет по одному дочернему узлу на каждое возможное значение своего признака. Процедура принятия решения выбирает дочерний узел на основе значения заданного признака в неизвестном векторе признаков. Дочерний узел может указывать на проверку другого признака или может оказаться листовым узлом, содержащим результат классификации, соответствующий пройденному пути на дереве решений.

**35 Определение. Бинарное дерево решений** — это структура данных в виде бинарного дерева, с каждым узлом которого связана функция выбора решения. Функция выбора решения применяется к неизвестному вектору признаков и определяет, какой дочерний узел текущего узла следует обрабатывать далее — левый или правый.

В простейшем случае, когда значения признаков являются числами, функция выбора решения в узле просто сравнивает значение конкретного признака неизвестного вектора признаков с порогом и выбирает левый дочерний узел, если это значение признака меньше порога, или правый дочерний узел в противном случае. При этом в каждом ветвящемся узле дерева надо хранить только тип учитываемого признака и пороговое значение. В каждом листовом узле хранится метка класса образов. Если процедура обхода дерева решения достигает листового узла, то неизвестный вектор признаков классифициру-



**Рис. 4.10.** Дерево решений, представляющее процедуру классификации из алгоритма 4.1.



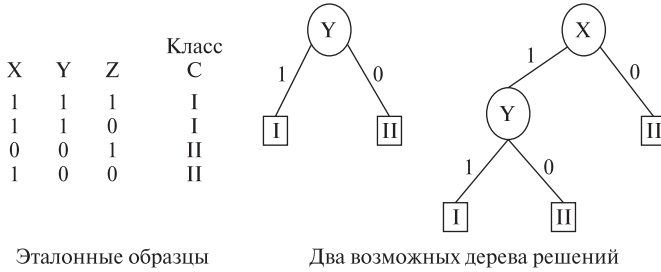
**Рис. 4.11.** Бинарное дерево решений, основанное на хранении в каждом внутреннем узле типа и порогового значения численного признака.

ется как элемент соответствующего класса образов. На рис. 4.11 поясняется структура дерева решения, которое было сформировано для корректной классификации приведенных эталонных данных.

Дерево на рис. 4.11 было построено вручную, после анализа данных для выбора подходящих признаков и пороговых значений. Показанные эталонные данные являются просто модельным примером; реальные данные обычно содержат гораздо больше признаков и намного больше образцов. В практических приложениях, например, в медицинской диагностике, нередко имеется несколько сотен признаков и тысячи эталонных образцов. В таких случаях необходима автоматическая процедура для формирования дерева решений. Кроме того, для любого данного множества эталонных образцов может существовать более одного классифицирующего дерева решений. Поэтому важно выбрать признаки, которые дают наилучшее, согласно некоторому критерию, дерево решений. Обычно предпочтительнее дерево, которое является наиболее простым или с наименьшим количеством уровней, т. е. с меньшим количеством выполняемых проверок.

Рассмотрим эталонные данные и два возможных дерева решений на рис. 4.12. Оба дерева позволяют различать два класса — класс I и класс II, показанные в эталонных данных. Дерево слева имеет очень простую структуру, оно позволяет классифицировать вектор признаков по результатам одного сравнения. Дерево справа имеет больший размер и требует большего количества сравнений.

**Автоматическое построение дерева решений.** Существует несколько различных методов построения оптимального дерева решения по эталонным данным, в каждом из которых принято собственное определение оптимальности (подробнее см. [Haralick, Shapiro, 1992], Vol. I, Chapter 4). Один простой, но эффективный способ основан на методах теории информации. Фундаментальным понятием теории информации является *энтропия*.



**Рис. 4.12.** Два различных дерева решений для классификации данных эталонных образцов.

**36 Определение.** **Энтропия множества событий**  $x = \{x_1, x_2, \dots, x_n\}$ , где каждое  $x_i$  представляет собой событие, равна

$$H(x) = - \sum_{i=1}^n p_i \log_2 p_i \tag{4.3}$$

где  $p_i$  — вероятность события  $x_i$ .

Энтропию можно интерпретировать как «среднюю неопределенность источника информации». В работе [Quinlan, 1986] использовалась мера на основе энтропии, названная «приростом информации», для вычисления признаков и формирования оптимальных деревьев решений.

**Примеры вычисления энтропии для множеств возможных событий**

Рассмотрим множество из трех возможных событий и соответствующие вероятности:

$$X = \{(x_1, 3/4), (x_2, 1/8), (x_3, 1/8)\}$$

Энтропия вычисляется следующим образом:

$$\begin{aligned} H(x) &= -[(3/4) \log_2(3/4) + (1/8) \log_2(1/8) + (1/8) \log_2(1/8)] \\ &= -[(3/4)(-0.415) + (1/8)(-3) + (1/8)(-3)] \\ &= 1.06 \end{aligned}$$

Аналогично, энтропия множества четырех равновероятных событий равна 2.0.

$$\begin{aligned} X &= \{(x_1, 1/4), (x_2, 1/4), (x_3, 1/4), (x_4, 1/4)\} \\ H(x) &= -[4((1/4)(-2))] = 2 \end{aligned}$$



### Упражнение 4.2

- (а) Вычислите энтропию множества двух равновероятных событий.  
 (б) Вычислите энтропию множества четырех событий с вероятностями  $\{1/8, 3/4, 1/16, 1/16\}$ .

Теория информации позволяет измерить количество информации для события. В частности, для нашей задачи полезно знать количество информации в событии класса («принадлежность данному классу») по отношению к каждому из событий признаков («наличие данного признака»). Количество информации  $I(C; F)$  переменной класса  $C$  с возможными значениями  $\{c_1, c_2, \dots, c_m\}$  по отношению к переменной признака  $F$  с возможными значениями  $\{f_1, f_2, \dots, f_d\}$  определяется так:

$$I(C; F) = \sum_{i=1}^m \sum_{j=1}^d P(C = c_i, F = f_j) \log_2 \frac{P(C = c_i, F = f_j)}{P(C = c_i)P(F = f_j)} \quad (4.4)$$

где  $P(C = c_i)$  — это вероятность появления класса  $C$ , имеющего значение  $c_i$ ;  $P(F = f_j)$  — вероятность признака  $F$ , имеющего значение  $f_j$ , а  $P(C = c_i, F = f_j)$  — совместная вероятность класса  $C = c_i$  и переменной  $F = f_j$ . Эти априорные вероятности можно оценить по частоте соответствующих событий в эталонных данных. Например, так как класс  $I$  наблюдается в двух из четырех эталонных образцов (см. рис. 4.12), то  $P(C = I) = 2/4 = 0.5$ . Так как у трех из четырех эталонных образцов признак  $X$  равен 1, то  $P(X = 1) = 3/4 = 0.75$ .

С помощью меры количества информации можно выбрать наилучший признак для выполнения проверки в корневом узле дерева. Вычислим  $I(C, F)$  для каждого из трех признаков:  $X$ ,  $Y$  и  $Z$ .

$$\begin{aligned} I(C, X) &= P(C = I, X = 1) \log_2 \frac{P(C = I, X = 1)}{P(C = I)P(X = 1)} \\ &+ P(C = I, X = 0) \log_2 \frac{P(C = I, X = 0)}{P(C = I)P(X = 0)} \\ &+ P(C = II, X = 1) \log_2 \frac{P(C = II, X = 1)}{P(C = II)P(X = 1)} \\ &+ P(C = II, X = 0) \log_2 \frac{P(C = II, X = 0)}{P(C = II)P(X = 0)} \\ &= 0.5 \log_2 \frac{.5}{0.5 \times 0.75} + 0 + 0.25 \log_2 \frac{0.25}{0.5 \times 0.25} + 0.25 \log_2 \frac{0.25}{0.5 \times 0.75} \\ &= 0.311 \end{aligned}$$

$$\begin{aligned} I(C, Y) &= 0.5 \log_2 \frac{0.5}{0.5 \times 0.5} + 0 + 0.5 \log_2 \frac{0.5}{0.5 \times 0.5} + 0 \\ &= 1.0 \end{aligned}$$

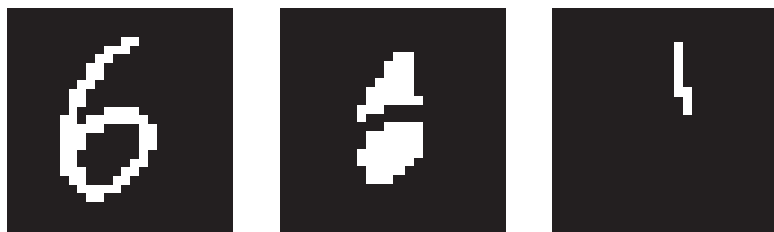
$$\begin{aligned}
 I(C, Z) &= 0.25 \log_2 \frac{.25}{0.5 \times 0.5} + 0.25 \log_2 \frac{0.25}{0.5 \times 0.5} + 0.25 \log_2 \frac{0.25}{0.5 \times 0.5} \\
 &\quad + 0.25 \log_2 \frac{0.25}{0.5 \times 0.5} \\
 &= 0.0
 \end{aligned}$$

Признак  $Y$ , количество информации которого равно 1.0, дает наибольшую информацию для определения класса и поэтому его надо выбрать в качестве первого проверяемого признака для корневого узла дерева решений. В данном примере два класса полностью различимы и дерево содержит только один разветвляющийся узел. В общем случае, в любом разветвляющемся узле дерева, в котором выбранный признак не полностью разделяет множество эталонных образцов на правильные классы, выполняется разбиение множества образцов в соответствии с процедурой выбора решений в данном узле. Алгоритм формирования дерева решений вызывается рекурсивно для подмножеств эталонных образцов в каждом дочернем узле, которому пока соответствует более одного класса.

Описанный здесь алгоритм был предложен для работы с деревьями решений, похожими на дерево с рис. 4.10. Это деревья общего вида с ветвями для каждого возможного значения признака, проверяемого в узле. Для адаптации алгоритма к бинарным деревьям, похожим на рис. 4.11, количество информации каждой пары «признак — пороговое значение» могло бы рассматриваться для каждого возможного порогового значения. Хотя можно предположить наличие бесконечного множества пороговых значений, но в эталонных данных встречается только конечное количество значений каждого признака, и достаточно ограничиться рассмотрением только этого конечного множества.

Вышеприведенный пример очень прост, но автоматически удается построить практически полезные деревья решений для десятков и даже сотен признаков. Снова рассмотрим задачу распознавания символов, но на этот раз более сложную, для случая рукописных символов. Некоторые полезные признаки рукописных символов — это *озера*, *заливы* и *крышки*, рассмотренные в разд. 4.6. Озера — это отверстия в символе (области значений 0, полностью окруженные пикселями символа со значениями 1), заливы — это проникновение фона внутрь символа (области значений 0, только частично окруженные пикселями символа со значениями 1), а крышки — это отрезки, которые могут использоваться для замыкания заливов. На рис. 4.13 показан рукописный символ 'б' (*a*), его признаки — залив и озеро (*b*), а также признак «крышка» (*c*). Для выделения этих элементарных признаков применялись морфологические операции, описанные в гл. 3. По этим признакам можно вычислить следующие численные признаки:

- **lake\_num**: количество озер;
- **bay\_num**: количество заливов;
- **lid\_num**: количество крышек;



(a) Исходное изображение

(b) Залив и озеро

(c) Крышка

**Рис. 4.13.** (a) Изображение рукописного символа '6'; (b) залив (сверху) и озеро (внизу), выделенные в результате морфологической обработки изображения; (c) крышка залива, выделенная после дополнительной морфологической обработки.

- **bay\_above\_bay:** логический признак, имеющий истинное значение, когда какой-нибудь залив полностью расположен выше другого;
- **lid\_rightof\_bay:** логический признак, имеющий истинное значение, когда какая-нибудь крышка полностью расположена справа от залива;
- **bay\_above\_lake:** логический признак, имеющий истинное значение, когда какой-нибудь залив полностью расположен выше озера;
- **lid\_bottom\_of\_image:** логический признак, имеющий истинное значение, когда нижняя точка какой-нибудь крышки находится на расстоянии нескольких пикселей вблизи нижней точки всего символа.

При наличии достаточного количества эталонных данных перечисленные признаки можно использовать для формирования дерева решений для классификации рукописных цифр. На рис. 4.14 показан пример множества эталонных данных для цифр от 0 до 9.

---

### Упражнение 4.3. Построение дерева решения

Даны эталонные данные, показанные на рис. 4.14. Напишите программу, в которой количество информации используется для построения дерева решения для различения между десятью классами цифр. Насколько хорошо дерево, построенное по всем 40 образцам, работает с эталонными данными? Что произойдет, если вы построите дерево по последним 20 образцам и затем протестируете его на первых 20 образцах?

---

### Упражнение 4.4

(a) Предложите способ обнаружения озера с помощью морфологических операций обработки изображений из гл. 3. (б) Предложите способ поиска крышки для уже обнаруженного залива.

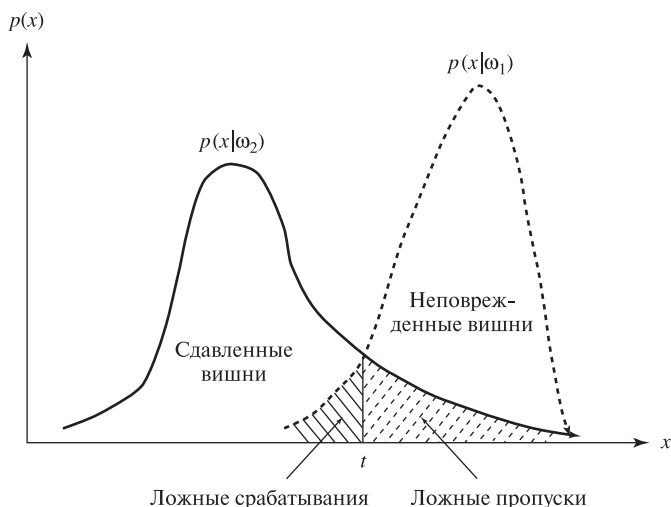
lake_num	bay_num	lid_num	bay_above_bay	lid_rightof_bay	bay_above_lake	lid_bottomof_image	class
1	0	0	F	F	F	F	0
1	0	0	F	F	F	F	0
0	0	0	F	F	F	F	1
0	2	2	F	T	F	T	2
0	2	2	T	F	F	F	3
1	1	1	F	F	F	T	4
1	1	1	F	T	T	F	6
0	2	2	T	T	F	F	2
0	2	2	T	T	F	T	4
0	1	1	F	F	F	F	7
0	0	0	F	F	F	F	1
1	0	0	F	F	F	F	0
1	1	1	F	F	F	F	9
0	2	2	T	T	F	F	2
1	1	1	F	F	F	F	9
0	1	1	F	F	F	F	1
0	2	2	T	F	F	T	4
0	2	2	T	T	F	F	5
1	1	1	F	T	T	F	6
0	2	2	T	F	F	F	3
0	1	1	F	F	F	F	1
1	0	0	F	F	F	F	0
0	2	2	T	T	F	F	5
1	1	1	F	T	T	F	6
0	1	1	F	F	F	T	7
2	0	0	F	F	F	F	8
1	1	1	F	F	F	F	9
1	0	0	F	F	F	F	0
2	0	0	F	F	F	F	8
1	1	1	F	T	T	F	6
0	2	2	F	T	F	F	7
1	1	1	F	F	F	F	9
1	0	0	F	F	F	F	0
0	2	2	T	F	F	F	3
0	2	2	T	T	F	F	5
0	2	2	T	T	F	F	2
0	2	2	T	T	F	F	2
0	1	1	F	F	F	F	1
0	1	1	F	F	F	F	7
0	2	2	T	T	F	F	5
0	2	2	T	F	F	T	4
0	2	2	T	F	F	F	3

Рис. 4.14. Эталонные данные для задачи распознавания рукописных символов.

#### 4.10 БАЙЕСОВСКИЙ ПОДХОД К ПРИНЯТИЮ РЕШЕНИЙ

Рассмотрим, как можно использовать априорную информацию о распределениях вероятности для принятия классификационных решений с наименьшим ожидаемым коэффициентом ошибок.

Предположим, что по инфракрасному изображению темно-красной вишни было произведено единственное измерение  $x$ , по которому требуется определить, является вишня поврежденной или нет. Неповрежденная вишня относится к классу  $\omega_1$ , а поврежденная (сдавленная) — к классу  $\omega_2$ . Также предположим, что мы проанализировали большое количество участков поверхности на многих поврежденных и хороших вишнях, так что у нас есть функции распределения, показанные на рис. 4.15. Кривая справа,  $p(x|\omega_1)$ , описывает распределение измеряемой величины  $x$  среди большого количества поверхностных участков неповрежденных вишен. Кривая слева,  $p(x|\omega_2)$ , показывает распределение измеряемой величины  $x$  среди большого количества поверхностных участков дефектных вишен. Данные были нормированы так, чтобы площадь под каждой кривой была равна 1.0, поэтому каждую кривую можно считать плотностью вероятности. (Сдавленные ткани вишни содержат воду, поглощающую инфракрасное излучение сильнее неповрежденных тканей, поэтому наличие участков с малым отражением более вероятно у дефектных вишен. Содержание воды варьируется, и так же изменчив оттенок вишни, поэтому кривые распределения перекрываются — некоторые темные неповрежденные вишни отражают ИК-излучение так же, как и некоторые светлые сдавленные вишни.)



**Рис. 4.15.** Распределение измеряемой величины яркости  $x$  для неповрежденных и сдавленных вишен.

Если появление дефектных и нормальных вишен равновероятно и если цена всех классификационных ошибок одинакова, то можно принять решение  $\omega_1$  при  $x > t$  и  $\omega_2$  в противном случае. При такой стратегии принятия решений заштрихованная область справа от  $t$  представляет (удвоенный) коэффициент ложных пропусков: это вероятность измерения большого значения  $x$  для сдавленной вишни. Площадь равна удвоенному коэффициенту ложных пропусков, так как *априорная* вероятность каждого класса равна 0.5, и поэтому каждая плотность вероятности должна быть масштабирована так, чтобы общая площадь под одной кривой была равна 0.5. Заштрихованная область слева от  $t$  представляет (удвоенную) вероятность ложных срабатываний: это вероятность того, что хорошая вишня будет отнесена к дефектным при  $x < t$ . Так как дефектные и хорошие вишни считаются примерно равновероятными на входе системы, то каждая кривая в действительности представляет только 0.5 от общей вероятности. Поэтому площадь отмеченных областей в два раза больше их действительного размера. Суммарная ошибка равна сумме заштрихованных площадей под обеими кривыми. Важно отметить, что перемещение порога решения  $t$  влево или вправо приводит к *увеличению* заштрихованной области, и, следовательно, к росту ошибки.

Рассмотренный выше пример демонстрирует особый случай двух равновероятных классов и одинаковой цены ошибок. Теперь расширим метод на случай  $m$  классов, возможно, имеющих различные *априорные* вероятности. Сохраним упрощающее допущение о том, что все ошибки равноценны. С использованием байесовской процедуры принятия решений объект следует отнести к тому классу, к которому он наиболее вероятно принадлежит.

**37 Определение. Байесовский классификатор** на основе наблюдаемых признаков относит объект к классу, к которому этот объект принадлежит с наибольшей вероятностью.

Для вычисления вероятностей по данному измеренному значению  $x$  необходимо знать следующие распределения.

**Распределение**

**условной вероятности:**  $p(x | \omega_i)$  для каждого класса  $\omega_i$  (4.5)

**Априорная вероятность:**  $P(\omega_i)$  для каждого класса  $\omega_i$  (4.6)

**Распределение**

**безусловной вероятности:**  $p(x)$  (4.7)

Если все классы  $\omega_i$  характеризуются собственными несовместными вероятностями, охватывающими все возможные случаи, то можно применить правило Байеса для вычисления *апостериорных* вероятностей каждого класса по *априорным* вероятностям этих классов и распределениям условной вероятности для  $x$ .

$$P(\omega_i | x) = \frac{p(x | \omega_i)P(\omega_i)}{p(x)} = \frac{p(x | \omega_i)P(\omega_i)}{\sum_{j=1,m} p(x | \omega_j)P(\omega_j)} \quad (4.8)$$

Возвращаясь к классификатору, изображенному на рис. 4.2, внутри каждого функционального блока производим вычисление  $f_i(x, K) = P(\omega_i | x)$ , что по правилу Байеса (4.8) можно вычислить как  $p(x | \omega_i)P(\omega_i)/p(x)$ . Так как  $p(x)$  одинакова для всех функциональных блоков, то ее можно игнорировать и принимать классификационное решение  $\omega_i$  по максимальному значению  $p(x | \omega_i)P(\omega_i)$ . Для проектирования байесовского классификатора необходимы сведения  $K$  об априорной вероятности каждого класса  $P(\omega_i)$  и распределение условной вероятности класса  $p(x | \omega_i)$ . Знание этих характеристик позволяет принимать оптимальные решения в процессе распознавания. Часто бывает трудно оценить эти априорные вероятности. Например, как узнать вероятность того, что произвольная вишня на входе сортировочной системы окажется поврежденной? Если вероятность этого события зависит от погоды и сборочных бригад, то при каждой смене условий для получения необходимой информации может потребоваться слишком много эталонных данных.

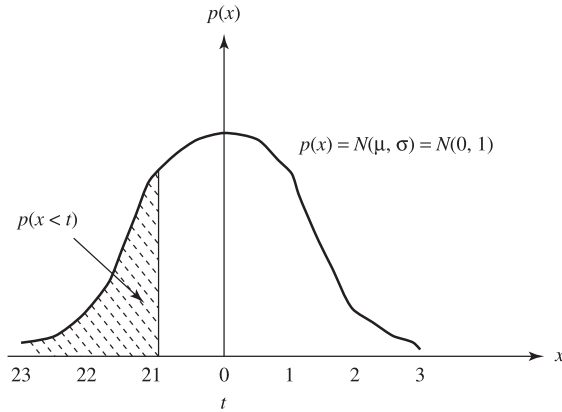
#### 4.10.1 Параметрические модели распределений

На практике необходимо каким-то образом обеспечить вычисление  $p(x | \omega_i)$ . Эмпирический метод предполагает разбиение диапазона  $x$  на интервалы, подсчет частот появления  $x$  среди образцов для каждого интервала, а затем сохранение результата в массиве или гистограмме. Можно предложить сглаживание этих данных гладкой сплайновой функцией, чтобы получить функцию вероятности для всех возможных  $x$ . Отметим, что необходимо нормировать функцию вероятности так, чтобы сумма по всем возможным значениям  $x$  равнялась 1.0. Если распределение  $x$  похоже на одну из известных параметрических моделей, то можно представить распределение с помощью небольшого количества характеризующих его параметров. Обычно используются распределения Пуассона, экспоненциальное и нормальное (Гауссовское). Кривая нормального распределения — это хорошо известная «колоколообразная» кривая (рис. 4.16).

**38 Определение.** Нормальное распределение характеризуется математическим ожиданием  $\mu$  и среднеквадратическим отклонением  $\sigma$ :

$$p(x) = N(\mu, \sigma)(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\left(\frac{x - \mu}{\sigma}\right)^2\right] \quad (4.9)$$

В учебниках по статистике можно найти описание «критерия  $\chi^2$ » для проверки нормальности распределения экспериментальных данных (например, см. учебник [Hogg, Craig, 1970]). Математическое ожидание (среднее значение) и среднеквадратическое отклонение легко вычислить по исходным данным и таким образом построить модель нормального распределения. Нормальная модель часто используется благодаря простоте и ряду других удобных математических свойств, даже если она служит лишь грубым приближением к реальным данным.



$t$	$p(x < t)$	$t$	$p(x < t)$	$t$	$p(x < t)$
-3.0	0.0014	-2.0	0.0227	-1.0	0.1587
-2.9	0.0019	-1.9	0.0287	-0.9	0.1841
-2.8	0.0026	-1.8	0.0359	-0.8	0.2119
-2.7	0.0035	-1.7	0.0446	-0.7	0.2420
-2.6	0.0047	-1.6	0.0548	-0.6	0.2743
-2.5	0.0062	-1.5	0.0668	-0.5	0.3085
-2.4	0.0082	-1.4	0.0808	-0.4	0.3446
-2.3	0.0107	-1.3	0.0968	-0.3	0.3821
-2.2	0.0139	-1.2	0.1151	-0.2	0.4207
-2.1	0.0179	-1.1	0.1357	-0.1	0.4602
		0.0	0.5000		

С учетом симметричности распределения заполним таблицу в диапазоне от 0.0 до 3.0. Например,

$$\begin{aligned}
 p(-2.0 < x < 1.0) &= p(-2.0 < x < 0.0) + p(0.0 < x < 1.0) \\
 &= [p(x < 0.0) - p(x < -2.0)] + p(-1.0 < x < 0.0) \\
 &= [0.5000 - 0.0227] + 0.1587 = 0.6360.
 \end{aligned}$$

**Рис. 4.16.** Нормальное распределение с математическим ожиданием  $\mu = 0$  и среднеквадратическим отклонением  $\sigma = 1$ .

### Упражнение 4.5

Как можно было бы оценить *априорные* вероятности следующих событий?  
 (а) Покупатель среди товаров в магазине выбирает шпинат. (б) Человек, проверяемый системой безопасности банкомата, оказывается мошенником, пользующимся чужой идентификационной картой. (в) Взятая наугад темная вишня является поврежденной. (г) Человек старше 40 лет болеет определенным заболеванием.

Для моделирования распределения эталонных образцов класса некоторой *параметрической модели*, например, нормального распределения, известны простые формулы для сравнения вероятностей в байесовской процедуре при-



нения решений, как показано на рис. 4.2. Если известны распределения  $p(x | \omega_i)$  для каждого класса  $i$ , то рис. 4.16 можно использовать для выбора пороговых значений  $x$  для разделения классов. Более того, вероятностная модель может быть непосредственно использована для оценки вероятности ошибки, так как теперь есть формула для расчета площади областей ошибок, показанных на рис. 4.15.

---

#### Упражнение 4.6. Классификация монет (Б)

Обратитесь к упражнению 4.1, в котором для классификации монет требовалось измерить их диаметр и толщину. Используйте данные о монетах трех номиналов. (а) Пусть признак  $x$  — это толщина монеты. Вычислите математическое ожидание и среднеквадратическое отклонение для каждого из трех классов. Можно ли выбрать пороговые значения  $t_1$  и  $t_2$  такие, чтобы разделить классы с общим коэффициентом ошибок менее 5%? Объясните свой ответ. (б) Повторите задание (а), используя в качестве признака  $x$  диаметр монеты.

### 4.11 ПРИНЯТИЕ РЕШЕНИЙ ПРИ ОБРАБОТКЕ МНОГОМЕРНЫХ ДАННЫХ

Во многих современных практических задачах размерность  $d = 10$  и более является обычной величиной. Как отмечалось выше, классификация по расстоянию до ближайших соседей может применяться для векторов признаков любой размерности. Существуют параметрические вероятностные модели для многомерных векторов признаков; читатель может обратиться к списку литературы за математическим описанием этих моделей. В данном разделе кратко рассмотрим понятие многомерной структуры класса. Хорошее интуитивное понимание упростит знакомство с увлекательными современными исследованиями, приведенными в списке литературы.

Рассмотрим два класса эталонных образцов в трехмерном пространстве. Форма каждого класса напоминает дерево, причем оба дерева «растут вместе». Данные класса 1 имеют форму клена и приблизительно их можно описать большой сферой. Данные класса 2 имеют форму сосны, высокой и намного более узкой, чем клен. Приблизительно их можно описать эллипсоидом, у которого большая ось намного больше двух его малых осей. Образцы класса 1 соответствуют листьям клена, а образцы класса 2 — иголкам сосны. Кроме того, допустим, что сосна прорастает через крону клена выше него. Задача классификации неизвестного трехмерного вектора признаков  $\mathbf{x}$  требует отнесения его к известной структуре образцов в трехмерном пространстве. Если  $\mathbf{x}$  лежит внутри кроны клена, но не близко к стволу сосны, то скорее всего  $\mathbf{x}$  является кленом (класс 1). С другой стороны, если  $\mathbf{x}$  лежит вне кроны клена близко к стволу сосны, то скорее всего  $\mathbf{x}$  будет сосной (класс 2). В пространстве существуют области, где сложно однозначно принять решение

из-за перекрытия двух классов. Наиболее важным моментом в этом примере является понимание того, что структура образцов в многомерном пространстве позволяет нам не только принимать осмысленные решения, но и понимать возможные ошибки. Структура пространства может быть представлена в виде большой базы данных образцов, структур данных с информацией о подмножествах образцов, или с помощью параметрических геометрических моделей подмножеств образцов.

---

**Упражнение 4.7.** Взаимосвязь ошибок и значения порога бинаризации\*

В этом упражнении рассматривается возможная ошибка вычисления значения площади, связанная с выбором порога для бинаризации изображения с целью отделения объекта от фона. Примем следующие допущения.

- На изображении размерами  $512 \times 512$  пикселей образ некоторого объекта занимает *точно* 3932 пиксела. (Частично заполненных пикселей нет. Границы объекта точно соответствуют границам пиксела. Также отсутствует размытие соседних пикселей из-за дефектов объектива.)
  - Вследствие неоднородности поверхности объекта яркость пикселей на его изображении описывается распределением  $N(80, 5)$  (нормальное распределение с математическим ожиданием 80 и среднеквадратическим отклонением 5).
  - Аналогично, яркость фона описывается распределением  $N(50, 10)$ .
  - Яркость каждого пиксела не зависит от яркости соседних пикселей.
1. Изображение бинаризовано с порогом по яркости 70, так что  $\text{LABEL}[\mathbf{r}, \mathbf{c}] = 1$  при  $\mathbf{I}[\mathbf{r}, \mathbf{c}] \geq 70$  и  $\text{LABEL}[\mathbf{r}, \mathbf{c}] = 0$  в противном случае. Сколько пикселей может быть помечено в качестве пикселей объекта?
  2. Где на изображении расположены пиксели, отнесенные к фону, хотя они на самом деле должны быть отнесены к объекту? (Это *ложные пропуски*.)
  3. Где на изображении расположены пиксели, отнесенные к объекту, хотя на самом деле это пиксели фона? (Это *ложные срабатывания*.)
  4. Какую (в процентах) ошибку можно ожидать при вычислении площади объекта простым подсчетом количества единичных пикселей на маркированном изображении?
  5. \* Предположим, что из маркированного изображения был удален шум типа «соль и перец». В результате было сформировано новое изображение, в котором каждый пиксел заменен на значение своих соседей, если все эти четырехсвязные соседи имеют одинаковое противоположное значение. Какая (в процентах) ошибка ожидается при вычислении площади объекта подсчетом количества '1' на этом новом маркированном изображении?

Приведем еще один поясняющий трехмерный пример. Допустим, что эталонные образцы класса 1 имеют структуру цилиндрической пружины, или,

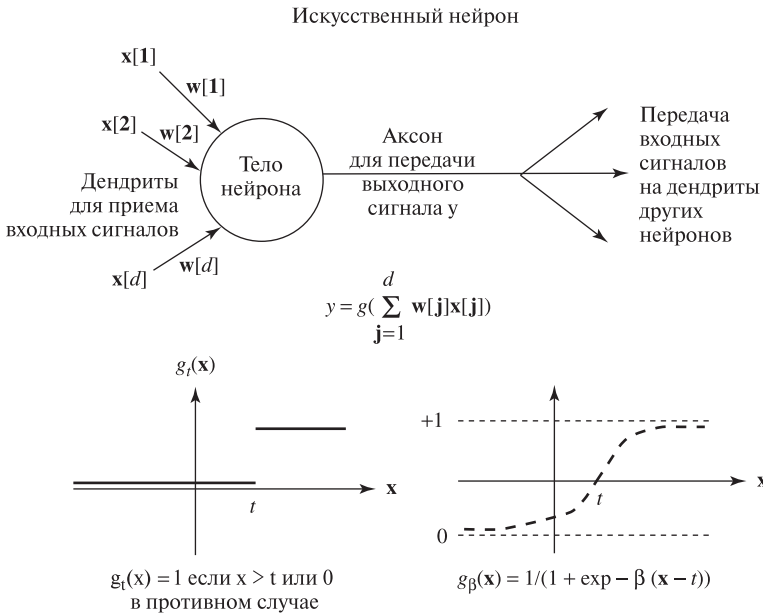
спирали, а образцы класса 2 имеют структуру стержня, расположенного по оси спирали. (Или представьте себе две свитых вместе пружины, которые можно увидеть в скобяной лавке.) Эти классы хорошо структурированы, в действительности они одномерные и могут быть легко разделены *при условии, что их структура известна*. Классификатор по близости к центру здесь бесполезен, так как центры классов совпадают. Масштабирование по одному из измерений не будет работать, так как образцы останутся переплетенными. Классификация по ближайшему соседу будет работать, но потребуются хранить много образцов. Практической альтернативой является приближение данных спирали моделью в виде композиции нескольких стержней. Стержень можно представить в виде цилиндрического фрагмента. Классификацию можно произвести на основе простых геометрических вычислений для проверки, лежит ли неизвестная точка  $x$  внутри одного из цилиндров. Лучшей альтернативой может оказаться использование формулы спирали, параметризованной по длине оси, радиусу и коэффициенту подъема.

В завершение рассмотрения этих мысленных экспериментов отметим ряд важных свойств. Во-первых, важно понять *внутреннюю структуру и размерность* эталонных данных. Структуру можно представить с помощью геометрических или статистических моделей: наличие моделей позволяет упростить вычисления при принятии решения, чтобы не выполнять поиск в большой неструктурированной базе данных образцов. Во-вторых, естественная структура данных может иметь форму, не выровненную по осям пространства измерений. Например, ось сосны или спирали не обязательно направлена по оси  $x[1]$ ,  $x[2]$  или  $x[3]$ . Методы для обнаружения структуры или преобразования координат представлены в работах, перечисленных в списке литературы.

## 4.12 САМООБУЧАЮЩИЕСЯ МАШИНЫ

Прервемся для того, чтобы подвести итог и отметить тот важный факт, что методы, до сих пор рассматривавшиеся в данной главе, демонстрируют основной тип обучения машин, называемый *супервизорным обучением*, или *обучением с учителем*. Мы полагали, что для всех распознаваемых классов доступны маркированные эталонные образцы. Другими словами, «учитель» знает структуру данных и желаемые результаты. Обучение без *учителя*, или *кластеризация*, также возможно. При таком подходе машина сама должна определить структуру классов, т. е. что за классы существуют и сколько их. По этому вопросу читатель может обратиться к дополнительной литературе.

При выполнении классификации по ближайшему соседу все эталонные данные просто загружаются в память и для распознавания неизвестного объекта надо перебрать их все. Поведение машины в процессе распознавания полностью определено эталонными образцами. При использовании параметрических моделей параметры моделей классов определяются по эталонным данным и используются для моделирования всего пространства возможных объектов. Ниже в дополнительном разделе описывается, как реализуется супервизорное



**Рис. 4.17.** Простая модель нейрона и две возможные функции для вычисления выходного сигнала.

обучение с использованием классифицирующих функций, которые были разработаны для моделирования нейронов живых организмов. Обучение машин представляет собой область интенсивных современных исследований и разработок, о чем читатель может узнать в дополнительной литературе.

### 4.13 ИСКУССТВЕННЫЕ НЕЙРОННЫЕ СЕТИ\*

Нейроны живых организмов благодаря своей способности к обучению исследовались применительно к задачам машинного обучения. Простая модель нейрона показана на рис. 4.17. Хотя она является лишь приближением к биологической модели, но приобрела большую важность в качестве модели для вычислений. Сети из подобных модельных нейронов, называемые *искусственными нейронными сетями* (artificial neural networks, ANN), были успешно применены во многих задачах машинного зрения, особенно благодаря своей способности к обучению. Нейронные сети могут быть обучены сложной структуре образцов в многомерном пространстве с меньшими затратами памяти, чем требуется для классификации по ближайшим соседям. Нейронные сети допускают реализацию в форме параллельных алгоритмов. В данной главе дается краткое введение в область нейронных сетей; более подробную информацию об этой большой и быстро развивающейся области можно найти в работах из списка литературы.

### 4.13.1 Персептрон

На рис. 4.17 показано, что искусственный нейрон (artificial neuron, AN) посредством дендритических соединений получает  $d$  входных сигналов  $\mathbf{x}[j]$  от других нейронов или, возможно, от ячеек датчика. Тело нейрона умножает все входные значения на соответствующие весовые коэффициенты  $\mathbf{w}[j]$  и суммирует их. Выходной сигнал нейрона  $y$  посылается по единственному аксону, который в конечном счете разветвляется на множество дендритических соединений, что дает входные сигналы для других нейронов сети. Один из способов определения значения  $y$  по сумме взвешенных входных сигналов заключается в том, что эта сумма сравнивается с пороговым значением  $t$  и выходное значение принимается равным  $y = 1$ , если сумма больше порогового значения, или  $y = 0$  в противном случае. Подобное бинарное поведение выходного сигнала показано на рис. 4.17 слева внизу. Для получения гладкого выходного сигнала в диапазоне от 0 до 1 может быть использована сигмоидная функция (см. рис. 4.17, справа внизу). Параметр  $\beta$  представляет наклон (усиление) при  $x = t$ , который определяет, как входное значение масштабируется для вычисления выходного сигнала в окрестности  $x = t$ . Для удобства обозначений и программирования, в  $\mathbf{w}[0]$  хранится отрицательное пороговое значение нейрона  $t$  и соответствующее входное значение  $\mathbf{x}[0]$  приравнивается к 1.0, так что уравнение для вычисления выходного сигнала имеет вид (4.10). Нейрон обучается путем подстройки весовых коэффициентов  $\mathbf{w}[j]$  для получения заданных выходных значений на известных эталонных векторах  $\mathbf{x}$ .

$$y = g \left( \sum_{j=0,d} w[j]x[j] \right) \quad (4.10)$$

---

#### Упражнение 4.8. Моделирование искусственного нейрона

(а) Изучите поведение искусственного нейрона с двумя входами  $\mathbf{x}[1]$  и  $\mathbf{x}[2]$ , с весовыми коэффициентами  $\mathbf{w}[1] = 0.8$  и  $\mathbf{w}[2] = 0.3$ , пороговым значением  $t = 1.0$  и ступенчатой выходной функцией  $G(\mathbf{x})$ . Постройте график результатов для 16 возможных комбинаций входных значений  $\mathbf{x}[1]$  и  $\mathbf{x}[2]$  из набора 0, 1, 2, 3. Помечайте точку на графике символом '1' при выходном значении 1 и символом '0' при выходном значении 0. (б) Постройте другой график, на этот раз пользуясь гладкой сигмоидной функцией при  $\beta = 4$ , оставляя все остальные условия задачи без изменений. Обратите внимание, что теперь выходные значения будут вещественными, а не дискретными значениями 0 или 1.

---

#### Упражнение 4.9. Реализация логических функций И, ИЛИ и НЕ с помощью искусственного нейрона.

1. Спроектируйте одиночный искусственный нейрон, который ведет себя аналогично логическому элементу ИЛИ. Входные сигналы  $\mathbf{x}[1]$  и  $\mathbf{x}[2]$

могут принимать только логические значения 0 или 1. Выходное значение нейрона должно быть 1, если один или оба входа равны 1, или 0, если оба входа равны 0. Напомним, что  $x[0] = 1$  и что пороговое значение равно  $-w[0]$ . Дополните набор весовых множителей для определения нейрона. Постройте график для четырех возможных входных комбинаций, используя двумерные оси, и покажите границу решения, реализуемую нейроном.

2. Повторите задание 1) для логической функции И. Ее выходное значение равно 1, только если оба входа равны 1.
3. Покажите, как искусственный нейрон с единственным входом может вести себя аналогично функции НЕ. Если входное значение нейрона равно 0, то выход равен 1, и наоборот.

Вычислительные возможности простейшего искусственного нейрона представляют большой теоретический и практический интерес. Из упражнения 4.9 известно, что искусственный нейрон может моделировать логические функции И, ИЛИ и НЕ. Значимость этого факта в том, что любую логическую функцию можно реализовать путем каскадного объединения нескольких искусственных нейронов. В Упражнении 4.10 демонстрируется, что одиночный нейрон не может воспроизвести одну простую логическую функцию — ИСКЛЮЧАЮЩЕЕ ИЛИ. Многие другие важные функции также нельзя реализовать на одном нейроне: результаты, опубликованные Минским и Пейпертом [Minsky, Papert, 1989], затормозили исследования в области искусственных нейронных сетей на довольно длительное время. Через несколько лет появился поток успешных работ по многослойным нейронным сетям, которые имеют более сложную структуру, но за счет этого меньше вычислительных ограничений. За теоретическими объяснениями вычислительных возможностей искусственных нейронных сетей читатель может обратиться к дополнительной литературе; мы вернемся к рассмотрению простой версии алгоритма обучения для одиночного искусственного нейрона.

Предположим, что два класса двумерных эталонных образцов можно разделить прямой: это мог бы быть пример с рис. 4.4, если мы удалим по одному образцу 'X' и 'O' из области перекрытия классов. Очевидно, тогда можно определить параметры разделяющей прямой и построить нейрон для принятия классификационного решения. Для трехмерных образцов можно пользоваться разделяющей плоскостью; в  $d$ -мерном пространстве нам потребовалась бы гиперплоскость, но суть метода остается прежней. Довольно неожиданно, что если разделяющая гиперплоскость существует для проблемы двух классов, то существует простой обучающий алгоритм для поиска уравнения гиперплоскости по данным эталонных образцов двух этих классов (алгоритм 4.3). Доказательство сходимости этого алгоритма к разделяющей гиперплоскости выходит за рамки данной книги, оно приведено в книге Дуды и Харта [Duda, Hart, 1973].

Алгоритм обучения персептрона начинается с выбора случайного множества весовых коэффициентов (включая пороговое значение). Алгоритм циклически обрабатывает маркированные эталонные образцы  $x$  и всякий раз, когда весовой вектор (персептрон) дает положительный выход для образца из Класса 1, алгоритм вычитает  $gain * x$  из весового вектора. Аналогично, если для образца из Класса 2 персептрон дает отрицательный выход, то к весовому вектору прибавляется  $gain * x$ . Таким образом текущая разделяющая прямая сдвигается в соответствующем направлении, согласно выходному сигналу от текущего образца. Масштабный множитель  $gain$  определяет величину изменений. Описанные действия выполняет процедура `training_pass`. После одного прохода по всем образцам вызывается процедура `check_samples` для подсчета образцов, которые не удалось правильно классифицировать с помощью нейрона с текущими весовыми коэффициентами. Если таких образцов нет, то алгоритм возвращает найденное решение. Иначе, если еще не было выполнено максимально допустимого количества проходов, то выполняется еще один проход обучения, на этот раз с множителем  $gain$ , уменьшенным в 2 раза. Существуют и другие реализации общего алгоритма обучения.

На рис. 4.18 показаны выходные данные программы обучения персептрона. Выбрано такое множество эталонных образцов, что все образцы класса 1 лежат ниже прямой  $y = 1 - x$ , а все образцы класса 2 — выше этой прямой. В пространстве между образцами существует «коридор», и алгоритм очень быстро находит прямую для разделения классов  $-1 + 5/4x_1 + 5/4x_2 = 0$ . Как показано в выходных данных, каждый образец Класса 1 дает отрицательный отклик нейрона, а каждый образец Класса 2 — положительный.

Структура основного алгоритма обучения проста, но для него можно перечислить ряд нетривиальных аспектов. (1) Какую последовательность образцов следует использовать для быстрого обучения? Теоретически известно, что для обеспечения сходимости каждый образец должен быть представлен случайное количество раз. Некоторые алгоритмы повторяют обучение на данном образце до тех пор, пока он не будет корректно классифицирован, и только после этого переходят к другим образцам. (2) Сходимость зависит от используемого масштабного множителя ( $gain$ ). Программа, для которой показаны выходные данные, уменьшает этот множитель вдвое при каждом проходе по всем эталонным образцам. (3) Для лучшей производительности на будущих данных может иметь смысл использовать алгоритм для поиска *наилучшей* прямой между классами, а не любой разделяющей прямой. (4) Если обучение занимает много времени, то как мы можем узнать, не происходит ли это по причине невозможности разделения образцов? (5) Как можно модифицировать алгоритм обучения, чтобы в случае линейно неразделимых эталонных образцов можно было найти прямую, соответствующую наименьшему количеству ошибок классификации? Читатель может рассмотреть перечисленные вопросы в качестве дополнительных упражнений и экспериментов.

```

Класс 1 = { ( 0 , 0.5 ), ( 0.5 , 0 ), ( 0 , 0 ), ( 0.25 , 0.25 ) }
Класс 2 = { ( 0 , 1.5 ), ( 1.5 , 0 ), ( 0.5 , 1 ), ( 1 , 0.5 ) }
Начальное значение масштабного множителя gain = 0.5
Предельное количество проходов = 5
Количество образцов класса 1 = 4; Количество образцов класса 2 = 4;

Значения весов в начале обучения          -1    0.5    0.5

====Настройка весовых коэффициентов====:  gain= 0.5
  Образ в виде вектора x =          1    0    1.5
Входные значения весов      -1    0.5    0.5
Выходные значения весов     -1    0.5    1.25

====Настройка весовых коэффициентов====:  gain= 0.5
  Образ в виде вектора x =          1    1.5    0
Входные значения весов      -1    0.5    1.25
Выходные значения весов     -1    1.25    1.25

Компоненты весового вектора: -1    1.25    1.25
Классификация на предмет принадлежности классу  1
  Входной вектор x / Отклик / Признак ошибки?
  1      0      0.5    -0.375    N
  1     0.5      0    -0.375    N
  1      0      0      -1        N
  1     0.25    0.25    -0.375    N

Компоненты весового вектора: -1    1.25    1.25
Классификация на предмет принадлежности классу  2
  Входной вектор x / Отклик / Признак ошибки?
  1      0      1.5    0.875    N
  1     1.5      0    0.875    N
  1     0.5      1    0.875    N
  1      1      0.5    0.875    N

Ошибки для класса 1: 0    Ошибки для класса 2: 0
Окончательные значения весовых коэффициентов:    -1    1.25    1.25

```

**Рис. 4.18.** Выходные данные программы, реализующей алгоритм обучения персептрона для классификации в случае двух линейно разделяемых классов.

#### Упражнение 4.10. Персептрон для реализации функции ИСКЛЮЧАЮЩЕЕ ИЛИ

Покажите, что с помощью одиночного искусственного нейрона нельзя реализовать функцию ИСКЛЮЧАЮЩЕЕ ИЛИ. Для этого отобразите графически указанные далее входные данные и попытайтесь найти разделяющую прямую. Входные данные для положительного отклика — это пары  $(0, 1)$ ,  $(1, 0)$ , а входные данные для отрицательного отклика —  $(0, 0)$ ,  $(1, 1)$ .



### Вычисление весового вектора $w$ для разделения классов 1 и 2.

$S_1$  и  $S_2$  — множества из  $n$  эталонных образцов каждое.

**gain** — масштабный множитель, используемый для изменения  $w$  при неправильной классификации  $x$ .

**max\_passes** — максимальное количество проходов по всем эталонным образцам.

```

procedure Perceptron_Learning(gain, max_passes, S1, S2)
{
  ввод множеств эталонных образцов S1 и S2;
  случайный выбор весового вектора w;
  “NE — общее количество неправильно классифицированных образцов”
  NE = check_samples (S1, S2, w);
  while (NE > 0 and passes < max_passes)
  {
    training_pass (S1, S2, w, gain);
    NE = check_samples (S1, S2, w);
    gain = 0.5 * gain;
    passes = passes + 1;
  }
  вывод количества ошибок NE и весового вектора w;
}
procedure training_pass (S1, S2, w, gain);
{
  for  $i = 1$  to (размер Sk)
  {
    “Символ скалярного произведения  $\circ$  обозначает вычисление,
    реализуемое искусственным нейроном”
    взять следующий вектор  $x$  из S1;
    if ( $w \circ x > 0$ ) w = w - gain *  $x$ ;
    Взять следующий вектор  $x$  из S2;
    if ( $w \circ x < 0$ ) w = w + gain *  $x$ ;
  }
}

```

**Алгоритм 4.3.** Алгоритм обучения персептрона в случае двух линейно разделяемых классов

### Упражнение 4.11. Реализация алгоритма обучения персептрона

Напишите программу, реализующую алгоритм обучения персептрона для произвольного  $d$ -мерного вектора признаков  $x$ . Протестируйте ее с исполь-

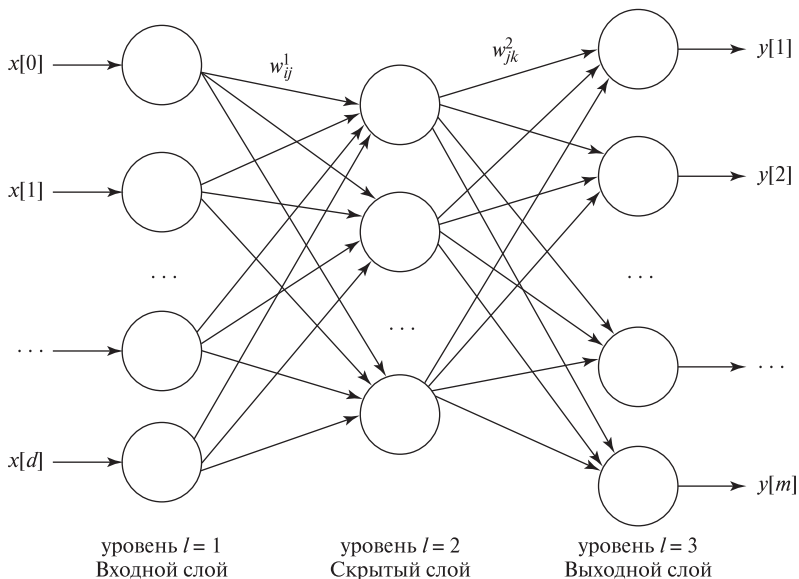
зованием двумерных векторов и покажите, что она допускает обучение для функционирования в режиме логических функций ИЛИ и И. Покажите, что для функции ИСКЛЮЧАЮЩЕЕ ИЛИ процесс обучения не сходится. Протестируйте программу на следующих двух классах искусственных трехмерных образцов: Класс 1 — некоторое множество случайных точек в первом октанте (все  $x_1, x_2, x_3$  положительны), а Класс 2 — некоторое множество точек в любом другом октанте.

### 4.13.2 Многослойная сеть прямого распространения

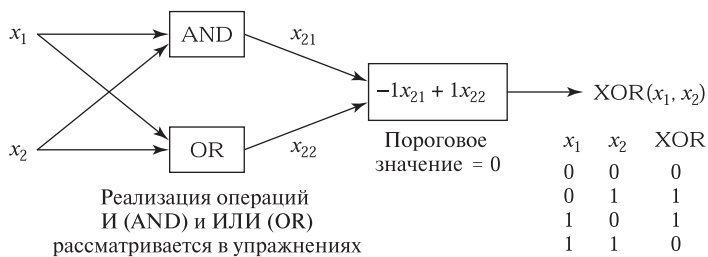
*Нейронная сеть прямого распространения* — это особый вид искусственных нейронных сетей, в которых каждый нейрон расположен на некотором уровне (в слое)  $l$ . Нейрон в слое  $l$  получает входные сигналы от всех нейронов предыдущего слоя  $l - 1$  и передает выходной сигнал всем нейронам слоя  $l + 1$ . Обратимся к рис. 4.19. Входные сигналы нейронов нижнего слоя 1 можно считать входными сигналами от датчика, а выходные сигналы последнего слоя  $L$  — результатами классификации. В качестве результата классификации можно рассматривать  $c$  такое, что  $y[c]$  имеет максимальное значение; или можно рассматривать выходные сигналы всех нейронов для выполнения *нечеткой классификации*. Промежуточные слои сети (т. е. все слои, кроме 1 и  $L$ ) называются *скрытыми*. Обратной связи от вышележащих слоев к нижележащим нет, этим и объясняется термин *прямое распространение*. Такая нейронная сеть работает подобно комбинационной схеме, в том смысле, что выходные данные вычисляются по входным без учета предыдущих входных значений.

В предыдущих примерах было показано, что одиночные нейроны могут вести себя как логические функции И, ИЛИ и НЕ. Это означает, что нейронные сети прямого распространения могут реализовать любую комбинационную логическую функцию. Поэтому такие сети оказываются удивительно мощными и могут моделировать поведение многих различных компьютерных программ. Более того, так как нейронные сети не ограничены логическими значениями, они могут представлять очень сложные геометрические разбиения  $d$ -мерного пространства и могут адаптивно обучаться таким структурам по эталонным образцам. На рис. 4.20 показано, как нейронная сеть прямого распространения может вычислить функцию ИСКЛЮЧАЮЩЕЕ ИЛИ, что невозможно сделать с помощью одиночного нейрона. Первый слой использует нейроны для реализации функций И и ИЛИ, как было рассмотрено в упражнениях. На последнем уровне есть только один нейрон: он имеет весовой вектор  $\mathbf{w} = [0, -1, 1]$  и выдает на выход 1, только если  $-1x_1 + 1x_2$  принимает положительное значение. Чтобы увидеть, как обучить многослойную сеть геометрической структуре сложного множества образцов, читатель может рассмотреть следующие далее упражнения.

Сеть прямого распространения может обучаться путем настройки весовых коэффициентов по последовательности эталонных образцов. В алгорит-



**Рис. 4.19.** Многослойная нейронная сеть прямого распространения: все нейроны в слое  $l$  получают входные сигналы от всех нейронов слоя  $l - 1$  и передают выходные сигналы всем нейронам слоя  $l + 1$ .



**Рис. 4.20.** Реализация функции ИСКЛЮЧАЮЩЕЕ ИЛИ (XOR) с помощью многослойной нейронной сети прямого распространения.

ме обучения, называемом *алгоритмом обратного распространения*, ошибки классификации передаются от выходных слоев назад, к входным слоям. Для вычисления выходного сигнала используется сигмоидная функция, что обеспечивает непрерывное управление взаимосвязью вход/выход. Вопросы применения и реализации алгоритма обратного распространения описаны в работах из списка литературы. В последнее время ряд успешных приложений обратного распространения и других алгоритмов обучения привели к развитию новых направлений и в области распознавания образов, и в области машинного обучения. См. список литературы для ссылок на работы с описанием других типов сетей и их приложений.

---

**Упражнение 4.12.** Искусственная нейронная сеть для распознавания двумерного класса с треугольной структурой

Постройте нейронную сеть прямого распространения, которая будет давать на выходе 1 для всех двумерных точек  $x$  внутри треугольника с вершинами  $(3, 3)$ ,  $(6, 6)$  и  $(9, 1)$  и 0 для всех точек за пределами треугольника. Примените ступенчатую выходную функцию  $G(x)$ . Подсказка: используйте три искусственных нейрона на первом слое для задания границ класса, которыми являются стороны треугольника. Второй слой сети должен интегрировать три выходных сигнала с первого слоя.

---

**Упражнение 4.13.** Искусственная нейронная сеть для задачи трех классов.

Выясните, как двухслойная нейронная сеть прямого распространения может распознавать двумерные входные векторы следующих *непересекающихся* классов. Векторы класса 1 лежат внутри некоторого треугольника; векторы класса 2 лежат внутри квадрата; а векторы класса 3 лежат внутри некоторого пятиугольника. С учетом результатов предыдущего упражнения покажите, что существует нейронная сеть для распознавания каждого отдельного класса (не надо работать с отдельными прямыми или уравнениями, просто назовите подсети *треугольник*, *квадрат* и *пятиугольник*). Второй слой должен генерировать выходное значение, обозначающее класс входного вектора на первом слое.

## 4.14 ЛИТЕРАТУРА

Известный учебник Дуды и Харта [Duda, Hart, 1973] остается полезным руководством по задачам и методам классического статистического распознавания образов (в 2000 г. было выпущено новое издание). Другим классическим учебником является книга [Fukunaga, 1972, 1990]. К этим книгам можно обратиться для изучения теории байесовской классификации  $d$ -мерных векторов признаков. В них показана важность *ковариационной матрицы* для моделирования многомерной структуры класса. Существует ряд хороших вводных книг по теории вероятностей и статистике, например, [Hogg, Craig, 1970] и [Feller, 1957]. С их помощью можно изучить вопросы, касающиеся распределения вероятностей, таких, как нормальное распределение и распределение хи-квадрат. В книге [Jain и др., 2000] дан обзор статистического распознавания образов, в который включены многие последние результаты.

*Собственные векторы* ковариационной матрицы дают естественные направления эллипсоидальных кластеров в пространстве, а соответствующие *собственные значения* характеризуют «разброс» образцов. Кроме того, даны несколько способов, позволяющих охарактеризовать плотность вероятности. В недавно изданных книгах [Schalkoff, 1992] и [Schurmann, 1996] кроме ста-

тистического распознавания образов рассматривается также синтаксическое и структурное распознавание. Широкое, но краткое рассмотрение искусственных нейронных сетей можно найти в учебнике [Jain и др., 1996]; более подробное рассмотрение содержится в книгах [Haykin, 1994], [Hertz и др., 1991] и [Schurmann, 1996]. В книге [Tanimoto, 1995] приведено хорошее описание нейронных сетей в контексте других способов обучения машин и также показано, как использовать в качестве входных данных символические признаки. Реализация обучения перцептрона и алгоритм обратного распространения приведены на языке Лисп. Отличное теоретическое рассмотрение того, что можно и что нельзя вычислить с помощью перцептронов, содержится в в книге [Minsky, Papert, 1989, 1969].

1. Duda, R. O., and P. E. Hart. 1973. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York.
2. Duda, R. O., D. Stork, and P. Hart. 2000. *Pattern Classification*. John Wiley & Sons, New York.
3. Feller, W. 1957. *An Introduction to Probability Theory and Its Applications, vols. I and II*. John Wiley & Sons, New York.
4. Haykin, S. 1994. *Neural Networks: A Comprehensive Foundation*. Macmillan College Publishing, New York.
5. Hertz, J., A. Krogh, and R. Palmer. 1991. *Introduction to the Theory of Neural Computation*. Addison-Wesley, Reading, MA.
6. Hogg, R., and A. Craig. 1970. *Introduction to Mathematical Statistics*.
7. Jain, A. K., J. Mao, and K. M. Mohiuddin. 1996. Artificial neural networks: A tutorial. *IEEE Comput.* 29(3).
8. Jain, R. Duin, and J. Mao. 2000. Statistical pattern recognition, a review. *IEEE-TPAMI*. 22(1):4–37.
9. Fukunaga, K. 1990. *Introduction to Statistical Pattern Recognition*, 2nd ed. Academic Press, New York.
10. Kulkarni, A. 1994. *Artificial Neural Networks for Image Understanding*. Van Nostrand-Reinhold, New York.
11. Minsky, M., and S. Papert. 1989. *Perceptrons*, 2nd ed. MIT Press, Cambridge, MA.
12. Proakis, J. G. 1989. *Digital Communications*. McGraw-Hill, New York.
13. Quinlan, J. R. 1986. Induction of decision trees. *Machine Learning*, 1(1):81–106.
14. Schalkoff, R. 1992. *Pattern Recognition: Statistical, Structural, and Neural Approaches*. John Wiley & Sons, New York.
15. Schurmann, J. 1996. *Pattern Classification: A Unified View of Statistical and Neural Approaches*. John Wiley & Sons, New York.
16. Tanimoto, S. 1995. *The Elements of Artificial Intelligence with Common LISP*, 2nd ed. Computer Science Press, New York.

# Фильтрация и улучшение изображений

В этой главе описываются методы улучшения изображений как для использования человеком, так и для дальнейшей автоматической обработки. Например, эти методы могут потребоваться для понижения шума на изображении или для усиления или подавления некоторых деталей изображения. Два метода фильтрации изображений уже были введены в гл. 1: во-первых, для удаления изолированных черных или белых пикселей из изображений бактерий, и, во-вторых, было показано, как с помощью оператора контраста можно усилить границы между различными объектами на изображении, например, повысить контраст между картиной и стеной.

Материал данной главы относится к области *обработки изображений*, так как описанные в главе методы в качестве исходных данных получают входное изображение и в результате формируют другое, выходное изображение. В качестве синонимов термина «обработка изображений» также часто используются термины *фильтрация изображений*, *улучшение*, *повышение качества* и *предварительная обработка изображений*. Основная идея состоит в том, что считается, что наряду с некоторым сигналом или структурой, которую необходимо извлечь, изображение содержит неважные или нежелательные данные, которые требуется подавить. Решения относительно изображения принимаются на уровне отдельного пиксела (элемента изображения) и его локальной окрестности. Ранее уже было показано, как пиксели изображения можно маркировать как пиксели, принадлежащие «объекту или фону» и «границе или не границе».

Подробное изложение теории и методов обработки изображений может занять несколько книг. В данной главе подробно рассматривается всего несколько классических понятий обработки изображений. Большинство представленных методов основаны на том предположении, что значение каждого пиксела выходного изображения можно вычислить по локальной окрестности соответствующего пиксела входного изображения. Однако ряд методов повышения качества изображений являются глобальными в том смысле, что при создании выходного изображения некоторым образом используются все пиксели входно-

го изображения. Среди рассматриваемых тем наиболее важными являются две: (1) сопоставление окрестности изображения с шаблоном или маской (*корреляция*), и (2) *свертка*, представляющая собой метод, позволяющий единообразно выполнять много различных операций фильтрации.

## 5.1 ЧТО НЕОБХОДИМО СДЕЛАТЬ С ИЗОБРАЖЕНИЕМ?

Перед тем как перейти к рассмотрению методов обработки изображений, перечислим ряд задач, для решения которых эти методы необходимы. Ниже приведены несколько задач, которые можно разделить на две категории.

### 5.1.1 Повышение качества изображений

- Предположим, что во время африканского сафари вы сделали снимок льва, догоняющего антилопу. К сожалению, солнце располагалось позади животных, и большая часть изображения оказалась слишком темной. Изображение можно улучшить путем увеличения малых значений интенсивности, не меняя при этом больших значений интенсивности.
- На старой фотографии нет никаких дефектов, кроме нескольких длинных ярких царапин. Эту фотографию можно оцифровать и затем удалить царапины из полученного цифрового изображения (см. рис. 5.1.)
- Бумажный документ требуется отсканировать и преобразовать в текстовый файл. Перед применением алгоритмов распознавания символов необходимо удалить шумовые пиксели, присутствующие на фоне изображения, а также заполнить недостающие пиксели в образах символов.

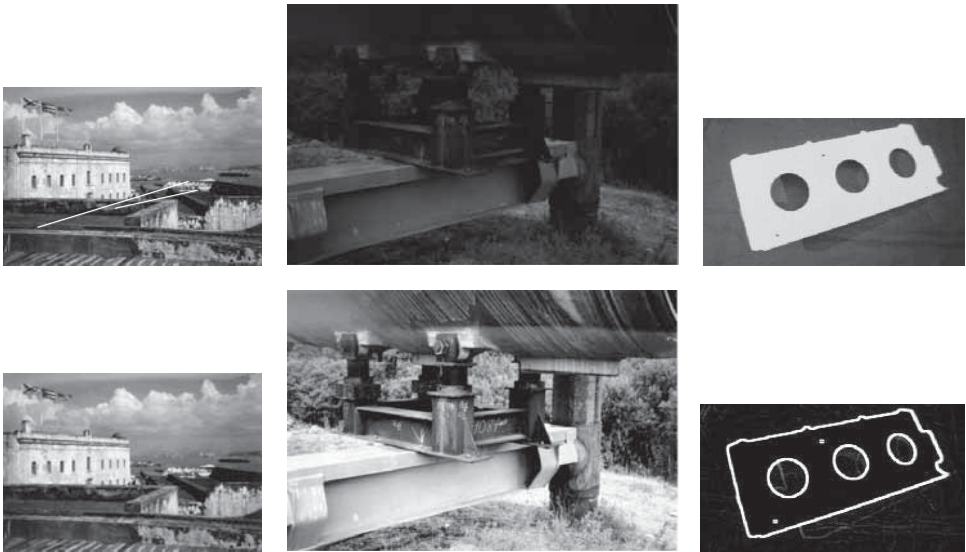
### 5.1.2 Обнаружение низкоуровневых признаков

- В процессе производства проволоки диаметром 3 мм требуется организовать цикл управления с обратной связью на основе зрительного датчика, который непрерывно измеряет диаметр проволоки. Расположение двух сторон проволоки можно установить с помощью оператора выделения края, способного точно определить границу между образом проволоки и фоном.
- Система автоматического управления автомобилем может управлять положением руля по результатам непрерывного наблюдения белых линий дорожной разметки. Белые линии на кадрах видеосъемки, получаемой с помощью направленной вперед видеокамеры, можно обнаружить путем поиска двух краев с противоположным контрастом и похожим направлением.
- Чертеж, выполненный на бумаге, требуется преобразовать в модель САПР (системы автоматизированного проектирования). В процессе этого преобразования одним из этапов обработки может быть поиск на изображении чертежа линий в виде черных полос шириной примерно 1 пиксел.

В данной главе в основном рассматриваются традиционные методы *улучшения изображений* и, в некоторой степени, методы *восстановления изображений* (см. рис. 5.2). Перед дальнейшим изложением необходимо определить и различить эти термины.

**39 Определение.** Операторы **улучшения изображения** предназначены для повышения вероятности обнаружения человеком или машиной важных деталей или объектов на изображении. Примерами таких операторов являются операторы подавления шума, сглаживания, повышения контраста и усиления краев.

**40 Определение.** Методы **восстановления изображения** предназначены для приближения испорченного изображения к идеальному состоянию. Это возможно только в тех пределах, в каких могут быть поняты и смоделированы физические процессы формирования идеального изображения и ухудшения его качества. В процессе восстановления изображения моделируется процесс, обратный процессу ухудшения качества (деградации) идеального изображения.



**Рис. 5.1.** (Слева) удаление паразитов с фотографии города Сан Жуан. (В центре) изменение шкалы уровней интенсивности на фотографии трубопровода на Аляске для обеспечения лучшей видимости деталей. (Справа) усиление краев на изображении детали самолета для обеспечения автоматического распознавания и выполнения измерений.





**Рис. 5.2.** (Слева) оригинальное изображение отпечатка пальца. (В центре) изображение, улучшенное посредством выделения и утоньшения линий. (Справа) обнаружение специальных признаков — *минюций* (minutia). Это локальные признаки папиллярного узора, которые могут использоваться для сопоставления неизвестного отпечатка с описаниями миллионов отпечатков из базы данных. (С любезного разрешения Shaoyun Chen, Anil Jain.)

## 5.2 ИЗМЕНЕНИЕ ТОНОВОГО РАСПРЕДЕЛЕНИЯ

Улучшение изображений часто выполняется путем изменения значений интенсивности (полутоновых уровней) пикселей. В большинстве программных пакетов для обработки изображений предусмотрено несколько команд для изменения вида изображения посредством преобразования значений пикселей с использованием некоторой функции, задающей способ замены входных значений интенсивности на новые выходные значения. Данный метод легко расширить таким образом, чтобы пользователь мог указать несколько различных областей изображения и применить к каждой из них различные функции преобразования — функции тонового распределения. Преобразование значений интенсивности часто называется *растяжением* (stretching), так как такая операция часто применяется для расширения динамического диапазона значений интенсивности слишком темных изображений на весь доступный диапазон уровней интенсивности.

На рис. 5.3 показано изображение, для которого была применена операция расширения динамического диапазона для растяжения диапазона используемых значений интенсивности в соответствии с двумя различными функциями тонового распределения. На рис. 5.3 (a) представлено исходное изображение и функция тонового распределения общего вида. На рис. 5.3 (b) представлен результат преобразования исходного изображения с применением функции тонового распределения вида  $f(x) = x^{0.5}$ . Эта нелинейная функция низкие значения интенсивности увеличивает сильнее, чем высокие. Применение к изображению функции тонового распределения вида  $f(x) = x^{1/\gamma}$  называется *гамма-коррекцией* (Gamma correction). Эта функция в ряде случаев может служить подходящей теоретической моделью для восстановления до исходного вида изображения, подвергнувшегося известной физической деформации. В рассматриваемом примере значение  $\gamma = 2.0$  приводит к увеличению значений

интенсивности. В данном случае уменьшение значений интенсивности (например, при  $\gamma = 0.3$ ), не имеет практического смысла, так как объекты сцены находятся в тени леса и трубопровода. На рис. 5.3 (с) представлен результат преобразования исходного изображения с использованием функции тонового распределения более сложного вида, которая была задана в интерактивном режиме. Пользователь определил функцию преобразования значений интенсивности  $g_{out} = f(g_{in})$  в интерактивной программе обработки изображений с помощью манипулятора «мышь». Программа построила гладкую сплайновую кривую по точкам, указанным пользователем. Функции на рис. 5.3 позволяют растянуть (расширить), по крайней мере, некоторые интервалы уровней интенсивности, чтобы на выходном изображении различие значений пикселей стало более заметным. Неоднородность данных изображения будет увеличиваться в тех интервалах значений интенсивности, в которых угловой коэффициент функции  $f(x)$  превышает 1.

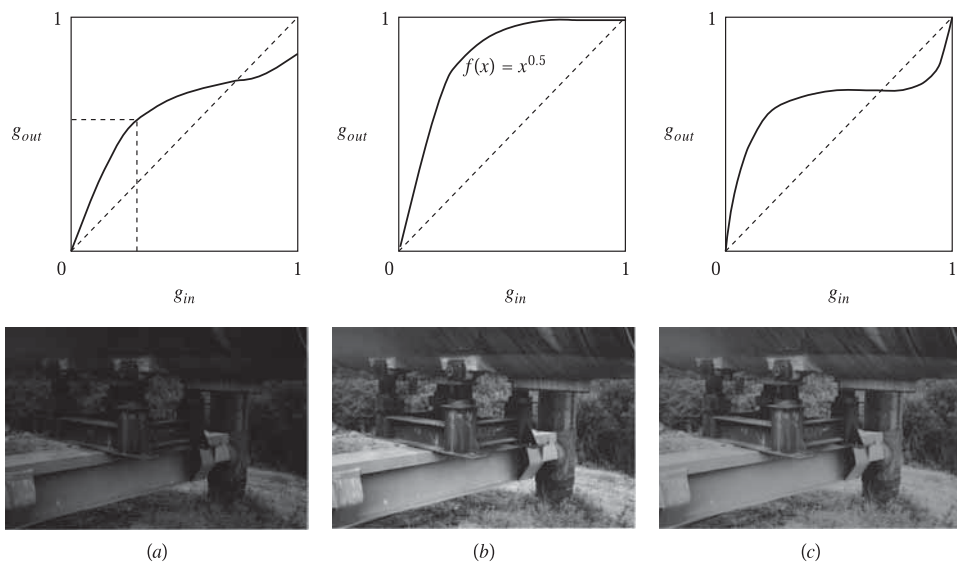
**41 Определение. Точечный оператор**, применяемый к изображению, это оператор, который определяет значение выходного пиксела по значению только одного входного пиксела,  $\text{Out}[\mathbf{x}, \mathbf{y}] = f(\text{In}[\mathbf{x}, \mathbf{y}])$ . Функция оператора  $f$  может зависеть от некоторых глобальных параметров.

**42 Определение. Оператор увеличения контраста** — это точечный оператор, в котором кусочно-гладкая функция  $f(\text{In}[\mathbf{x}, \mathbf{y}])$  применяется к входным значениям интенсивности для усиления важных деталей изображения.

Так как точечные операторы преобразуют один входной пиксел в один выходной, то они могут применяться к массиву пикселей в произвольном порядке или параллельно. Специфические функции тонового распределения, в том числе немонотонные (как на рис. 5.3), могут оказаться очень полезными для улучшения качества изображений, воспринимаемых человеком, например, в дизайне и журналистике. Однако в некоторых областях, таких, как рентгенология, подобные преобразования необходимо применять с осторожностью, чтобы не изменить значимую шкалу значений интенсивности, которой пользуются люди-эксперты и на которую могут быть настроены различные датчики. В заключение отметим, что после монотонного растяжения диапазона значений интенсивности (когда для всех значений интенсивности  $g_2 > g_1$  выполняется  $(f(g_2) > f(g_1))$ ) результаты работы некоторых алгоритмов компьютерного зрения могут не измениться, несмотря на то, что наблюдающие за обработкой люди могут отмечать улучшение качества изображений.

### 5.2.1 Выравнивание гистограммы

Для улучшения изображений часто применяется операция *выравнивания гистограммы* (histogram equalization). При этом к изображению применяется оператор, на который накладываются два условия: (а) в выходном изображении должны использоваться все допустимые значения интенсивности и



**Рис. 5.3.** (Верхний ряд) функции тонового распределения  $f$ , задающие преобразование значений интенсивности. (Нижний ряд) выходные изображения, полученные после преобразования исходного изображения (показано внизу слева) с использованием различных функций  $f$ . (а) Исходное, довольно темное, изображение трубопровода на Аляске и пример функции тонового распределения общего вида. (б) Гамма-коррекция изображения с функцией тонового распределения  $f(x) = x^{0.5}$  увеличивает значения интенсивности темных пикселей сильнее, чем светлых. (с) Результат преобразования изображения с использованием программы, позволяющей оператору задавать функцию преобразования в интерактивном режиме. Показана функция, которая повышает яркость темных пикселей и понижает яркость светлых. Отметим, что четкость отдельных объектов сцены различается на разных изображениях.

(б) выходное изображение должно содержать примерно равное количество пикселей для каждого значения интенсивности. Условие (а) обеспечивает хорошее восприятие изображения, а условие (б) является дополнительным и его эффективность должна оцениваться из экспериментов. На рис. 5.4 показаны результаты применения операции выравнивания гистограммы к двум изображениям. Можно увидеть, что изменение кривой тонового распределения изображения действительно приводит к изменению вида некоторых областей; например, после преобразования становится легче различить швы на арке. (Преобразование с кривой тонового распределения, показанной на рис. 5.3, может дать лучшие результаты. Почему?) Изображение лица, вырезанное из изображения большого размера, и арка, показанные на рис. 5.4, содержат небольшое количество пикселей с малыми значениями интенсивности. Условие (б) приведет к тому, что большие однородные области, например, небо, после



**Рис. 5.4.** Выравнивание гистограммы преобразует динамический диапазон изображения так, чтобы выходное изображение содержало все допустимые значения интенсивности и чтобы при этом количество пикселей с различными значениями было примерно одинаковым. Изображения после выравнивания гистограммы показаны справа. (С любезного разрешения Ф. Виосса.)

выравнивания гистограммы будут содержать больше различных значений интенсивности. Следовательно, они будут иметь более выраженную текстуру. Это может оказаться полезным (или нет) для интерпретации изображений.

Условия (а) и (б) означают, что в выходном изображении будут присутствовать все допустимые значения интенсивности (полутонные уровни)  $z = z_1, z = z_2, \dots, z = z_n$ , причем каждое значение  $z_k$  будет использоваться примерно  $q = (R \times C)/n$  раз. Здесь  $R$  и  $C$  представляют количество строк и столбцов изображения. Для определения функции преобразования  $f$ , растягивающей динамический диапазон изображения, необходимо знать только гистограмму входного изображения  $H_{in}[i]$ . В каждом разряде гистограммы  $H_{in}[i]$  хранится количество пикселей входного изображения, имеющих значение интенсивности  $z_i$ . Первое пороговое значение интенсивности  $t_1$  находится путем увеличения индекса  $i$  на гистограмме входного изображения до тех пор, пока суммарное количество пикселей с меньшими значениями интенсивности не превысит порог  $q_1$ . Всем пикселям входного изображения со значениями интенсивности  $z_k < t_1 - 1$  в результате преобразования на выходном изображении будут присвоены значения  $z_1$ . Пороговое значение  $t_1$  можно определить с помощью следующего выражения:

$$\sum_{i=1}^{t_1-1} H_{in}[i] \leq q_1 < \sum_{i=1}^{t_1} H_{in}[i]$$

Это выражение означает, что  $t_1$  является таким наименьшим значением интенсивности, для которого на гистограмме имеется не более  $q_1$  пикселей с меньшими значениями. Итерационные вычисления позволяют определить  $k$ -е пороговое значение  $t_k$ .

$$\sum_{i=1}^{t_k-1} H_{in}[i] \leq (q_1 + q_2 + \dots + q_k) < \sum_{i=1}^{t_k} H_{in}[i]$$

На практике функция преобразования  $f$  обычно задается в виде *таблицы преобразования* (lookup table, LUT). Ее легко сформировать с помощью приведенных выше выражений. Вычисленные значения  $t_k$  помещаются (возможно, повторно) в массив  $T[i]$  до тех пор, пока выполняется неравенство. Таким образом можно получить функцию  $z_{out} = f(z_{in}) = T[z_{in}]$ .

---

### Упражнение 5.1

Входное изображение содержит 200 пикселей и имеет следующую гистограмму:  $H_{in} = [0, 0, 20, 30, 5, 5, 40, 40, 30, 20, 10, 0, 0, 0, 0, 0]$ . а) Каким будет значение интенсивности  $f(8)$  выходного изображения, если входное изображение обрабатывается с использованием формулы для выравнивания гистограммы (с учетом 15 полутоновых уровней)? б) Ответьте на вопрос а) для  $f(11)$ . в) Запишите таблицу преобразования  $T[i]$ , задающую функцию  $f$  для преобразования всех значений входного изображения.

---

### Упражнение 5.2. Алгоритм выравнивания гистограммы

Запишите алгоритм выравнивания гистограммы на псевдокоде. Обязательно убедитесь в корректности определения всех используемых переменных и структур данных.

---

### Упражнение 5.3. Программа для выравнивания гистограммы

а) Пользуясь алгоритмом на псевдокоде из предыдущего упражнения, напишите и протестируйте программу для выравнивания гистограммы. б) Оцените качество обработки различных изображений.

Диапазон допустимых значений интенсивности выходного изображения часто бывает больше, чем у входного. В таком случае невозможно построить функцию  $f$  для отображения полутоновых уровней входного изображения на весь диапазон уровней выходного изображения. Если требуется обеспечить примерно равномерную гистограмму выходного изображения, то для отображения входного значения  $z_{in}$  в некоторую окрестность значения  $T[z_{in}]$  можно использовать генератор случайных чисел. Предположим, что описанная процедура применяется для отображения  $2q$  пикселей со значениями  $g$  на выходной уровень  $g_1$  и 0 пикселей на уровень  $g_1 + 1$ . Тогда можно имитировать подбрасывание монеты, чтобы входной пиксел со значением  $g$  с вероятностью 50 процентов преобразовывался в выходной пиксел со значением  $g_1$  или  $g_1 + 1$ .

## 5.3 УДАЛЕНИЕ МАЛЫХ ОБЛАСТЕЙ ИЗОБРАЖЕНИЯ

Часто бывает полезно удалить малые области из изображения. Малая область может образоваться из-за шума или может представлять низкоуровневую

деталь, которую не надо учитывать при формировании описания изображения. Малые области можно удалять посредством изменения отдельных пикселей или путем удаления компонент после выделения связанных компонент на изображении.

---

## Упражнение 5.4

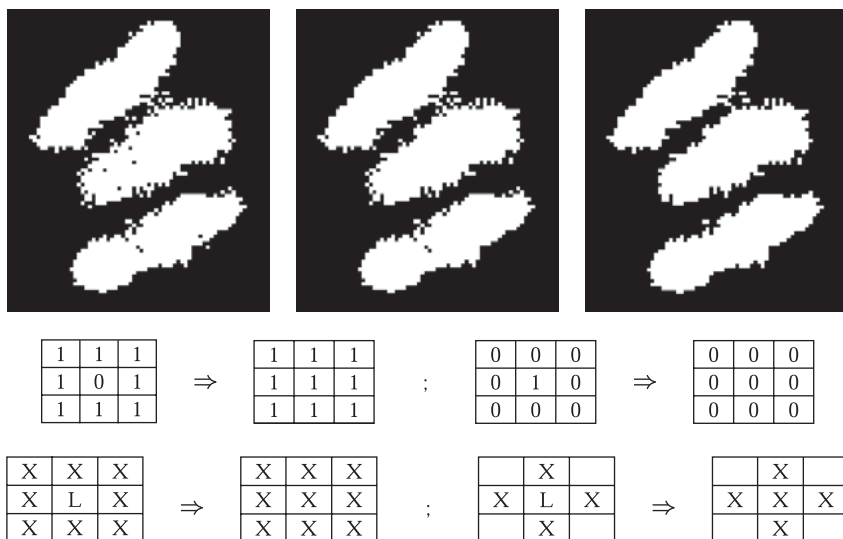
Приведите несколько аргументов в пользу и против применения *рандомизированной* функции  $f$  для того, чтобы сделать выходную гистограмму более равномерной.

### 5.3.1 Удаление шума «соль и перец»

Методы для удаления одиночных аномальных пикселей, лежащих внутри в целом однородных областей, кратко были рассмотрены во вводной главе, а затем более подробно в гл. 3. Одиночные темные пиксели, лежащие внутри светлых областей, или одиночные светлые пиксели внутри темных областей образуют шум «соль и перец» (salt-and-pepper noise). Название отражает очевидную бытовую аналогию. Шум «соль и перец» часто возникает в результате пороговой бинаризации изображения. «Соль» соответствует таким пикселям из темных областей, которые по каким-либо причинам преодолели порог, установленный для светлых пикселей. «Перец», наоборот, соответствует тем пикселям из светлых областей, значения которых оказались меньше порога бинаризации. Шум «соль и перец» может возникать из-за шума процесса аналого-цифрового преобразования во фреймграббере или вследствие ошибок классификации, вызванных неоднородностями на поверхностях объектов или изменением освещения. В некоторых случаях эти изолированные пиксели не являются классификационными ошибками, а соответствуют небольшим контрастным деталям на фоне более крупных объектов (например, пуговица на рубашке, поляна в лесу и т. п.). При формировании описаний изображений может потребоваться игнорировать такие мелкие детали.

На рис. 5.5 показано удаление шума «соль и перец» из бинарного изображения бактерий. Операции, применяемые к входному изображению, представлены с помощью *масок*, показанных в нижней части рисунка. Если окрестность входного изображения совпадает с маской, показанной слева, то эта окрестность заменяется на окрестность, заданную правой маской. В описанном методе применяются всего две маски.

Если входное изображение является маркированным изображением, сформированным с использованием набора порогов или каких-либо процедур классификации, то для удаления шума можно применять маску обобщенного вида. Как показано в нижнем ряду на рис. 5.5, любому пикселу с меткой  $L$ , который является изолированным по отношению к своим однородно маркированным 8-связным соседям, присваивается метка  $X$ . Метка  $L$  может быть любой из  $k$  меток, используемых в изображении. На рисунке также показано, что



**Рис. 5.5.** (Сверху слева) бинарное изображение бактерий. (Сверху в центре) удаление шума «соль и перец» с использованием 8-связности. (Сверху справа) удаление шума «соль и перец» с использованием 4-связности. Средний ряд: шаблоны, показывающие способы очистки окрестностей пикселей бинарного изображения. Нижний ряд: шаблоны, описывающие удаление изолированного пиксела на маркированном входном изображении общего вида. (Снизу слева) 8-связное решение. (Снизу справа) 4-связное решение. (С любезного разрешения Frank Dazzo.)

для принятия решения можно использовать 8- или 4-связность. В случае 4-связности при принятии решения не учитываются четыре угловых пиксела маски. В гл. 3 было показано, что при обработке с использованием различных окрестностей могут получаться разные выходные изображения. Это заметно и в рассматриваемом примере с изображением бактерий. Представленные на рис. 5.5 результаты обработки с использованием 8- или 4-связности имеют ряд различий.

### 5.3.2 Удаление малых компонент

В гл. 3 обсуждались способы выделения связанных компонент на бинарном изображении и рассматривалось большое количество признаков, которые можно вычислять по множеству пикселей, составляющих одну связную компоненту. Описание, сформированное для изображения, может содержать множество компонент, каждая из которых представляет область, выделенную на фоне изображения, и множество признаков, вычисленных для каждой компоненты. С учетом вычисленных признаков алгоритм может удалить из этого описания любую компоненту. Например, могут быть удалены компоненты, содержащие небольшое количество пикселей, или очень тонкие компоненты. Подобная





**Рис. 5.6.** (Слева) изображение бактерии, полученное в результате удаления шума «соль и перец» с использованием 4-связности из изображения, показанного на рис. 5.5. (Справа) изображение после удаления малых компонент. (С любезного разрешения Frank Dazzo.)

обработка могла бы привести к удалению ряда шумовых областей вблизи границы бактерии.

После удаления малых областей из описания изображения может потребоваться (или нет) построить соответствующее выходное изображение. Если выходное изображение необходимо, то информацию об удаленных компонентах надо сохранить, чтобы можно было правильно преобразовать пиксели входного изображения, принадлежащие областям удаленных компонент. На рис. 5.6 показаны изображения бактерии после удаления шума «соль и перец» и компонент, площадь которых не превышает 12 пикселей.

## 5.4 СГЛАЖИВАНИЕ ИЗОБРАЖЕНИЯ

Часто изображение содержит некоторую основную идеальную структуру, которую требуется обнаружить и описать, а также какой-либо случайный шум или ложные признаки, которые желательно удалить. Например, для построения простой модели такого изображения можно считать, что пиксели области изображения, соответствующей образу однородного объекта, имеют значения  $g_r + N(0, \sigma)$ , где  $g_r$  — это некоторое ожидаемое значение интенсивности, которое было бы получено на идеальном изображении, а  $N(0, \sigma)$  — гауссов шум, описываемый нормальным распределением с математическим ожиданием 0 и среднеквадратическим отклонением  $\sigma$ . На рис. 5.7 (сверху слева) показано идеальное изображение шахматной доски с однородными областями. Для формирования зашумленного изображения, показанного в центре, к идеальному изображению был добавлен гауссов шум. При этом шумовые значения были ограничены диапазоном  $[0, 255]$ . На рис. 5.7 (сверху справа) представлена диаграмма значений интенсивности пикселей, расположенных вдоль одной (горизонтальной) строки изображения.

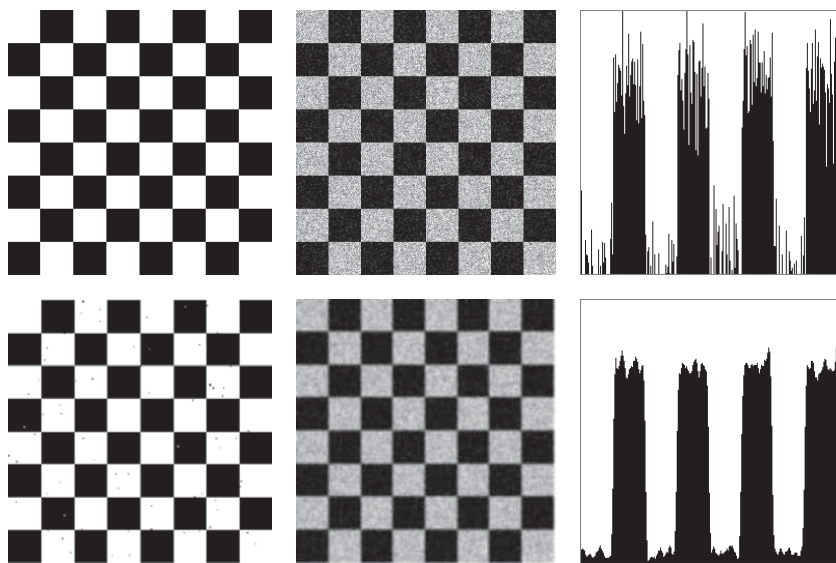


Шум, значения которого случайным образом отличаются в большую и меньшую сторону от номинального значения интенсивности пикселей области, можно уменьшить путем усреднения значений интенсивности в окрестности пикселей.

$$OutputImage[r, c] = \text{среднее значение некоторой окрестности } InputImage[r, c] \quad (5.1)$$

$$Out[r, c] = \left( \sum_{i=-2}^{+2} \sum_{j=-2}^{+2} In[r + i, c + j] \right) / 25 \quad (5.2)$$

Уравнение (5.2) определяет сглаживающий фильтр, который присваивает каждому пикселу выходного изображения среднее значение интенсивности



**Рис. 5.7.** (Сверху слева) идеальное изображение шахматной доски, на котором значения пикселей в черных клетках равны 0, а значения пикселей в белых клетках равны 255. (Сверху в центре) изображение с добавленным гауссовым шумом со среднеквадратическим отклонением 30. (Сверху справа) диаграмма значений интенсивности пикселей из 100-й сверху горизонтальной строки зашумленного изображения. (Снизу слева) бинарное изображение, полученное в результате пороговой бинаризации центрального верхнего изображения. В качестве порога бинаризации было выбрано значение из впадины гистограммы изображения. На бинарном изображении присутствует перечный шум. (Снизу в центре) результат сглаживания зашумленного изображения путем усреднения значений интенсивности в окрестности  $5 \times 5$  с центром в каждом пикселе. (Снизу справа) диаграмма значений интенсивности пикселей из 100-й сверху горизонтальной строки сглаженного зашумленного изображения.

по значениям 25 пикселей в окрестности  $5 \times 5$  соответствующего пиксела входного изображения. В результате фильтр формирует сглаженное выходное изображение. На рис. 5.7 представлен результат применения этого фильтра к изображению шахматной доски. Строка изображения, представленная на диаграмме в нижнем ряду на рис. 5.7, является более гладкой по сравнению со строкой входного изображения, показанной на диаграмме в верхнем ряду. Сглаженная строка в действительности получена усреднением значений пикселей не из одной, а из пяти строк входного изображения. Также отметим, что сглаженное изображение выглядит более чистым, но менее резким, чем исходное.

**43 Определение.** Сглаживание изображения с использованием прямоугольной окрестности пикселей с равновзвешенными значениями называется применением **усредняющего фильтра**.

Вместо присвоения всем пикселям фильтра равных весов предпочтительнее выбрать веса такими, чтобы они уменьшались с увеличением расстояния входных пикселей от центрального пиксела  $\mathbf{I}[\mathbf{x}_c, \mathbf{y}_c]$ . Подобные весовые коэффициенты имеет фильтр Гаусса, который, возможно, является наиболее широко используемым фильтром. Важные свойства этого фильтра более подробно обсуждаются в разделе 5.7.

**44 Определение.** При применении **фильтра Гаусса** значение пиксела  $\mathbf{I}[\mathbf{x}, \mathbf{y}]$  взвешивается с использованием весовой функции

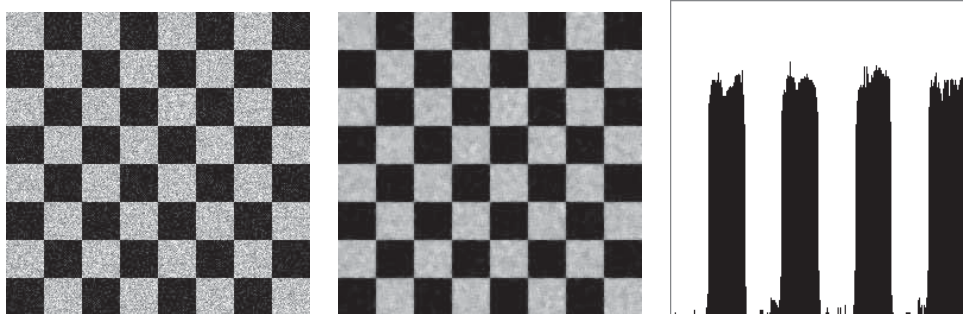
$$g(x, y) = \frac{1}{\sqrt{2\pi} \sigma} e^{-\frac{d^2}{2\sigma^2}}$$

где  $d = \sqrt{(x - x_c)^2 + (y - y_c)^2}$  представляет расстояние пиксела  $\mathbf{I}[\mathbf{x}, \mathbf{y}]$  от центрального пиксела окрестности  $\mathbf{I}[\mathbf{x}_c, \mathbf{y}_c]$  на выходном изображении, для формирования которого применяется фильтр.

Далее в этой главе будут более подробно рассмотрены как теоретические вопросы, так и методы сглаживания. Этот же подход будет использован применительно к задаче обнаружения краев. Перед рассмотрением этих вопросов сначала будет представлен полезный и интуитивно понятный *медианный фильтр*.

## 5.5 МЕДИАННАЯ ФИЛЬТРАЦИЯ

Осреднение приводит к получению более точных оценочных значений  $\mathbf{I}[\mathbf{x}, \mathbf{y}]$  при вычислении средних значений по однородным окрестностям, шум в которых описывается распределением с нулевым математическим ожиданием. Когда обрабатываемая окрестность попадает на границу между такими областями, то использование значений из двух областей приводит к размытию границы. Другим распространенным фильтром является *медианный фильтр*,



**Рис. 5.8.** (Слева) зашумленное изображение шахматной доски. (В центре) результат обработки посредством присвоения каждому выходному пикселу значения медианы окрестности  $5 \times 5$  пикселей с центром в соответствующем пикселе входного изображения. (Справа) значения пикселей из 100-й сверху горизонтальной строки изображения. Сравните с диаграммой на рис. 5.7.

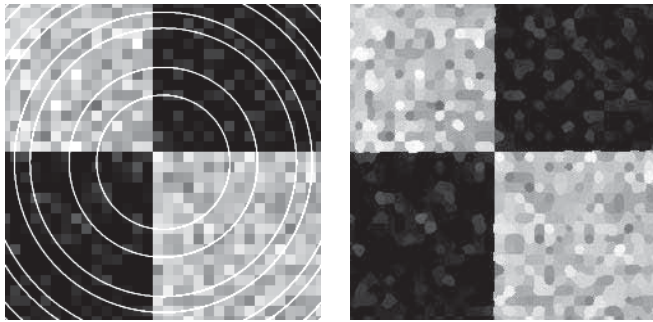
который заменяет значение пиксела на *медиану* значений из окрестности данного пиксела.

**45 Определение.** Обозначим  $\mathbf{A}[i]_{i=0 \dots (n-1)}$  отсортированный массив  $n$  действительных чисел. **Медиана** множества чисел, хранящихся в массиве  $\mathbf{A}$ , равна  $\mathbf{A}[(n-1)/2]$

Иногда значение медианы по-разному вычисляется для массивов с четным и нечетным количеством элементов. При нечетном значении  $n$  средний элемент массива определяется единственным образом, как было сделано выше. Если  $n$  имеет четное значение, то медиану можно определять по-разному: выбрать  $\mathbf{A}[n/2]$ , или  $\mathbf{A}[n/2 - 1]$ , или вычислить среднее двух этих значений. Хотя в определении медианы рассматривается отсортированный массив, на практике обычно не требуется выполнять полную сортировку множества чисел, медиану которого необходимо вычислить. Известный алгоритм быстрой сортировки<sup>1</sup> можно изменить так, чтобы рекурсивный вызов в нем выполнялся только для той части массива  $\mathbf{A}$ , в которой содержится  $(n+1)/2$ -й элемент. Как только центр разбиения (pivot) попадет на элемент массива, положение которого соответствует медиане всего исходного массива, то искомое значение медианы будет определено.

Рисунок 5.8 демонстрирует, что медианный фильтр способен хорошо сглаживать зашумленные области, сохраняя структуру границ между ними. Когда обрабатываемый пиксел выбирается из некоторой белой клетки близко к ее краю, вероятно, большинство пикселей в его окрестности тоже окажутся

<sup>1</sup> Исходный массив разбивается на две половины так, чтобы любой элемент первой половины был упорядочен относительно любого элемента второй половины. Затем эта процедура рекурсивно применяется к каждой половине. — *Прим. ред.*



**Рис. 5.9.** (Слева) входное изображение содержит гауссов шум и ложные детали в виде ярких окружностей, которые были добавлены к четырем ранее однородным областям. (Справа) результат применения медианного фильтра с маской размера  $7 \times 7$ .

белыми зашумленными пикселями. Если это так, то для определения выходного значения обрабатываемого пикселя соседние пиксели из черных клеток использоваться не будут. Аналогично, при вычислении выходного пикселя вблизи границы со стороны черной клетки высока вероятность того, что большинство пикселей в окрестности обрабатываемого пикселя будут черными зашумленными пикселями. Значит, для вычисления выходного значения в этом случае не будут использоваться белые пиксели. Таким образом, в отличие от усредняющего сглаживания, медианный фильтр стремится сохранять структуру краев, в то же время сглаживая однородные области. Медианный фильтр позволяет удалять шум «соль и перец» и большинство других малых деталей, которые можно рассматривать как присутствие на идеальном изображении произвольных шумовых значений. Рис. 5.9 демонстрирует удаление из изображения малых структурированных деталей с одновременным уменьшением изменчивости в пределах однородных областей и сохранением границ между ними.

Так как для вычисления медианы требуется выполнять сортировку (хотя бы частичную), то эта операция занимает больше времени по сравнению с вычислением среднего значения интенсивности в пиксельной окрестности. Кроме того, медианную фильтрацию сложнее реализовать на аппаратном уровне, что может быть необходимо для обработки в реальном времени (например, для организации конвейерной видеобработки). Однако во многих задачах анализа изображений достоинства медианной фильтрации оправдывают увеличение вычислительных затрат.

---

### Упражнение 5.5. Модификация алгоритма быстрой сортировки (1)

(а) В одном из учебников по структурам данных и алгоритмам найдите представление классического алгоритма быстрой сортировки на псевдокоде.

Измените алгоритм таким образом, чтобы он максимально быстро возвращал значение медианы. (б) Какова вычислительная эффективность поиска медианы по сравнению с полной сортировкой? (в) Реализуйте разработанный алгоритм на любом языке программирования. Обработайте с помощью полученной программы несколько различных изображений.

---

### Упражнение 5.6. Модификация алгоритма быстрой сортировки (2)

Рассмотрите возможность применения алгоритма быстрой сортировки для обнаружения ступенчатых изменений функции интенсивности (например, таких, как ступенчатые перепады между черными и белыми клетками на изображении шахматной доски). Предположим, что медиана окрестности пиксела  $\mathbf{I}[\mathbf{r}, \mathbf{c}]$  только что была обнаружена, так как на очередной итерации центр разбиения попал в позицию массива  $\mathbf{A}[n/2]$ . Предложите способ обработки оставшейся части массива для принятия решения о том, находится или нет пиксел  $[\mathbf{r}, \mathbf{c}]$  на границе областей различной яркости.

## 5.5.1 Вычисление выходного изображения по данным входного изображения

Теперь, после рассмотрения нескольких примеров, демонстрирующих различные типы операций улучшения изображений, важно рассмотреть способы реализации этих операций. Приведенный ниже обобщенный алгоритм предоставляет различные возможности управления фильтрацией входного изображения для получения улучшенного выходного изображения.

Алгоритм 5.1 имеет простую последовательную структуру. Этот алгоритм вычисляет пиксели выходного изображения  $\mathbf{G}$  в растровом порядке. При вычислении значения каждого выходного пиксела  $\mathbf{G}[\mathbf{r}, \mathbf{c}]$  используется окрестность пикселей входного изображения с центром  $\mathbf{F}[\mathbf{r}, \mathbf{c}]$ . Очевидно, что значения пикселей выходного изображения  $\mathbf{G}$  можно вычислять в произвольном порядке, а не только в порядке последовательного просмотра строк и столбцов. В принципе, значения всех пикселей можно вычислять параллельно. Это объясняется тем, что в процессе вычислений в окрестностях пикселей входного изображения оно не изменяется.

Процедура `compute_using_neighbors`, которая выполняет вычисления в окрестности пикселей, может быть реализована для выполнения усредняющей (boxcar) или медианной фильтрации. Для выполнения усредняющей фильтрации в процедуре необходимо только вычислять сумму элементов  $w \times h$  пикселей из окрестности  $\mathbf{F}[\mathbf{r}, \mathbf{c}]$ , а затем делить значение суммы на количество пикселей окрестности  $w \times h$ . Для выполнения медианной фильтрации в процедуре можно было бы предусмотреть копирование  $w \times h$  значений пикселей в локальный массив  $\mathbf{A}$  и вычислять медиану этих значений посредством частичной сортировки массива.

**Вычисление значения пиксела  $G[r, c]$  выходного изображения по окрестности пиксела  $F[r, c]$  входного изображения.**

$F[r, c]$  — входное изображение, содержащее **MaxRow** строк и **MaxCol** столбцов.

Изображение **F** алгоритмом не изменяется.

$G[r, c]$  — выходное изображение, содержащее **MaxRow** строк и **MaxCol** столбцов.

Граница изображения **G** определяется пикселями, окрестности которых не помещаются целиком на изображении **G**.

$w$  и  $h$  — ширина и высота пиксельной окрестности (в пикселах).

```

procedure enhance_image(F,G,w,h);
{
  for r := 0 to MaxRow - 1
    for c := 0 to MaxCol - 1
      {
        if [r, c] является граничным пикселом then G[r, c] := F[r, c];
        else G[r, c] := compute_using_neighbors (F, r, c, w, h);
      } ;
}
procedure compute_using_neighbors (IN, r, c, w, h)
{
  использование всех пикселов в пределах  $w/2$  и  $h/2$  от пиксела
  IN[r, c],
  вычисление возвращаемого значения, представляющего IN[r, c]
}

```

**Алгоритм 5.1.** Вычисление значения пиксела выходного изображения  $G[r, c]$  на основе значений окрестности пиксела входного изображения  $F[r, c]$ .

### Упражнение 5.7

Реализуйте Алгоритм 5.1 на любом языке программирования. Предусмотрите в программе выполнение операций усредняющей и медианной фильтрации. Протестируйте их на нескольких изображениях, например, таких, как на рис. 5.9.

Фильтрация может быть организована таким образом, чтобы одновременно в оперативной памяти хранились только  $h$  строк изображения. Значения выходных пикселов  $G[r, c]$  вычисляются только для средней строки  $r$ . Затем вводится новая строка, которая заменяет наиболее старую строку в памяти, и вычисляются значения  $G[r, c]$  для следующей строки выходного изображения.

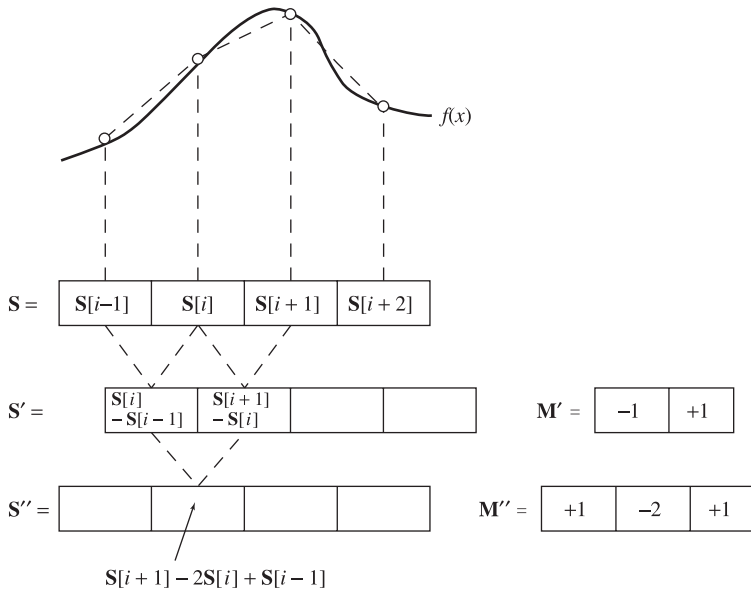
Этот процесс повторяется до тех пор, пока не будут вычислены все строки выходного изображения. Много лет назад, когда компьютеры оснащались оперативной памятью небольшого объема, а в качестве основной памяти для хранения изображений использовались диски, многие алгоритмы обрабатывали изображения частями по несколько строк. Сегодня такая структура алгоритмов обработки остается интересной благодаря возможности использования в специальных устройствах обработки изображений с конвейерной архитектурой.

## 5.6 ОБНАРУЖЕНИЕ КРАЕВ С ПОМОЩЬЮ ДИФФЕРЕНЦИАЛЬНЫХ МАСОК

Точки высокого контраста можно обнаружить на изображении посредством вычисления разностей значений интенсивности в локальных областях изображения. Обычно такие точки образуют границы между различными объектами или частями сцены. В данном разделе описывается, как такие точки можно находить с помощью шаблонов пиксельных окрестностей или *масок*. Сначала рассматривается обработка одномерных сигналов: это позволяет дать наглядную интерпретацию ряда понятий, а также ввести используемые далее математические обозначения. Кроме того, обработка одномерных сигналов имеет важное практическое значение. В качестве одномерных сигналов можно рассматривать строки или столбцы двумерного изображения. В конце раздела описываются более общие двумерные задачи.

### 5.6.1 Дифференцирование одномерных сигналов

На рис. 5.10 поясняется использование маски для вычисления приближенных значений производных одномерного сигнала. Будем считать, что сигнал  $\mathbf{S}$  задан в виде последовательности выборочных значений некоторой функции  $f$ . Тогда  $f'(x_i) \approx (f(x_i) - f(x_{i-1})) / (x_i - x_{i-1})$ . Предположим, что для выборки дискретных значений область определения функции  $f(x)$  разбивается на интервалы шириной  $\Delta x = 1$ . Тогда производную функции  $f(x)$  можно приближенно вычислить путем применения маски  $\mathbf{M}' = [-1, 1]$  к выборочным значениям  $\mathbf{S}$ , как показано на рис. 5.10. В результате можно получить сигнал  $\mathbf{S}'$ . Как показано на рисунке, удобно считать, что значения  $\mathbf{S}'$  расположены между значениями сигнала  $\mathbf{S}$ . Элементы  $\mathbf{S}'[i]$  с большими абсолютными значениями соответствуют областям, в которых значение сигнала быстро изменяется или сигнал имеет *высокий контраст*. Сигнал  $\mathbf{S}'$  тоже можно продифференцировать с использованием маски  $\mathbf{M}'$ , чтобы получить выходной сигнал  $\mathbf{S}''$ , соответствующий второй производной исходной функции  $f$ . Приближенные значения второй производной можно вычислить, применяя маску  $\mathbf{M}''$  к исходной последовательности выборочных значений  $\mathbf{S}$ . Получение этого важного результата,



**Рис. 5.10.** (Слева) первый ( $S'$ ) и второй ( $S''$ ) разностные сигналы являются приближенными представлениями первой и второй производной сигнала  $S$ . (Справа) маски  $M'$  и  $M''$  представляют операции вычисления производных.

который также демонстрируется на рис. 5.10, выполняется в приведенных далее выражениях.

$$S'[i] = -S[i-1] + S[i] \quad (5.3)$$

$$\text{маска } M' = [-1, +1] \quad (5.4)$$

$$S''[i] = -S'[i] + S'[i+1] \quad (5.5)$$

$$= -(S[i] - S[i-1]) + (S[i+1] - S[i]) \quad (5.6)$$

$$= S[i-1] - 2S[i] + S[i+1] \quad (5.7)$$

$$\text{маска } M'' = [1, -2, 1] \quad (5.8)$$

Если требуется обнаружить только точки высокого контраста, то после применения маски к элементу сигнала  $S[i]$  обычно используется абсолютное значение результата. При использовании абсолютных значений в качестве маски первой производной можно выбрать любую из двух масок  $M' = [-1, +1]$  или  $[+1, -1]$ , а в качестве маски второй производной — маску  $M'' = [+1, -2, +1]$  или  $[-1, +2, -1]$ . Как будет показано далее, аналогичным образом можно поступать и при обработке двумерных изображений. Всякий раз, когда будут рассматриваться только абсолютные значения (модули) величин, будем считать эти маски одинаковыми. Если же знак производной (направление изменения сигнала) игнорировать нельзя, то будем считать эти маски различными.



маска  $M = [-1, 0, 1]$

$S_1$			12	12	12	12	12	24	24	24	24	24
$S_1$	$\otimes$	$M$	0	0	0	0	12	12	0	0	0	0

(a)  $S_1$  представляет ступенчатый край, направленный вверх

$S_2$			24	24	24	24	24	12	12	12	12	12
$S_2$	$\otimes$	$M$	0	0	0	0	-12	-12	0	0	0	0

(b)  $S_2$  представляет ступенчатый край, направленный вниз

$S_3$			12	12	12	12	15	18	21	24	24	24
$S_3$	$\otimes$	$M$	0	0	0	3	6	6	6	3	0	0

(c)  $S_3$  представляет наклонный край, направленный вверх

$S_4$			12	12	12	12	24	12	12	12	12	12
$S_4$	$\otimes$	$M$	0	0	0	12	0	-12	0	0	0	0

(d)  $S_4$  представляет яркостной импульс или линию

**Рис. 5.11.** Кросс-корреляция четырех сигналов различного вида с маской первой производной  $M = [-1, 0, 1]$ ; для обнаружения края. (a) Ступенчатый край, направленный вверх. (b) Ступенчатый край, направленный вниз. (c) Наклонный край, направленный вверх. (d) Яркостной импульс. Сумма элементов  $M$  равна 0, поэтому выходные значения будут равны нулю в областях постоянных входных значений.

На рис. 5.11. показано применение еще одной часто используемой маски первой производной. Эта маска содержит 3 элемента и центрируется в точке сигнала  $S[i]$ , чтобы вычислить разность значений в элементах, смежных с центральным. Так как  $\Delta x = 2$ , то отклик маски будет превышать истинное значение производной, поэтому вычисленное значение надо разделить на 2. На идеальный ступенчатый край эта маска дает отклик в двух дискретных элементах (рис. 5.11 (a)–(b)). На рис. 5.12 показан отклик маски второй производной на сигналы нескольких видов. Как видно на рис. 5.12, высокий контраст сигнала определяется точками *перехода через ноль*<sup>1</sup> Эти точки определяют местоположение и усиливают изменение между двумя последовательными значениями сигнала. Совместное рассмотрение сигналов, соответствующих первой и второй производной, позволяет обнаружить много локальных особенностей исходного сигнала. На рис. 5.13 поясняется, как можно по аналогии с дифференцированием сигнала выполнить его сглаживание. Ниже для сравнения перечислены основные свойства сглаживающих масок и масок производных.

<sup>1</sup> Также называются точками пересечения нуля или точками смены знака, англ. zero-crossing. — Прим. перев.

маска  $M = [-1, 2, -1]$

$S_1$			12	12	12	12	12	24	24	24	24	24
$S_1$	$\otimes$	$M$	0	0	0	0	212	12	0	0	0	0

(a)  $S_1$  представляет ступенчатый край, направленный вверх

$S_2$			24	24	24	24	24	12	12	12	12	12
$S_2$	$\otimes$	$M$	0	0	0	0	12	-12	0	0	0	0

(b)  $S_2$  представляет ступенчатый край, направленный вниз

$S_3$			12	12	12	12	15	18	21	24	24	24
$S_3$	$\otimes$	$M$	0	0	0	-3	0	0	0	3	0	0

(c)  $S_3$  представляет наклонный край, направленный вверх

$S_4$			12	12	12	12	24	12	12	12	12	12
$S_4$	$\otimes$	$M$	0	0	0	-12	24	-12	0	0	0	0

(d)  $S_4$  представляет яркостной импульс или линию

**Рис. 5.12.** Кросс-корреляция четырех сигналов различного вида с маской второй производной  $M = [-1, 2, -1]$  для обнаружения края. (a) Ступенчатый край, направленный вверх. (b) Ступенчатый край, направленный вниз. (c) Наклонный край, направленный вверх. (d) Яркостной импульс. Так как сумма элементов  $M$  равна нулю, то отклик маски на областях постоянных значений тоже будет равен нулю. Обратите внимание, что в выходном сигнале в тех положениях, в которых во входном сигнале соединяются фрагменты различных направлений, появляются точки *перехода через ноль*.

### Некоторые свойства масок для вычисления производных:

- Для получения высокого отклика в областях сигнала с высоким контрастом элементы масок производных имеют разные знаки.
- Сумма элементов масок производных равна 0, поэтому в областях сигнала с постоянными значениями маски дают нулевой отклик.
- Точкам высокого контраста соответствуют большие абсолютные значения отклика масок первой производной.
- Среди значений отклика масок второй производной точкам высокого контраста соответствуют точки перехода через ноль.

### Для сравнения перечислим ряд свойств сглаживающих масок:

- Элементы сглаживающих масок имеют положительные значения и их сумма равна единице. Поэтому отклик масок на областях постоянных значений совпадает с входными значениями.

**Маска усредняющего сглаживания  $M = [1/3, 1/3, 1/3]$**

$S_1$			12	12	12	12	12	24	24	24	24	24
$S_1$	$\otimes$	$M$	12	12	12	12	16	20	24	24	24	24

(a)  $S_1$  представляет ступенчатый край, направленный вверх

$S_4$			12	12	12	12	24	12	12	12	12	12
$S_4$	$\otimes$	$M$	12	12	12	16	16	16	12	12	12	12

(d)  $S_4$  представляет яркостной импульс или линию

**Маска гауссовского сглаживания  $M = [1/4, 1/2, 1/4]$**

$S_1$			12	12	12	12	12	24	24	24	24	24
$S_1$	$\otimes$	$M$	12	12	12	12	15	21	24	24	24	24

(a)  $S_1$  представляет ступенчатый край, направленный вверх

$S_4$			12	12	12	12	24	12	12	12	12	12
$S_4$	$\otimes$	$M$	12	12	12	15	18	15	12	12	12	12

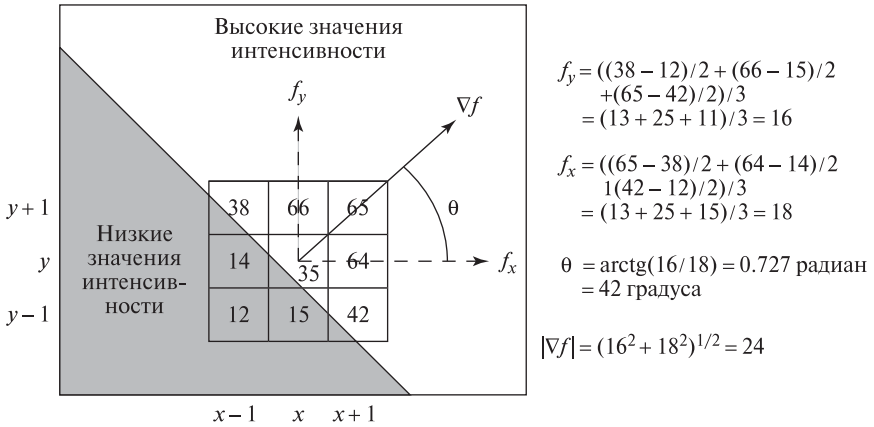
(d)  $S_4$  представляет яркостной импульс или линию

**Рис. 5.13.** (Две верхние строки) сглаживание ступенчатого и импульсного сигналов с использованием усредняющей маски  $[1/3, 1/3, 1/3]$ . (Две нижние строки) сглаживание ступеньки и импульса ступенчатого и импульсного сигналов с использованием гауссовой маски  $[1/4, 1/2, 1/4]$ .

- Степень сглаживания и снижения шума пропорциональна размеру маски.
- Ступенчатые края размываются пропорционально размеру маски.

## 5.6.2 Дифференциальные операторы для двумерных изображений

Направление контраста двумерной функции интенсивности  $f(x, y)$  может быть произвольным. Из математического анализа известно, что максимальные изменения значений функции происходят вдоль направления ее градиента. Вектор градиента двумерной функции имеет координаты  $[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}]$ . На рис. 5.14 понятие градиента поясняется применительно к его дискретной аппроксимации на цифровом изображении. В качестве оценки контраста изображения в пикселе  $\mathbf{I}[\mathbf{x}, \mathbf{y}]$  вдоль направления оси  $x$  можно взять разность  $(I[x + 1, y] - I[x - 1, y])/2$ . Это значение представляет изменение интенсивности между левым и правым соседями пиксела  $[\mathbf{x}, \mathbf{y}]$ , которые лежат на расстоянии  $\Delta x = 2$  пиксела. Для окрестности, показанной на рис. 5.14, контраст в направлении оси  $x$  будет равен  $(64 - 14)/2 = 25$ . Так как в значениях пикселей может присутствовать шум, а также потому, что край может пересекать пиксельный



**Рис. 5.14.** Оценка величины и направления контраста изображения в точке  $\mathbf{I}[\mathbf{x}, \mathbf{y}]$  посредством вычисления модуля и направления градиента функции интенсивности  $f(x, y)$  с использованием дискретных значений из пиксельного массива изображения.

массив под любым углом, для вычисления более точной оценки контраста в направлении оси  $x$  можно усреднить три различные оценки контраста в окрестности пиксела  $[\mathbf{x}, \mathbf{y}]$ :

$$\begin{aligned} df/\partial x \equiv f_x \approx \frac{1}{3} & [(I[x+1, y] - I[x-1, y])/2 \\ & + (I[x+1, y-1] - I[x-1, y-1])/2 \\ & + (I[x+1, y+1] - I[x-1, y+1])/2] \end{aligned} \quad (5.9)$$

Это приближенное значение контраста изображения в направлении оси  $x$  вычисляется как среднее значение контраста изображения вдоль строки с координатой  $y$ , а также строк, расположенных выше и ниже строки  $y$ . Приближенное значение контраста изображения в направлении оси  $y$  можно вычислить аналогичным образом:

$$\begin{aligned} df/\partial y \equiv f_y \approx \frac{1}{3} & [(I[x, y+1] - I[x, y-1])/2 \\ & + (I[x-1, y+1] - I[x-1, y-1])/2 \\ & + (I[x+1, y+1] - I[x+1, y-1])/2] \end{aligned} \quad (5.10)$$

Для ускорения вычислений деление на 6 часто не выполняется, т. е. оценочные значения вычисляются в увеличенном масштабе. Маски двух этих операторов контраста, обозначенные как  $M_x$  и  $M_y$ , показаны в верхней части рис. 5.15. Применяя эти маски к 8-связной окрестности  $N_8[x, y]$  пиксела  $(\mathbf{x}, \mathbf{y})$  в соответствии с уравнениями (5.11)–(5.14), можно получить оценку градиента функции интенсивности. Эти маски служат определением *оператора Превитта* (Prewitt operator), названного так в честь доктора Джудит Превитт

$$\begin{array}{l}
 \text{Превитт: } M_x = \begin{array}{|c|c|c|} \hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline \end{array} ; \quad M_y = \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -1 & -1 \\ \hline \end{array} \\
 \\
 \text{Собель: } M_x = \begin{array}{|c|c|c|} \hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array} ; \quad M_y = \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -2 & -1 \\ \hline \end{array} \\
 \\
 \text{Робертс: } M_x = \begin{array}{|c|c|} \hline 0 & 1 \\ \hline -1 & 0 \\ \hline \end{array} ; \quad M_y = \begin{array}{|c|c|} \hline 1 & 0 \\ \hline 0 & -1 \\ \hline \end{array}
 \end{array}$$

**Рис. 5.15.** Маски, применяемые для оценки градиента функции интенсивности  $f(x, y)$ : (Сверху) маски Превитт (Prewitt). (В середине) маски Собеля (Sobel). (Внизу) маски Робертса (Roberts).

(Judith Prewitt), которая применила этот оператор для обнаружения границ на медицинских изображениях.

$$\frac{\partial f}{\partial x} \approx (1/6)(M_x \circ N_8[x, y]) \quad (5.11)$$

$$\frac{\partial f}{\partial y} \approx (1/6)(M_y \circ N_8[x, y]) \quad (5.12)$$

$$|\nabla f| \approx \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} \quad (5.13)$$

$$\theta \approx \arctg\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right) \quad (5.14)$$

Операция скалярного произведения  $M \circ N$  подробно рассматривается в следующем разделе. Для выполнения этой операции маска  $M$  накладывается на окрестность  $N$  изображения так, чтобы каждое значение интенсивности  $N_{ij}$  можно было умножить на весовой коэффициент  $M_{ij}$ , и затем вычисляется сумма этих произведений. В средней строке на рис. 5.15 приведены две другие маски аналогичного назначения — *маски Собеля*. Способ их получения и интерпретации аналогичен маскам Превитт, за исключением того, что оценка контраста по центральной строке и столбцу берется с удвоенным весом по сравнению с оценками, получаемыми по разные стороны от центральной точки.

*Маски Робертса* имеют размеры всего лишь  $2 \times 2$ . Они применяются в меньшей локальной окрестности и обработка с этими масками выполняется быстрее по сравнению с масками большего размера. Эти маски часто называются *крестообразным оператором Робертса*. Фактически они дают оценку градиента в центре окрестности из 4 пикселей, а не в центральном пикселе. Кроме того, система координат, в которой задан этот оператор, повернута на  $45^\circ$  относительно направления строк изображения.

Результаты применения оператора Робертса показаны на рис. 5.16. Входное изображение представлено вверху слева на рис. 5.16 (a), а выходные изображения, полученные с помощью незначительно измененных операторов Робертса, приведены на рис. 5.16 (b) и (c). Изображения (d) и (e) демонстрируют результаты вычисления обычных разностей значений интенсивности вдоль столбцов и строк изображения, соответственно. Изображение (f) получено путем объединения разностных изображений (d) и (e) с помощью операции ИЛИ (OR). С качественной точки зрения, эти результаты типичны для операторов, обрабатывающих малые пиксельные окрестности — на многих краях многие пикселы выделяются, но при этом также многие оказываются пропущенными. Например, отклики масок на текстурированной области травы и на верхней части гаража теряются из-за того, что они оказываются сравнимыми с откликами в области неба. Результаты, полученные с помощью оператора Робертса, можно сравнить с результатами применения комбинации простых одномерных масок, выполняющих обработку в направлении строк и столбцов изображения (рис. 5.16 (d-f)).

Часто при вычислении величины градиента стараются избежать вычисления квадратного корня.

Альтернативными оценочными выражениями являются  $\max\left(\left|\frac{\partial f}{\partial x}\right|, \left|\frac{\partial f}{\partial y}\right|\right)$ ,  $\left|\frac{\partial f}{\partial x}\right| + \left|\frac{\partial f}{\partial y}\right|$  и  $\left(\left(\frac{\partial f^2}{\partial x}\right) + \left(\frac{\partial f^2}{\partial y}\right)\right)/2$ . Сравнение рис. 5.16 (b) и (c, f) подтверждает допустимость отказа от операции извлечения квадратного корня. Но использование таких оценочных значений для интерпретации фактической величины или направления градиента требует особого внимания.

На рис. 5.17 (b) показаны результаты применения оператора Собеля с маской  $3 \times 3$  для вычисления среднеквадратических значений модуля градиента. Рис. 5.17 (c) содержит данные о направлении градиента. Маленькие квадраты на исходном изображении занимают области размером  $8 \times 8$  пикселов. Оператор Собеля выделяет многие, но не все, края<sup>1</sup> исходного изображения.

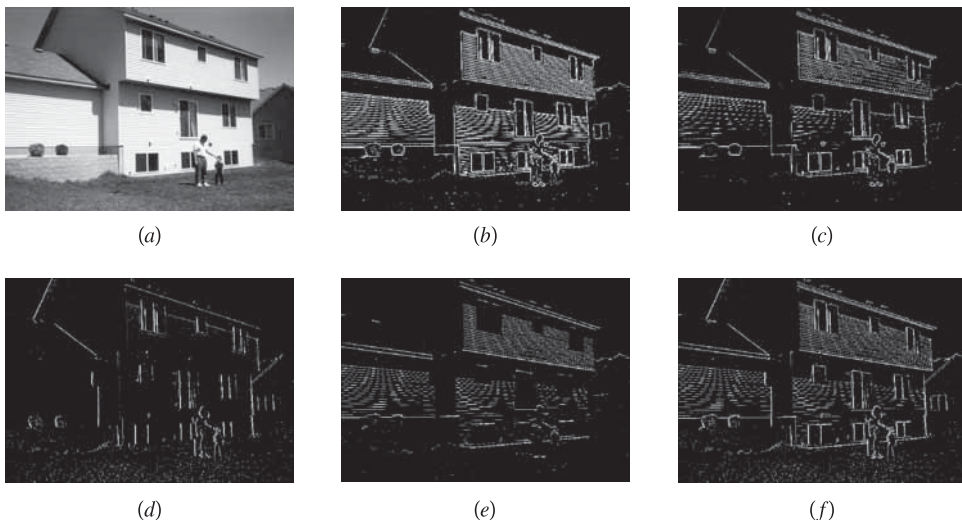
## Упражнение 5.8

Почему маски Собеля, по сравнению с масками Превитт, позволяют быстрее оценить истинное значение градиента?

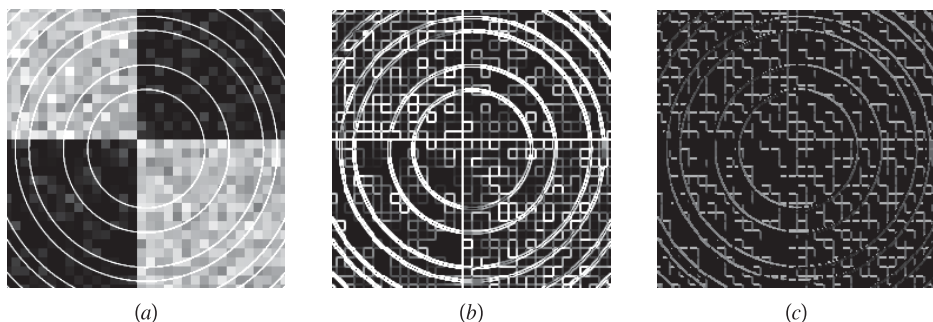
## Упражнение 5.9. Оптимальность масок Превитт.\*

Требуется доказать, что маски Превитт содержат весовые коэффициенты, соответствующие наилучшей аппроксимирующей плоскости в пиксельной окрестности  $3 \times 3$ . Предположим, что по 9 значениям интенсивности  $I[r+i, c+j]$  при  $i, j = -1, 0, 1$ , взятым из пиксельной окрестности  $3 \times 3$ , методом наименьших квадратов вычисляются коэффициенты аппроксимирующей плоскости

<sup>1</sup> Здесь было бы уместнее говорить о перепадах яркости. — *Прим. ред.*



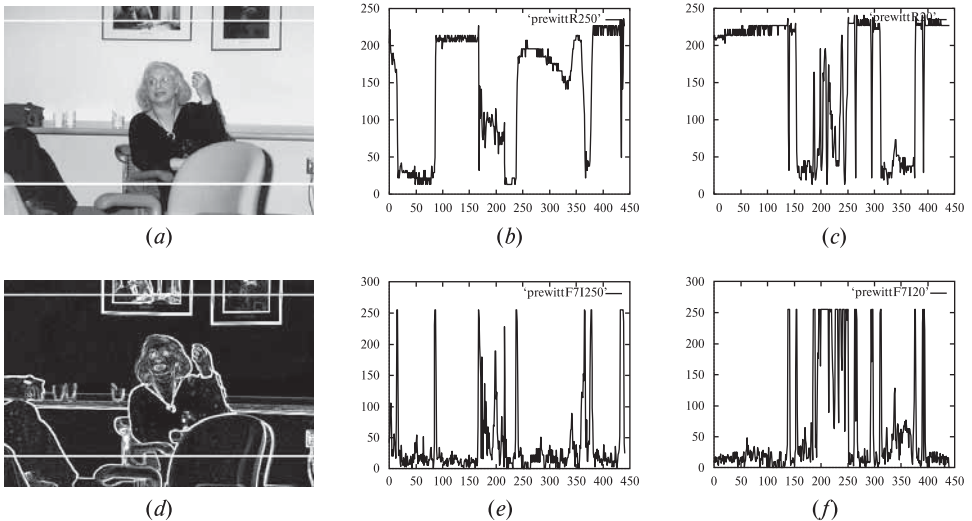
**Рис. 5.16.** (a) Исходное изображение. (b) 5 процентов наибольших значений сумм модулей откликов двух масок Робертса. (c) 5 процентов наибольших значений среднеквадратического значения откликов двух масок Робертса. (d) 2 процента наибольших абсолютных значений откликов маски  $[-1, +1]$ , примененной для выделения края в направлении оси  $y$ . (e) 3 процента наибольших абсолютных значений откликов маски  $[-1, +1]^t$  для выделения края в направлении оси  $x$ . (f) Логическое сложение изображений (d) и (e) с помощью операции ИЛИ. Между изображениями (b), (c) и (f) заметны небольшие отличия. (С любезного разрешения Ida Stockman.)



**Рис. 5.17.** (a) Зашумленное изображение, содержащее квадраты и окружности. (b) Среднеквадратичный отклик оператора Собеля с маской  $3 \times 3$ . (c) Графическое представление данных о направлении градиента, определенного с помощью оператора Собеля с маской  $3 \times 3$ .

$I[r, c] = z = pr + qc + z_0$ . (Расстояния между соседними пикселями окрестности одинаковы.) Докажите, что маски Превитт позволяют получить оценочные значения  $p$  и  $q$  как частные производные модели плоскости, подобранной методом наименьших квадратов для аппроксимации функции интенсивности.

На рис. 5.18 (b, c) представлены диаграммы значений интенсивности пикселей в двух выделенных строках изображения сцены внутри помещения. Как видно на исходном полутоновом изображении и на диаграмме, нижняя строка (b) пересекает четыре темные области. Это (1) обивка стула слева (столбцы 20-80), (2) кресло и одежда доктора Превитт в центре (столбцы 170-240), (3) тень от крайнего справа стула (столбцы 360-370) и (4) электрические провода (столбец 430). Отметим, что все переходы между темными и светлыми значениями интенсивности являются резкими, за исключением границы между стулом и его тенью. В области этой границы значения интенсивности постепенно уменьшаются от значения 220 до 20 на промежутке примерно в 10 пикселей вдоль строки. Диаграмма значений интенсивности для верхней строки, представленная на рис. 5.18 (c), содержит резкие перепады на границах рамок картин и матовой стены и на внутренних областях картин. По сравнению



**Рис. 5.18.** (a) Изображение Джудит Превитт с двумя выделенными строками. (b) Диаграмма значений интенсивности вдоль нижней выделенной строки. (c) Диаграмма значений интенсивности вдоль верхней выделенной строки. (d) *Градиентное изображение*, содержащее значения суммы  $|f_x| + |f_y|$ , вычисленной с помощью оператора Превитт с маской  $3 \times 3$ . (e) Диаграмма значений интенсивности вдоль нижней выделенной строки градиентного изображения. (f) Диаграмма значений интенсивности вдоль верхней выделенной строки градиентного изображения.



с картиной, висящей на стене справа, левая картина содержит намного больше изменений значений интенсивности.

На рис. 5.18 (d)–(f) показаны результаты применения к исходному изображению градиентного оператора Превитт с маской  $3 \times 3$ . Сумма абсолютных значений компонент градиента в направлении строк ( $f_x$ ) и столбцов ( $f_y$ ) показана на диаграммах для тех же строк изображений, для которых были построены диаграммы (a)–(c). Большие значения отклика оператора Превитт хорошо согласуются с местоположением основных границ, пересекаемых выделенными строками. Однако оказывается сложно интерпретировать несколько пиков средней величины, соответствующих креслу доктора Превитт и расположенных на рис. (d) в интервале между столбцами 170 и 210. Данные о контрасте вдоль верхней выделенной строки, показанные на диаграмме на рис. (f), интерпретируются аналогичным образом. Наибольшие значения модуля градиента хорошо согласуются с местоположением границ между рамками картин и матовой стеной, но также имеется большое количество перепадов интенсивности в области картины, висящей слева на стене.

В целом можно отметить, что градиентные операторы дают хорошие результаты при обнаружении границ изолированных объектов. Но у таких операторов есть ряд общих проблем. Из-за кривизны границ объектов или из-за плавных теневых переходов границы иногда теряются. С другой стороны, на высококонтрастных областях иногда возникают границы шириной в несколько пикселей. Для утоньшения таких границ может потребоваться морфологическая обработка. Градиентные операторы также могут давать большое количество откликов на текстурных областях (этот вопрос более подробно рассматривается в гл. 7).

## 5.7 ГАУССОВСКАЯ ФИЛЬТРАЦИЯ И ЛОГ-ФИЛЬТРАЦИЯ ДЛЯ ОБНАРУЖЕНИЯ КРАЕВ

Функция Гаусса используется во многих областях и приложениях математики, в том числе для фильтрации изображений. В данном разделе будут описаны те свойства функции Гаусса, которые делают ее полезной для сглаживания изображений и для обнаружения краев после сглаживания.

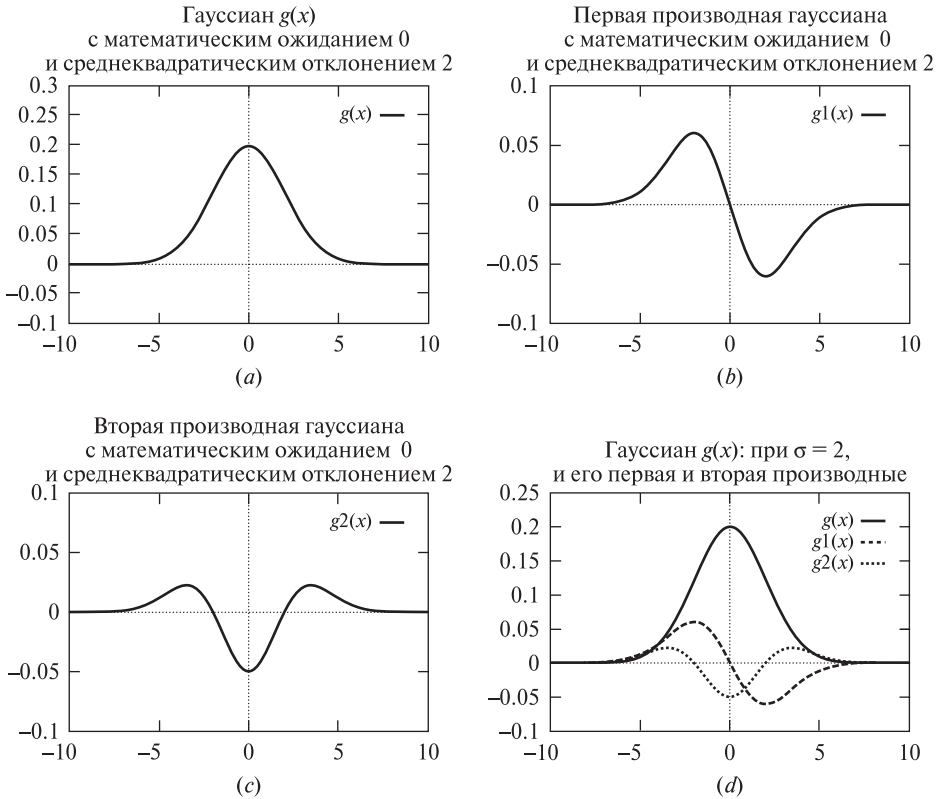
**46 Определение.** Функция Гаусса (гауссиан), зависящая от одной переменной, имеет вид:

$$g(x) = ce^{-\frac{x^2}{2\sigma^2}} \quad (5.15)$$

где  $\sigma$  — среднее квадратическое отклонение, а  $c$  — некоторый масштабный множитель.

Функция Гаусса двух переменных имеет вид:

$$g(x, y) = ce^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (5.16)$$



**Рис. 5.19.** (a) Гауссиан  $g(x)$  со среднеквадратическим отклонением  $\sigma = 2$ . (b) Первая производная  $g'(x)$ . (c) Вторая производная  $g''(x)$ . График этой функции напоминает поперечное сечение перевернутого сомбреро (мексиканской шляпы). (d) Все три функции показаны на одном графике, позволяющем увидеть соответствие положения нулей второй производной, точек экстремума первой производной и ветвей кривой гауссиана с максимальным наклоном.

Эти функции аналогичны функции нормального распределения (распределения Гаусса), которое рассматривалось в гл. 4. Константа  $c$  у функции нормального распределения была выбрана таким образом, чтобы площадь под кривой распределения была равна 1. При формировании маски для фильтрации значение  $c$  обычно выбирается таким, чтобы значения всех элементов маски были целочисленными.

Гауссиан задан в виде функции, симметричной относительно начала координат. Поэтому для него не требуется задавать параметр  $\mu$  (математическое ожидание), как для нормального распределения. Алгоритм обработки изображений будет выполнять перенос гауссиана таким образом, чтобы применять его к обрабатываемой окрестности сигнала или изображения.

На рис. 5.19 приведены графики гауссиана от одной переменной и его первой и второй производной. Производные гауссиана тоже имеют важное значение для операций фильтрации изображений. Получение функций производных гауссиана показано ниже, в выражениях (5.17)–(5.22). Площадь под кривой функции  $g(x)$  (т. е. интеграл этой функции) равна 1. Следовательно, эту функцию можно использовать в качестве сглаживающего фильтра, который не изменяет содержимого областей постоянных значений.

Функция  $g(x)$  является положительной четной функцией.  $g'(x)$  имеет вид функции  $g(x)$ , умноженной на нечетную функцию  $-x$  и деленную на  $\sigma^2$ . Функция  $g''(x)$  имеет более сложную структуру. Выражение (5.21) показывает, что  $g''(x)$  равна разности двух четных функций и что вершина выпуклости при  $x \approx 0$  лежит в отрицательной полуплоскости. Из выражения (5.22) следует, что вторая производная гауссиана обращается в 0 в точках  $x = \pm\sigma$  (см. также график на рис. 5.19).

$$g(x) = \frac{1}{\sqrt{2\pi} \sigma} e^{-\frac{x^2}{2\sigma^2}} \quad (5.17)$$

$$g'(x) = \frac{-1}{\sqrt{2\pi} \sigma^3} x e^{-\frac{x^2}{2\sigma^2}} \quad (5.18)$$

$$= \frac{-x}{\sigma^2} g(x) \quad (5.19)$$

$$g''(x) = \left( \frac{x^2}{\sqrt{2\pi} \sigma^5} - \frac{1}{\sqrt{2\pi} \sigma^3} \right) e^{-\frac{x^2}{2\sigma^2}} \quad (5.20)$$

$$= \frac{x^2}{\sigma^4} g(x) - \frac{1}{\sigma^2} g(x) \quad (5.21)$$

$$= \left( \frac{x^2}{\sigma^4} - \frac{1}{\sigma^2} \right) g(x) \quad (5.22)$$

С учетом свойств одномерного гауссиана, с помощью подстановки  $r = \sqrt{x^2 + y^2}$  можно получить выражение соответствующей двумерной функции  $g(x, y)$  и ее производных. Построение этих двумерных функций можно наглядно представить в форме вращения их одномерных аналогов вокруг вертикальной оси. Полученные двумерные функции являются *изотропными* функциями, у которых поперечные сечения в плоскостях, содержащих вертикальную ось, имеют одинаковый вид, совпадающий с одномерным гауссианом. Функция второй производной двумерного гауссиана представляет поверхность, часто называемую «сомбреро» или «мексиканской шляпой». Из математического выражения этой функции следует, что ее центральная выпуклость направлена вверх вдоль оси  $z = g(x, y)$ . Но на графиках и при фильтрации изображений эта функция обычно используется с обратным знаком, при этом выпуклость на графике функции будет направлена вверх и будет располагаться в положительной полуплоскости.

Далее перечислены несколько свойств гауссиана, имеющие важное значение для фильтрации изображений.

1. По мере увеличения расстояния от начала координат функция непрерывно убывает до 0. Следовательно, значения интенсивности изображения вблизи центральной выпуклости при фильтрации будут более значимыми, чем значения более далеко расположенных пикселей. От значения параметра  $\sigma$  зависит ширина обрабатываемой пиксельной окрестности — насколько она будет широкой или узкой. 95 процентов весовых коэффициентов маски фильтра лежат в пределах расстояния  $2\sigma$  от центра гауссиана.
2. Поворот функции относительно горизонтальной оси  $x$  не изменяет ядра, используемого в операции свертки.
3. Преобразование Фурье преобразует гауссиан в другой гауссиан в области частот. Поэтому свертка с маской гауссиана в пространственной области приводит к гладкому уменьшению высокочастотных компонент (в соответствии с увеличением значения частоты).
4. На графике второй производной одномерного гауссиана  $g''(x)$  заметна центральная выпуклость, лежащая в отрицательной полуплоскости, и две боковые выпуклости, лежащие в положительной полуплоскости. Функция обращается в ноль в точках  $-\sigma$  и  $+\sigma$ . Они соответствуют точкам перегиба  $g(x)$  и точкам экстремума  $g'(x)$ .
5. Дифференциальный фильтр второго порядка, основанный на применении оператора Лапласа к функции Гаусса, называется **ЛОГ-фильтром** (Laplacian of the Gaussian, лапласиан гауссиана). ЛОГ-фильтр можно приближенно представить в виде разности двух гауссианов  $g''(x) \approx c_1 e^{-\frac{x^2}{2\sigma_1^2}} - c_2 e^{-\frac{x^2}{2\sigma_2^2}}$ . Такая аппроксимация часто называется **DOG-фильтром** (Difference Of Gaussians, разность гауссианов). Чтобы центральная выпуклость лежала в положительной полуплоскости, должно быть  $\sigma_1 < \sigma_2$ . Значение  $\sigma_2$  должно выбираться с учетом значения  $\sigma_1$  так, чтобы точки перехода через ноль среди значений отклика фильтра располагались корректно и чтобы сумма отрицательных коэффициентов фильтра уравновешивалась суммой положительных коэффициентов.
6. ЛОГ-фильтр дает значимый отклик для разностей интенсивности двух типов — для небольших пятен, центр которых совпадает с центром маски фильтра, и для широких ступенчатых краев, расположенных близко от центра маски.

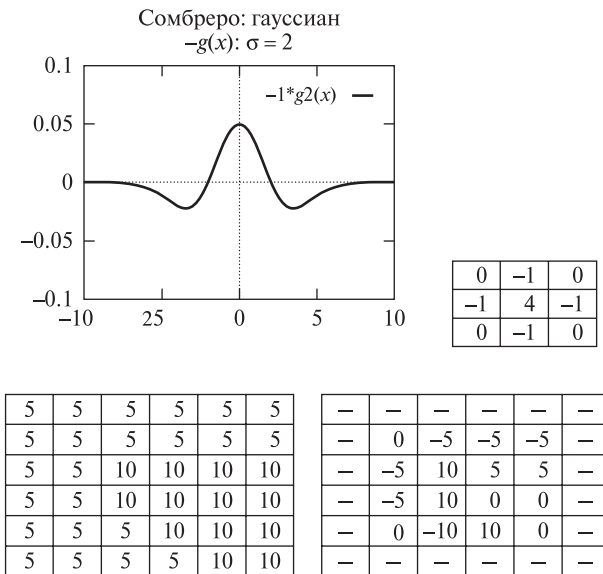
$$\mathbf{G}_{3 \times 3} = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}; \quad \mathbf{G}_{7 \times 7} = \begin{bmatrix} 1 & 3 & 7 & 9 & 7 & 3 & 1 \\ 3 & 12 & 26 & 33 & 26 & 12 & 3 \\ 7 & 26 & 55 & 70 & 55 & 26 & 7 \\ 9 & 33 & 70 & 90 & 70 & 33 & 9 \\ 7 & 26 & 55 & 70 & 55 & 26 & 7 \\ 3 & 12 & 26 & 33 & 26 & 12 & 3 \\ 1 & 3 & 7 & 9 & 7 & 3 & 1 \end{bmatrix}$$

**Рис. 5.20.** (Слева) маска  $3 \times 3$  для аппроксимации гауссиана была вычислена посредством матричного умножения  $[1, 2, 1]^t \otimes [1, 2, 1]$ . (Справа) маска  $7 \times 7$  для аппроксимации гауссиана при  $\sigma^2 = 2$  была получена с помощью выражения (5.16). Сначала вычислялись значения функции при целочисленных  $x$  и  $y$ , а затем значение константы было выбрано  $c = 90$  так, чтобы минимальный элемент маски был равен 1.

На рис. 5.20 показаны две различные маски гауссовского сглаживания. Маски для обнаружения краев рассматриваются ниже.

### 5.7.1 Обнаружение краев с помощью ЛОГ-фильтра

На рис. 5.21 и 5.22 представлены две различные маски для реализации ЛОГ-фильтра. Первая маска имеет размеры  $3 \times 3$ . Эта маска ЛОГ-фильтра наименьшего возможного размера позволяет детектировать детали изображе-



**Рис. 5.21.** (Сверху) поперечное сечение функции ЛОГ-фильтра и маска  $3 \times 3$  для аппроксимации этой функции. (Снизу) входное изображение и результат его обработки с использованием приведенной выше маски.

0	0	0	-1	-1	-2	-1	-1	0	0	0
0	0	-2	-4	-8	-9	-8	-4	-2	0	0
0	-2	-7	-15	-22	-23	-22	15	-7	-2	0
-1	-4	-15	-24	-14	-1	-14	-24	-15	-4	-1
-1	-8	-22	-14	52	103	52	-14	-22	-8	-1
-2	-9	-23	-1	103	178	103	-1	-23	-9	-2
-1	-8	-22	-14	52	103	52	-14	-22	-8	-1
-1	-4	-15	-24	-14	-1	-14	-24	-15	-4	-1
0	-2	-7	-15	-22	-23	-22	15	-7	-2	0
0	0	-2	-4	-8	-9	-8	-4	-2	0	0
0	0	0	-1	-1	-2	-1	-1	0	0	0

**Рис. 5.22.** Маска  $11 \times 11$ , аппроксимирующая лапласиан от гауссиана при  $\sigma^2 = 2$ . (Соответствует книге [Naralick, Shapiro, том I, с. 349].)

ния с характерным размером порядка одного пиксела. Отклик маски  $11 \times 11$  вычисляется с учетом 121-го входного значения. Эта маска дает значимый отклик на крупные признаки изображения и не реагирует на малые. В случае программной реализации фильтрации обработка 121 пиксела может потребовать значительно больше времени, чем 9 пикселей.

### Упражнение 5.10. Свойства ЛОГ-фильтра

Предположим, что 9 значений интенсивности из окрестности изображения  $3 \times 3$  можно идеально аппроксимировать с помощью модели плоскости  $I[r, c] = z = pr + qc + z_0$ . (Отметим, что соседние точки в направлениях  $r$  и  $c$  расположены на равных расстояниях.) Покажите, что простая маска ЛОГ-фильтра

0	-1	0
-1	4	-1
0	-1	0

дает нулевой отклик на такую окрестность. Это означает, что

ЛОГ-фильтр дает нулевой отклик как на постоянные области, так и на области с наклонным изменением значений интенсивности.

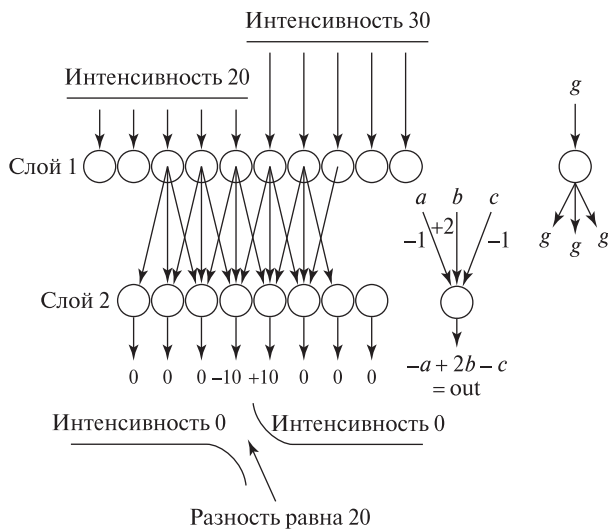
## 5.7.2 Обнаружение краев в зрительной системе человека

Рассмотрим теперь архитектуру искусственной нейронной сети (artificial neural network, ANN), на основе которой можно построить параллельную реализацию ЛОГ-фильтрации. Было продемонстрировано, что искусственная нейронная сеть способна имитировать некоторые свойства зрительной системы человека. Кроме того, в результате лабораторных экспериментов над животными были получены данные о том, что в зрительных системах кошек и обезьян генерируются электрические сигналы, которые можно смоделировать с помощью искусственных нейронных сетей.

Обработка одномерных сигналов поясняется на рис. 5.23. Ступенчатый край воспринимается в различных точках ячейками светочувствительного массива («сетчатки»). Эти ячейки первого слоя посылают сигналы интегрирующим ячейкам второго слоя. Каждому соединению между  $i$ -й ячейкой первого слоя и  $j$ -й ячейкой второго слоя сопоставлен весовой коэффициент  $w_{ij}$ . Значения, передаваемые от ячеек первого слоя ячейке второго слоя, сначала умножаются на весовые коэффициенты, а затем суммируются. Следовательно, выходное значение  $j$ -й ячейки равно  $y_j = \sum_{i=1}^N w_{ij} x_i$ , где  $x_i$  представляет выходное значение  $i$ -й ячейки первого слоя, а  $N$  равно количеству ячеек первого слоя. (В действительности требуется учитывать только те ячейки  $i$ , которые непосредственно соединены с рассматриваемой  $j$ -й ячейкой второго слоя.) Так как весовые коэффициенты назначаются всем соединениям ячеек, то возможно (и так часто делается), чтобы одна и та же ячейка  $i$  передавала положительные значения ячейке  $j$ , а отрицательные — другой ячейке  $k \neq j$ .

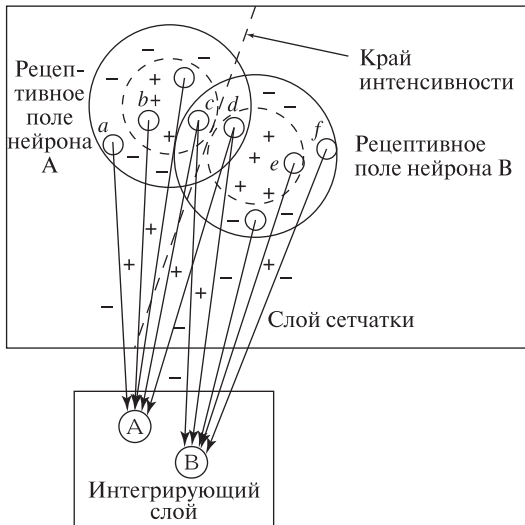
На рис. 5.23 показана сеть, в которой выходное значение каждой  $j$ -й ячейки второго слоя вычисляется согласно выражению  $-a + 2b - c$ . Этому выражению соответствует маска  $[-1, 2, -1]$ . К центральному входному сигналу применяется весовой коэффициент, равный 2, а к двум другим (тормозящим) входным сигналам  $a$  и  $c$  применяются весовые коэффициенты, равные  $-1$ .

Подобную архитектуру можно определить для маски любого вида. Эта архитектура обеспечивает параллельную реализацию кросс-корреляции для фильтрации и обнаружения признаков.



**Рис. 5.23.** Моделирование эффекта *полос Маха* с помощью искусственной нейронной сети. Интенсивность воспринимается с помощью ячеек сетчатки (Слой 1), которые затем передают входные сигналы интегрирующим ячейкам следующего слоя (Слой 2).

**Рис. 5.24.** Семь полос постоянной интенсивности, заполненных полутоновыми значениями  $31 + 32k$ , где  $k = 1, 7$ . Вследствие эффекта полос Маха люди могут увидеть на этом изображении ребристые или вогнутые пластины.



**Рис. 5.25.** Архитектура трехмерной искусственной нейронной сети для выполнения ЛОГ-фильтрации.

Психолог Мах обнаружил, что при наблюдении края между двумя областями люди воспринимают край так, как будто он растянут для усиления различия интенсивности по разные стороны края (как на рис. 5.23). Отметим, что рассмотренные маска и нейронная сеть формируют *точки перехода через ноль* в местоположении края, которое соответствует границе между двумя ячейками, одна из которых генерирует положительное, а другая — отрицательное входное значение. Эффект полос Мах приводит к изменению восприятия формы соединяющихся поверхностей и очевидным образом проявляется в компьютерной графике при отображении многогранных объектов с тонированными гранями. На рис. 5.24 показаны семь однородных областей, интенсивность которых выбрана из интервала значений  $31 \dots 255$  с шагом 32. Можете ли вы заметить на этом изображении трехмерный объект, напоминающий фрагмент колонны древнегреческого храма?

Применение нейронной сети для обработки двумерных изображений поясняется на рис. 5.25. Каждое множество ячеек сетчатки, связанных с интегрирующей ячейкой  $j$ , называется *рецептивным полем* этой ячейки. Для обнару-



жения краев на основе вычисления второй производной, каждое рецептивное поле содержит центральное множество ячеек с положительными весовыми коэффициентами  $w_{ij}$  (для соединений с  $j$ -й ячейкой) и окружающее множество периферических ячеек с отрицательными весовыми коэффициентами. Ячейки сетчатки  $b$  и  $c$  расположены в центре рецептивного поля интегрирующей ячейки  $A$ , а ячейки  $a$  и  $d$  расположены на периферии рецептивного поля и генерируют тормозящие входные сигналы. Но по отношению к интегрирующей ячейке  $B$  ячейка  $d$  лежит в центре рецептивного поля, а ячейка  $c$  — на периферии. Сумма весовых коэффициентов центральных и периферических ячеек рецептивного поля должна быть равна 0, чтобы интегрирующая ячейка генерировала нейтральный выходной сигнал на областях постоянной интенсивности. Так как центральная и периферическая области имеют круговую форму, то выходной сигнал будет отличаться от нейтрального каждый раз, когда в центральную область под любым углом будет попадать прямолинейный фрагмент границы. Поэтому каждая интегрирующая ячейка является *изотропным* детектором края. Кроме того, если в центральной части рецептивного поля окажется малая область изображения, контрастная с окружающим фоном, то интегрирующая ячейка тоже даст значимый отклик. Следовательно, ячейка также является детектором пятен (малых контрастных объектов).

На рис. 5.21 показан результат применения ЛОГ-маски наименьшего возможного размера к изображению, содержащему две области. Результат, представленный в правой части рисунка, демонстрирует, как в результате обработки граница между областями оказывается выделена в виде переходов через ноль. На рис. 5.22 представлена маска  $11 \times 11$ , соответствующая лапласиану от гауссиана при  $\sigma^2 = 2$ . Маска малого размера позволяет обнаруживать границы между малыми областями и является чувствительной к фрагментам границ с большой кривизной, но также дает отклики на шумовые значения. Маска большего размера выполняет более существенное сглаживание и дает отклик только на гладкие границы между большими областями.

---

### Упражнение 5.11

В дополнение к вышеизложенному материалу, приведите более подробные аргументы в подтверждение того, что интегрирующие ячейки на рис. 5.25: (а) дают отклик на контрастные пятна и «каплеобразные» образы, попадающие в центральную часть рецептивного поля, и (б) дают отклик на границы между двумя большими областями, если эти границы пересекают центральную часть рецептивного поля.

### 5.7.3 Теория Марра—Хилдрет

Дэвид Марр (David Marr) и Эллен Хилдрет (Ellen Hildreth) выдвинули гипотезу, объясняющую низкоуровневые свойства человеческого зрения на основе ЛОГ-фильтрации. Марр предположил, что цель низкоуровневой обработки

в зрительной системе человека заключается в построении *первоначального эскиза* (primal sketch), для которого можно сформировать двумерное описание, состоящее из прямых, ребер и пятен. (Первоначальные эскизы, полученные от двух глаз, затем могут подвергаться дальнейшей обработке для формирования трехмерной интерпретации сцены.) Для построения первоначального эскиза Марр и Хилдрет предложили использовать ЛОГ-фильтрацию с 4 или 5 различными значениями  $\sigma$ . Перечисленные выше математические свойства позволяют объяснить как результаты экспериментов по зрительному восприятию людей, так и эксперименты по анализу электрических сигналов в зрительной системе животных. ЛОГ-фильтры с большими значениями  $\sigma$  будут выделять широкие края, а фильтры с малыми значениями  $\sigma$  в основном будут выделять мелкие детали. Согласование выходных данных фильтров различного масштаба может выполняться на более высоком уровне обработки. Например, результаты, полученные при анализе данных крупномасштабной фильтрации, можно затем учитывать в процессе обработки данных от фильтров малого размера. В последующих работах были предложены различные практические методы использования *масштабного пространства* (scale space) для обработки выходных данных от фильтров разных размеров.

На рис. 5.26 представлены результаты обработки изображения с использованием двух различных уровней гауссовского сглаживания. Центральное изображение может служить хорошим представлением основных объектов и краев исходного изображения. Изображение, показанное справа, содержит больше деталей исходного изображения, но вместе с тем, больше шумовых помех. Отметим, что корабль (виден вдали в правой части изображения) и граница песка и воды есть на правом изображении, но отсутствуют на центральном.



(a) Входное изображение

(b) Результат сглаживания при  $\sigma = 4$ (c) Результат сглаживания при  $\sigma = 1$ 

**Рис. 5.26.** Выделение краев на входном изображении (a) после предварительного сглаживания с использованием гауссовских фильтров со среднеквадратическим отклонением (b)  $\sigma = 4$  и (c)  $\sigma = 1$ . При использовании фильтра малого размера на изображении присутствует больше деталей, но вместе с тем больше шума. (С любезного разрешения David Shaffer, 1998.)



**Рис. 5.27.** (Слева) виртуальная кривая, образованная концевыми точками полос линейчатой текстуры (возможно, это изображение двух перекрывающихся кусочков оберточной бумаги). (В центре) результат обработки с использованием специфического ЛОГ-фильтра с маской  $4 \times 4$ , который дает отклик как на края, так и на концевые точки. (Справа) результат обработки с использованием другого ЛОГ-фильтра с маской  $3 \times 3$ , который откликается только на концевые точки.



**Рис. 5.28.** Изображение, полученное с помощью пороговой бинаризации. Границы объектов образованы виртуальными кривыми, которые определяются концевыми областями полос, имеющих вид поперечных сечений обобщенных цилиндров. (С любезного разрешения Eleanor Harding.)

В первоначальном эскизе, предложенном Марром, также содержится описание виртуальных кривых. Они задаются однотипными обнаруженными признаками, расположенными вдоль кривых на изображении. Например, это могут быть пунктирные отрезки, образы расположенных в ряд кустарников и т. п. На рис. 5.27 представлен пример искусственного изображения, которое содержит виртуальную кривую, и результаты обработки этого изображения с использованием двух различных ЛОГ-фильтров. Оба ЛОГ-фильтра дают

отклик на концах полос. К краям полос чувствителен только один из этих фильтров. Виртуальные кривые также можно обнаружить на рис. 5.28. На нем показано изображение природной сцены, которое было подвергнуто пороговой бинаризации с целью получения декоративной текстуры.

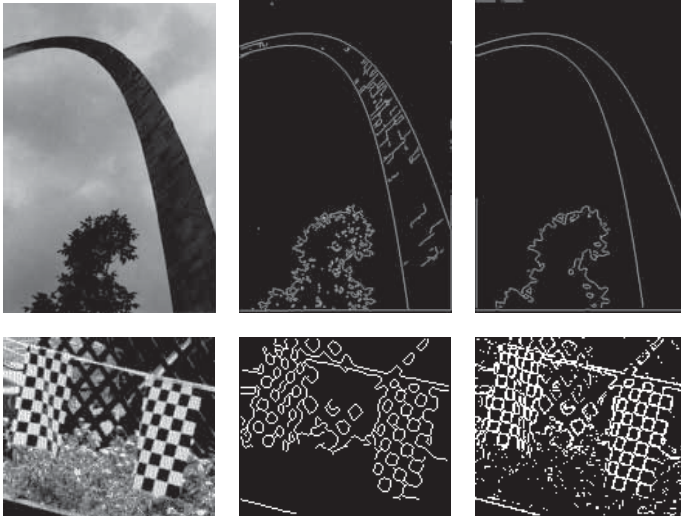
В последнее время наблюдается значительный прогресс в области исследований зрительной системы и мозга человека. Они свидетельствуют о необходимости построения более сложных моделей, чем модель Марра и Хилдрет. Тем не менее, методы обработки на основе использования гауссианов и ЛОГ-фильтров переменного размера находят широкое применение в компьютерном зрении.

## 5.8 ДЕТЕКТОР КРАЕВ КЭННИ

Детектор краев Кэнни (Canny) является очень популярным и эффективным оператором. Здесь он рассматривается очень кратко, а более подробное описание приведено в гл. 10. Оператор Кэнни сначала выполняет сглаживание полутонового изображения, а затем формирует протяженные контурные сегменты, прослеживая соседние пиксели с большими значениями модуля градиента. На рис. 5.29 показаны результаты выделения краев на достаточно сложных изображениях естественных сцен. Контурные арки в Сент-Луисе на рис. 5.29 выделены довольно хорошо. При использовании значения  $\sigma = 1$  выделяются некоторые швы на металлическом покрытии арки и изменения интенсивности в области кроны дерева. При использовании значения  $\sigma = 4$  выделяются только внешние границы этих объектов. Как показано в нижней части рис. 5.29, оператор Кэнни обнаруживает много текстурных элементов флага с шахматной текстурой. Для сравнения представлены результаты применения оператора Робертса с небольшим пороговым ограничением по величине градиента. На выходном изображении заметно больше текстурных элементов сцены (в области травы и изгороди), хотя это изображение менее структурировано, чем выходное изображение оператора Кэнни. Алгоритм формирования контурных сегментов подробно рассматривается в разделе 10.3.2.

## 5.9 ИСПОЛЬЗОВАНИЕ МАСОК В КАЧЕСТВЕ СОГЛАСОВАННЫХ ФИЛЬТРОВ\*

В данном разделе вводятся основные теоретические понятия, касающиеся использования отклика маски на некоторую окрестность изображения в качестве меры сходства (согласования) этой окрестности с маской. На практике это позволяет разрабатывать маски для выделения признаков конкретного вида. Этот метод может применяться для обнаружения краев и текстуры, а также для выделения признаков более специфического вида, например, отверстий или углов. Понятия согласованной фильтрации будут сначала представлены применительно к одномерным сигналам, которые сами по себе имеют важное



**Рис. 5.29.** (Сверху слева) изображение большой арки в Сент-Луисе. (Сверху в центре) результаты обработки оператором Кэнни при  $\sigma = 1$ . (Сверху справа) результаты обработки оператором Кэнни при  $\sigma = 4$ . (Внизу слева) изображение с текстурными областями. (Снизу в центре) результаты обработки оператором Кэнни при  $\sigma = 1$ . (Снизу справа) результаты обработки оператором Робертса с применением порогового значения для отбора 20 процентов пикселей с наибольшими значениями модуля градиента.

значение. В качестве таких сигналов можно рассматривать строки и столбцы произвольного фрагмента двумерного изображения. Понятия и математические приемы, определенные для одномерных сигналов, непосредственным образом обобщаются на двумерный случай.

### 5.9.1 Векторное пространство всех сигналов, заданных в виде $n$ дискретных значений

Множество всех векторов, состоящих из  $n$  действительных чисел (для любого  $n \geq 1$ ), образует *векторное пространство*. Ниже кратко описываются некоторые операции над векторным пространством, которые будут использоваться в дальнейшем. Читатель, вероятно, при изучении аналитической геометрии или математического анализа уже встречался с векторами на плоскости ( $n = 2$ ) и в трехмерном пространстве ( $n = 3$ ). Понятие длины вектора при  $n = 2$  и  $n = 3$  совпадает с тем, которое использовалось в геометрии на плоскости и в трехмерной аналитической геометрии. В области обработки сигналов длина вектора связывается с энергией сигнала. Энергия сигнала определяется как квадрат длины вектора, представляющего сигнал, или, эквивалентным образом, как сумма квадратов всех координат вектора. Как будет показано ниже, понятие энергии сигнала имеет важное практическое значение.

**47 Определение.** Энергия сигнала  $S = [s_1, s_2, \dots, s_n]$  равна  $\|S\|^2 = s_1^2 + s_2^2 + \dots + s_n^2$ .

**Основные понятия векторного пространства, на котором определена длина вектора**

В приведенных определениях  $U$  и  $V$  обозначают два произвольных вектора, а  $u_i$  и  $v_i$  — координаты этих векторов, являющиеся действительными числами. Символами  $a, b, c$  и т. д. обозначены произвольные действительные числа (скалярные величины).

**48 Определение.** Векторная сумма двух векторов  $U = [u_1, u_2, \dots, u_n]$  и  $V = [v_1, v_2, \dots, v_n]$  равна вектору  $U \oplus V = [u_1 + v_1, u_2 + v_2, \dots, u_n + v_n]$ .

**49 Определение.** Произведение вектора  $V = [v_1, v_2, \dots, v_n]$  на число  $a$  равно вектору  $aV = [av_1, av_2, \dots, av_n]$ .

**50 Определение.** Скалярное произведение двух векторов  $U = [u_1, u_2, \dots, u_n]$  и  $V = [v_1, v_2, \dots, v_n]$  равно действительному числу  $U \circ V = u_1v_1 + u_2v_2 + \dots + u_nv_n$ .

**51 Определение.** Длина, или норма, вектора  $V = [v_1, v_2, \dots, v_n]$  является неотрицательным действительным числом и равна  $\|V\| = V \circ V = (v_1v_1 + v_2v_2 + \dots + v_nv_n)^{1/2}$ .

**52 Определение.** Векторы  $U$  и  $V$  являются ортогональными тогда и только тогда, когда  $U \circ V = 0$ .

**53 Определение.** Расстояние между векторами  $U = [u_1, u_2, \dots, u_n]$  и  $V = [v_1, v_2, \dots, v_n]$  равно длине вектора их разности  $d(U, V) = \|U - V\|$ .

**54 Определение.** Базис  $n$ -мерного векторного пространства представляет собой множество  $n$  векторов  $\{w_1, w_2, \dots, w_n\}$ , которые являются линейно независимыми и покрывают векторное пространство. Свойство покрытия состоит в том, что любой вектор  $V$  можно представить в виде линейной комбинации базисных векторов:  $V = a_1w_1 \oplus a_2w_2 \oplus \dots \oplus a_nw_n$ . Свойство линейной независимости означает, что никакие из этих базисных векторов  $w_i$  нельзя представить в виде линейной комбинации остальных базисных векторов.

Хотя компоненты векторов, представляющих сигналы, являются действительными числами, во многих приложениях сигналы не могут принимать

отрицательных значений. Например, отрицательных координат не может быть у 12-мерного вектора, представляющего количество осадков в некотором месте для каждого из 12 месяцев года. Значения интенсивности пикселей строки изображения также обычно принадлежат неотрицательному целочисленному диапазону. Тем не менее интерпретация на основе векторного пространства остается полезной и в таких случаях. Часто в процессе интерпретации сигнала из всех его координат вычитается среднее значение сигнала. При этом некоторые значения сигнала становятся отрицательными. Отрицательные значения часто используются в качестве коэффициентов масок, применяемых в качестве шаблонов или моделей формы фрагментов сигналов.

**Свойства векторных пространств, вытекающие из приведенных выше определений.**

1.  $U \oplus V = V \oplus U$
2.  $U \oplus (V \oplus W) = (U \oplus V) \oplus W$
3. Существует нулевой вектор  $O$  такой, что для всех векторов  $V$  выполняется равенство  $O \oplus V = V$
4. Для каждого вектора  $V$  существует такой вектор  $(-1)V$ , что  $V \oplus (-1)V = O$
5. Для любых чисел  $a$  и  $b$  и любого вектора  $V$  выполняется равенство  $a(bV) = (ab)V$
6. Для любых чисел  $a$  и  $b$  и любого вектора  $V$  выполняется равенство  $(a + b)V = aV \oplus bV$
7. Для любого числа  $a$  и двух векторов  $U$  и  $V$  выполняется равенство  $a(U \oplus V) = aU \oplus aV$
8. Для любого вектора  $V$ ,  $1V = V$
9. Для любого вектора  $V$ ,  $(-1V) \circ V = -\|V\|^2$

### Упражнение 5.12

Докажите справедливость любых 5 из 9 перечисленных свойств векторного пространства.

### 5.9.2 Использование ортогонального базиса

Два наиболее важных результата изучения векторных пространств состоят в том, что (1) любой вектор можно единственным способом представить в виде линейной комбинации базисных векторов и (2) любое множество базисных векторов должно содержать в точности  $n$  векторов. *Ортогональный базис* позволяет получить полезную наглядную интерпретацию представления произвольного вектора  $V$ .

В приведенном далее примере показано, как можно представить сигнал  $[10, 15, 20]$  с использованием базиса из трех заданных векторов  $\{w_1, w_2, w_3\}$ , обладающих специальными свойствами. Любой сигнал  $S = [a_1, a_2, a_3]$  можно представить в виде  $S = a_1 w_1 \oplus a_2 w_2 \oplus a_3 w_3$ . Тогда  $S \circ w_i = a_1(w_1 \circ w_i) \oplus a_2(w_2 \circ w_i) \oplus a_3(w_3 \circ w_i) = a_i(w_i \circ w_i) = a_i$ , если скалярное произведение  $w_i \circ w_j$  равно 0 при  $i \neq j$  и равно 1 при  $i = j$ .

### Пример представления сигнала в виде комбинации базисных сигналов.

Рассмотрим векторное пространство всех сигналов, состоящих из 3 дискретных значений, т. е.  $n = 3$  и сигналы имеют вид  $[v_1, v_2, v_3]$ . С помощью *стандартного базиса* любой вектор  $V$  можно представить в виде  $V = [v_1, v_2, v_3] = v_1[1, 0, 0] \oplus v_2[0, 1, 0] \oplus v_3[0, 0, 1]$ . Стандартные базисные векторы являются ортогональными и имеют единичную длину. Такое множество базисных векторов называется *ортонормированным*.

Рассмотрим теперь другое множество базисных векторов  $\{w_1, w_2, w_3\}$ , где  $w_1 = [-1, 0, 1]$ ,  $w_2 = [1, 1, 1]$  и  $w_3 = [-1, 2, -1]$ . Любые два из этих базисных векторов ортогональны, так как  $w_i \circ w_j = 0$  при  $i \neq j$  (убедитесь в этом). Масштабируя эти векторы для приведения к единичной длине, получим новый базис  $\{\frac{1}{\sqrt{2}}[-1, 0, 1], \frac{1}{\sqrt{3}}[1, 1, 1], \frac{1}{\sqrt{6}}[-1, 2, -1]\}$ .

Теперь представим сигнал  $S = [10, 15, 20]$  в новом ортонормированном базисе. Координаты  $[10, 15, 20]$  вектор  $S$  имеет относительно *стандартного базиса*.

$$\begin{aligned} S \circ w_1 &= \frac{1}{\sqrt{2}}(-10 + 0 + 20) \\ S \circ w_2 &= \frac{1}{\sqrt{3}}(10 + 15 + 20) \\ S \circ w_3 &= \frac{1}{\sqrt{6}}(-10 + 30 - 20) \\ S &= (S \circ w_1)w_1 \oplus (S \circ w_2)w_2 \oplus (S \circ w_3)w_3 \end{aligned} \quad (5.23)$$

$$S = (10/\sqrt{2})w_1 \oplus (45/\sqrt{3})w_2 \oplus 0w_3$$

$$\|S\|^2 = 100 + 225 + 400 = 725$$

$$= (10/\sqrt{2})^2 + (45/\sqrt{3})^2 + 0^2 = 725$$

Последние два уравнения показывают, что при использовании ортонормированного базиса полную энергию легко можно вычислить в виде суммы энергетических составляющих, связанных с каждым базисным вектором.



Итак, работать с ортонормированным базисом очень удобно. С его помощью можно представить энергию сигнала в виде суммы отдельных энергетических компонент, связанных с каждым базисным вектором. Предположим, что рассмотренный пример повторяется для сигнала  $S_2 = [-5, 0, 5]$ , который можно получить, вычитая из сигнала  $S$  его среднее значение:  $S_2 = S \oplus (-1[15, 15, 15])$ .

Вектор сигнала  $S_2$  совпадает с вектором  $S \circ w_1$ , так как из него была удалена компонента в направлении  $[1, 1, 1]$ .  $S_2$  можно представить в виде произведения вектора  $w_1$  на действительное число  $S_2 = (10/\sqrt{2})w_1 = (10/\sqrt{2})((1/\sqrt{2})[-1, 0, 1]) = [-5, 0, 5]$ . Будем говорить, что структура вектора  $S_2$  совпадает со структурой вектора  $w_1$ . Если  $w_1$  использовать в качестве фильтра, то он будет очень хорошо согласован с сигналом  $S_2$ . В некотором смысле, он также окажется хорошо согласованным с сигналом  $S$ . Подробнее этот подход будет рассмотрен далее. Отметим также, что в  $n$ -мерном векторном пространстве векторов сигналов существует множество различных ортонормированных базисов.

### Упражнение 5.13

(а) По аналогии с рассмотренным примером, запишите представление вектора  $[10, 14, 15]$  в базисе  $\{\frac{1}{\sqrt{2}}[-1, 0, 1], \frac{1}{\sqrt{3}}[1, 1, 1], \frac{1}{\sqrt{6}}[-1, 2, -1]\}$ . (б) Теперь запишите представление вектора  $[10, 19, 10]$ : на какой базисный вектор он похож сильнее всего? Почему?

С учетом свойств векторов и скалярного произведения можно получить *неравенство Коши—Шварца* (5.24). Его основной смысл заключается в том, что значение скалярного произведения единичных векторов должно лежать в диапазоне от  $-1$  до  $1$ . Следовательно, эту величину можно использовать в качестве меры сходства двух векторов. Отметим, что если  $U = V$ , то скалярное произведение равно  $+1$ , а при  $U = -V$  скалярное произведение равно  $-1$ .

*Нормированное скалярное произведение* позволяет вычислить угол между двумя векторами. Значение этого угла совпадает с тем, которое можно получить с помощью тригонометрических формул в двумерном или трехмерном пространстве. При  $n \geq 3$  угол, или его косинус, является абстрактной мерой сходства двух векторов. Если нормированное скалярное произведение векторов равно  $0$ , то векторы считаются непохожими, если же скалярное произведение равно  $1$ , то векторы считаются максимально похожими. Скалярное произведение равно  $-1$  в том случае, когда векторы направлены противоположно. В таком случае, в зависимости от конкретной задачи, они могут считаться как похожими, так и непохожими.

### 5.9.3 Неравенство Коши—Шварца

Для любых двух ненулевых векторов  $U$  и  $V$  выполняется условие

$$-1 \leq \frac{U \circ V}{\|U\| \|V\|} \leq +1 \quad (5.24)$$

**55 Определение.** Нормированное скалярное произведение двух произвольных ненулевых векторов  $U$  и  $V$  равно  $(\frac{U \circ V}{\|U\| \|V\|})$

**56 Определение.** Угол между двумя произвольными ненулевыми векторами  $U$  и  $V$  равен  $\cos^{-1}\left(\frac{U \circ V}{\|U\| \|V\|}\right)$

### Упражнение 5.14

Постройте на графике пять векторов:  $[5, 5]$ ,  $[10, 10]$ ,  $[-5, 5]$ ,  $[-5, -5]$ ,  $[-10, 10]$ . Для каждой пары перечисленных векторов вычислите нормированное скалярное произведение или косинус угла. Какие векторы перпендикулярны? Какие векторы направлены одинаково, а какие противоположно? Сопоставьте относительное направление векторов со значением их нормированного скалярного произведения.

## 5.9.4 Векторное пространство изображений размерами $m \times n$

Множество всех матриц  $m \times n$  с действительными значениями образует векторное пространство размерности  $m \times n$ . Далее будет показано, как основные понятия и операции в векторном пространстве можно использовать применительно к маскам и областям изображений. В данном разделе в качестве модели изображения будет рассматриваться функция изображения  $I[x, y]$ , заданная на дискретной области определения из  $m \times n$  точек. В основном будут использоваться матрицы  $2 \times 2$  и  $3 \times 3$ , но весь материал легко расширить на случай изображений и масок произвольного размера.

## 5.9.5 Базис Робертса для окрестностей $2 \times 2$

Для интерпретации структуры окрестности  $2 \times 2$  полутонового изображения можно использовать базис, показанный на рис. 5.30. Этот базис называется *базисом Робертса*. Два из четырех векторов этого базиса уже были показаны на рис. 5.15. Как показывает следующее упражнение, любая окрестность  $2 \times 2$  действительных значений интенсивности может быть уникальным образом представлена в виде суммы четырех этих базисных векторов, взятых с соответствующими масштабными множителями. Относительная величина масштабного множителя непосредственно характеризует сходство окрестности изображения с базисным вектором и, следовательно, может использоваться для интерпретации структуры окрестности. Несколько примеров приведены на рис. 5.30.

### Упражнение 5.15

Убедитесь, что базисные вектора Робертса, показанные на рис. 5.30, являются ортонормированными.

### Упражнение 5.16

Будем рассматривать векторное пространство всех изображений  $2 \times 2$ , пиксели которых имеют действительные значения. (а) Определите значения  $a_j$  для

$$\text{базис Робертса: } \mathbf{W}_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad \mathbf{W}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad \mathbf{W}_3 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad \mathbf{W}_4 = \frac{1}{2} \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$\text{Область постоянных значений: } \begin{bmatrix} 5 & 5 \\ 5 & 5 \end{bmatrix} = \frac{20}{2} \left( \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \right) = 10\mathbf{W}_1 \oplus 0\mathbf{W}_2 \oplus 0\mathbf{W}_3 \oplus 0\mathbf{W}_4$$

$$\text{Ступенчатый край: } \begin{bmatrix} -1 & +1 \\ -1 & +1 \end{bmatrix} = 0\mathbf{W}_1 \oplus \frac{2}{\sqrt{2}}\mathbf{W}_2 \oplus \frac{-2}{\sqrt{2}}\mathbf{W}_3 \oplus 0\mathbf{W}_4$$

$$\text{Ступенчатый край: } \begin{bmatrix} +1 & +1 \\ -3 & +1 \end{bmatrix} = 0\mathbf{W}_1 \oplus \frac{4}{\sqrt{2}}\mathbf{W}_2 \oplus 0\mathbf{W}_3 \oplus \frac{-4}{2}\mathbf{W}_4$$

$$\text{Линия: } \begin{bmatrix} 0 & 8 \\ 8 & 0 \end{bmatrix} = 8\mathbf{W}_1 \oplus 0\mathbf{W}_2 \oplus 0\mathbf{W}_3 \oplus 8\mathbf{W}_4$$

**Рис. 5.30.** (Верхняя строка) базис для представления всех изображений размеров  $2 \times 2$ . В этот базис входят две градиентные маски Робертса. (Строка 2) область постоянных значений можно представить в виде произведения константного базисного изображения на число. (Строка 3) вертикально ориентированный ступенчатый край характеризуется энергией только в направлении градиентных масок. (Строка 4) диагонально ориентированный край характеризуется наибольшей энергией в направлении согласованной (с этим направлением) градиентной маски. (Строка 5) Энергия линейного шаблона распределена между константной маской  $\mathbf{W}_1$  и маской линии  $\mathbf{W}_4$ .

представления изображения  $\begin{bmatrix} 10 & 5 \\ 5 & 0 \end{bmatrix}$  в виде линейной комбинации четырех изображений  $\mathbf{W}_j$  базиса Робертса. (б) Объясните, почему для любого изображения  $2 \times 2$  всегда можно найти уникальные значения  $a_j$ .

### Упражнение 5.17.

Предположим, что изображение  $2 \times 2$  вида  $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$  характеризуется энергетическими компонентами  $e_1, e_2, e_3, e_4$ , соответствующими четырем базисным векторам Робертса  $\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3, \mathbf{W}_4$ . Запишите формулы для вычисления значений  $e_i$  с использованием  $a, b, c$  и  $d$ .

### 5.9.6 Базис Фрея—Чена для окрестностей $3 \times 3$

Для обработки изображений обычно используются маски размерами  $3 \times 3$  и более. Стандартный базис для окрестностей изображений  $3 \times 3$  приведен на рис. 5.31. Одно из преимуществ стандартного базиса состоит в том, что он

1	0	0	0	1	0	0	0	1	...	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	0	0	0	0		0	1	0	0	0	1	0	0	1

Девять векторов стандартного базиса для пространства всех матриц  $3 \times 3$ .

$$\begin{array}{|c|c|c|} \hline 9 & 5 & 0 \\ \hline 5 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} = 9 \begin{array}{|c|c|c|} \hline 1 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} + 5 \begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} + 5 \begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 1 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}$$

**Рис. 5.31.** Любую матрицу  $3 \times 3$  можно представить в виде суммы не более чем 9 стандартных матриц, умноженных на соответствующие масштабные множители. (Верхняя строка) девять базисных векторов. (Нижняя строка) пример представления некоторой матрицы в виде комбинации базисных векторов.

позволяет очевидным образом представить любую окрестность изображения. Однако вид разложения по стандартному базису ничего не говорит о двумерной структуре окрестности. Базис Фрея—Чена (Frei—Chen basis), показанный на рис. 5.32, представляет собой множество ортонормированных масок, с помощью которых легко можно интерпретировать структуру пиксельных окрестностей  $3 \times 3$ .

Представление окрестности изображения с помощью базиса Фрея—Чена позволяет интерпретировать энергию изображения, рассматривая ее составляющие, соответствующие маскам градиента, пульсации, линии и т. п. Значение энергетической составляющей, соответствующей некоторому базисному вектору (т. е. маске), будет большим, если структура изображения похожа на эту маску. Структура каждого базисного вектора была разработана специальным образом. Базисные векторы  $\mathbf{W}_1$  и  $\mathbf{W}_2$  похожи на градиентные маски Превитт и Собеля, а базисные векторы  $\mathbf{W}_7$  и  $\mathbf{W}_8$  аналогичны известной маске лапласиана  $3 \times 3$ . Маска линии дает значимый отклик на линии шириной один пиксел, проходящие через окрестность  $3 \times 3$ . Две пульсирующие маски напоминают два перпендикулярных колебания с двумя пиками, двумя впадинами и тремя переходами через ноль. Элементы векторов немного отличаются от элементов масок, которые были ранее спроектированы без учета требования ортогональности базисного множества.

Алгоритм 5.2 формирует бинарное изображение с выделенными структурными элементами полутоновых окрестностей, у которых существенная доля энергии приходится на векторы, принадлежащие заданному подпространству. Например, для обнаружения краев можно было бы выбирать пиксели в соответствии со значениями энергии их окрестностей, приходящейся на базисные векторы  $\mathbf{W}_1, \mathbf{W}_2$ . Для этого в качестве рассматриваемого подпространства можно было бы указать множество  $\mathbf{S} = \{1, 1, 0, 0, 0, 0, 0, 0\}$ . Пример вычислений, в которых выполняется проектирование полутоновой окрестности на базисные векторы Фрея—Чена, приводится далее.

### Пример представления полутоновой окрестности с использованием базиса Фрея—Чена

Рассмотрим полутоновую окрестность  $N =$

10	10	10
10	10	5
10	5	5

Найдем компоненту этого вектора в направлении каждого базисного вектора с использованием, как и ранее, скалярного произведения. Поскольку базис является ортонормированным, то полная энергия изображения равна сумме компонент энергии, приходящихся на каждый базисный вектор. Структуру окрестности  $N$  можно интерпретировать в терминах этих компонент.

$$N \circ \mathbf{W}_1 = \frac{5 + 5\sqrt{2}}{\sqrt{8}} \approx 4.3; \text{ энергия} \approx 18$$

$$N \circ \mathbf{W}_2 = \frac{5 + 5\sqrt{2}}{\sqrt{8}} \approx 4.3; \text{ энергия} \approx 18$$

$$N \circ \mathbf{W}_3 = 0; \text{ энергия} = 0$$

$$N \circ \mathbf{W}_4 = \frac{5\sqrt{2} - 10}{\sqrt{8}} \approx -1; \text{ энергия} \approx 1$$

$$N \circ \mathbf{W}_5 = 0; \text{ энергия} = 0$$

$$N \circ \mathbf{W}_6 = 2.5; \text{ энергия} \approx 6$$

$$N \circ \mathbf{W}_7 = 2.5; \text{ энергия} \approx 6$$

$$N \circ \mathbf{W}_8 = 0; \text{ энергия} = 0$$

$$N \circ \mathbf{W}_9 = 25; \text{ энергия} = 625$$

Полная энергия  $N$  равна  $N \circ N = 675$ , причем энергетическая компонента 625 приходится на средний уровень интенсивности и распределена в направлении вектора  $\mathbf{W}_9$ . Энергия, приходящаяся на остальные базисные компоненты, равна 50, причем 36 (или 72 процента) приходится на градиентные базисные векторы  $\mathbf{W}_1$  и  $\mathbf{W}_2$ . Следовательно, центр окрестности можно было бы отметить в качестве обнаруженного признака в случае, если бы в качестве интересующего подпространства было задано подпространство градиентов.

### Упражнение 5.18

Покажите, что множество из девяти векторов, показанных на рис. 5.32, является ортонормированным множеством.

$$\begin{array}{l} \text{Градиент: } \mathbf{W}_1 = 1/\sqrt{8} \begin{array}{|c|c|c|} \hline 1 & \sqrt{2} & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & - & -1 \\ \hline \end{array} \quad \mathbf{W}_2 = 1/\sqrt{8} \begin{array}{|c|c|c|} \hline 1 & 0 & -1 \\ \hline \sqrt{2} & 0 & - \\ \hline 1 & 0 & -\frac{1}{2} \\ \hline \end{array} \\ \\ \text{Пульсации: } \mathbf{W}_3 = 1/\sqrt{8} \begin{array}{|c|c|c|} \hline 0 & \frac{2}{-1} & \sqrt{2} \\ \hline 1 & 0 & -1 \\ \hline - & 1 & 0 \\ \hline \end{array} \quad \mathbf{W}_4 = 1/\sqrt{8} \begin{array}{|c|c|c|} \hline \sqrt{2} & -1 & 0 \\ \hline -1 & 0 & 1 \\ \hline 0 & 1 & - \\ \hline \end{array} \\ \\ \text{Линия: } \mathbf{W}_5 = 1/2 \begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline -1 & 0 & -1 \\ \hline 0 & 1 & 0 \\ \hline \end{array} \quad \mathbf{W}_6 = 1/2 \begin{array}{|c|c|c|} \hline -1 & 0 & 1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 0 & -1 \\ \hline \end{array} \\ \\ \text{Лапласиан: } \mathbf{W}_7 = 1/6 \begin{array}{|c|c|c|} \hline 1 & -2 & 1 \\ \hline -2 & 4 & -2 \\ \hline 1 & -2 & 1 \\ \hline \end{array} \quad \mathbf{W}_8 = 1/6 \begin{array}{|c|c|c|} \hline -2 & 1 & -2 \\ \hline 1 & 4 & 1 \\ \hline -2 & 1 & -2 \\ \hline \end{array} \\ \\ \text{Константа: } \mathbf{W}_9 = 1/3 \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \end{array}$$

**Рис. 5.32.** Базис Фрея—Чена для представления множества всех изображений  $3 \times 3$  с действительными значениями интенсивности.

### Упражнение 5.19.

(а) Представьте полутоновую окрестность  $\begin{array}{|c|c|c|} \hline 0 & 0 & 1 \\ \hline 0 & 1 & 0 \\ \hline 1 & 0 & 0 \\ \hline \end{array}$  в виде комбинации

базисных векторов, показанных на рис. 5.32. Распределена ли вся энергия между компонентами, соответствующими базисным векторам  $\mathbf{W}_5$  и  $\mathbf{W}_6$ , представляющим линии? (б) Повторите задание (а) для полутоновой окрестности

$$\begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 1 & 0 \\ \hline \end{array}.$$

### Упражнение 5.20.

(а) Представьте полутоновую окрестность  $\begin{array}{|c|c|c|} \hline 10 & 10 & 10 \\ \hline 10 & 20 & 10 \\ \hline 10 & 10 & 10 \\ \hline \end{array}$  в виде комбинации

базисных векторов, показанных на рис. 5.32. Совпадает ли распределение энергии между базисными векторами с вашими ожиданиями? (б) Может ли

### Обнаружение окрестностей, имеющих высокую энергию в указанном подпространстве

$\mathbf{F}[\mathbf{r}, \mathbf{c}]$  — входное полутоновое изображение. Содержимое  $\mathbf{F}$  алгоритмом не изменяется.

$\mathbf{S}$  — битовый вектор такой, что  $\mathbf{S}[\mathbf{j}] = 1$  тогда и только тогда, когда вектор  $\mathbf{W}_j$  включен в интересующее подпространство.

*thresh* — пороговое значение, определяющее требуемую долю энергии.

*noise* — энергетический уровень шума.

$\mathbf{G}[\mathbf{r}, \mathbf{c}]$  — выходное бинарное изображение. Значение  $\mathbf{G}[\mathbf{r}, \mathbf{c}] = 1$  означает, что  $\mathbf{F}[\mathbf{r}, \mathbf{c}]$  имеет энергию в заданном подпространстве  $\mathbf{S}$ , превышающую указанное пороговое значение *thresh*.

**procedure** detect\_neighborhoods( $\mathbf{F}, \mathbf{G}, \mathbf{S}, \textit{thresh}, \textit{noise}$ );

```
{
  for r := 0 to MaxRow - 1
  for c := 0 to MaxCol - 1
    {
      if [r,c] является граничным пикселом then  $\mathbf{G}[\mathbf{r}, \mathbf{c}] := 0$ ;
      else  $\mathbf{G}[\mathbf{r}, \mathbf{c}] := \text{compute\_using\_basis}(\mathbf{F}, r, c, \mathbf{S}, \textit{thresh}, \textit{noise})$ ;
    }
}
```

**procedure** compute\_using\_basis( $\mathbf{IN}, r, c, \textit{thresh}, \textit{noise}$ )

```
{
   $\mathbf{N}[\mathbf{r}, \mathbf{c}]$  представляет окрестность  $3 \times 3$  с центром в пикселе [r,c]
  изображения  $\mathbf{IN}[\mathbf{i}]$ .
  average_energy :=  $\mathbf{N}[\mathbf{r}, \mathbf{c}] \circ \mathbf{W}_9$ ;
  subspace_energy := 0.0;
  for j := 1 to 8
    {
      if ( $\mathbf{S}[\mathbf{j}]$ ) subspace_energy := subspace_energy +  $(\mathbf{N}[\mathbf{r}, \mathbf{c}] \circ \mathbf{W}_j)^2$ ;
    }
  if subspace_energy < noise return 0;
  if subspace_energy / (( $\mathbf{N}[\mathbf{r}, \mathbf{c}] \circ \mathbf{N}[\mathbf{r}, \mathbf{c}]$ ) - average_energy)
  < thresh return 0;
  else return 1;
}
```

**Алгоритм 5.2.** Вычисление изображения  $\mathbf{G}[\mathbf{r}, \mathbf{c}]$ , представляющего результаты выделения признаков на входном изображении  $\mathbf{F}[\mathbf{r}, \mathbf{c}]$  в соответствии с содержимым указанного подпространства  $\mathbf{S}$ .

0	0	0
0	1	0
0	0	0

полутонная окрестность иметь различные варианты интерпрета-

ции? Почему? (в) На какие полутонные окрестности значимые отклики дают только маски  $W_7$  и  $W_8$ ?

### Упражнение 5.21

Напишите программу, реализующую выделение пикселей с использованием базиса Фрея—Чена на основе представленного выше алгоритма. Предусмотрите, чтобы пользователь мог ввести интересующее подпространство  $S$  в виде строки из 9 бит. Пользователь также должен иметь возможность указать уровень энергии шума и пороговое значение энергии, требуемой в указанном подпространстве. Проверьте работу программы на нескольких реальных изображениях, а также на нескольких искусственных изображениях шаблонов (например, таких, которые рассматривались в предыдущих упражнениях).

## 5.10 СВЕРТКА И КРОСС-КОРРЕЛЯЦИЯ\*

В предыдущих разделах было показано, как много полезной информации можно получить в результате сопоставления маски или шаблона с окрестностью изображения. Также отмечалось, что аналогичным образом можно выполнять сглаживание изображений. В данном разделе будут приведены определения двух важных операций — *кросс-корреляции* и *свертки*, которые формализуют подход на основе перемещения маски по изображению и сохранения значений скалярного произведения маски с каждой окрестностью изображения.

### 5.10.1 Определение операций посредством масок

Начнем рассмотрение с переопределения простой операции сглаживания изображения. Представим ее в виде кросс-корреляции изображения со сглаживающей маской. Усредняющий фильтр вычисляет значение пикселя выходного изображения как среднее значение в окрестности соответствующего пикселя входного изображения. Эти вычисления эквивалентны скалярному произведению изображения-шаблона  $m \times n$ , у которого значения пикселей равны весовым коэффициентам  $\frac{1}{mn}$  (на рис. 5.33 этот шаблон показан в виде маски  $3 \times 3$ ). В предположении, что  $m$  и  $n$  имеют нечетные значения и что остаток от деления на 2 игнорируется, уравнение (5.25) определяет скалярное произведение, используемое для вычисления значения выходного пикселя  $G[x, y]$  по окрестности входного пикселя  $F[x, y]$  с применением маски  $H[x, y]$ . При этом начало координат маски  $H$  выбирается в ее центре, так что пиксел  $H(0, 0)$  является центральным пикселем маски. Способ использования  $H$  для вычисления взвешенной суммы пикселей окрестности  $F[x, y]$  очевиден. Заме-



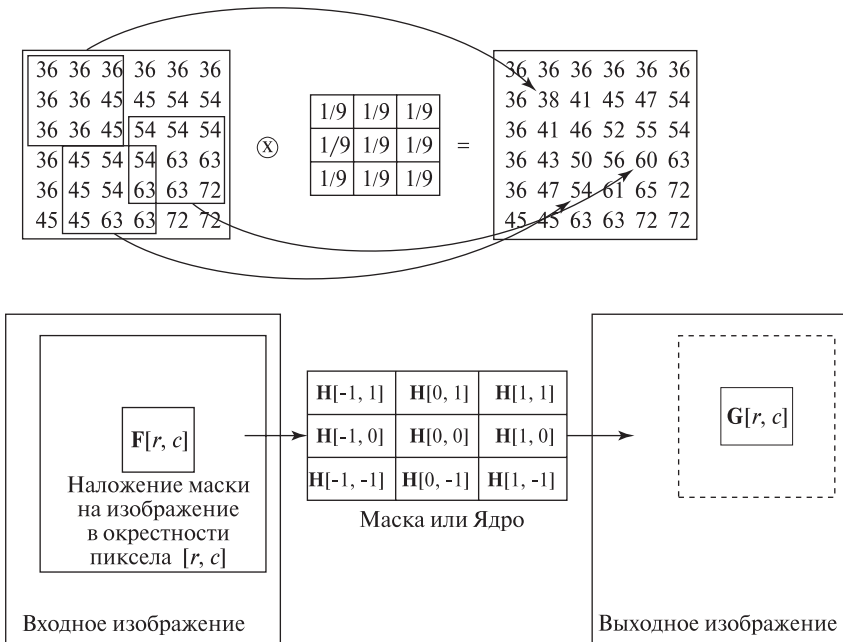
на переменных в (5.25) позволяет получить альтернативную формулировку, которую удобно использовать для вычисления всех выходных пикселей  $\mathbf{G}$ . Это уравнение (5.26), допускающее использование масок  $\mathbf{H}[i, j]$  с четными размерами.

### Упражнение 5.22

Предположим, что изображение  $\mathbf{F}$  содержит пиксеты со значениями 0 и единственный пиксел со значением 1, расположенный в центре изображения. Какое выходное изображение  $\mathbf{G}$  будет получено в результате свертки  $\mathbf{F}$  с усредняющим фильтром  $3 \times 3$ , показанным на рис. 5.33?

### Упражнение 5.23

Спроектируйте одиночную маску для выделения краевых элементов, направленных под углом 30 градусов относительно горизонтальной оси  $X$ . Маска не должна давать существенного отклика на краевые элементы с другими направлениями или на какие-либо иные образы.



**Рис. 5.33.** Сглаживание изображения с использованием усредняющего фильтра  $3 \times 3$  можно представить в форме скалярного произведения каждой окрестности входного изображения и маски, имеющей вид небольшого изображения, пиксеты которого имеют равные значения.

**Упражнение 5.24.** Обнаружение углов

(а) Спроектируйте набор из четырех масок  $5 \times 5$  для выделения углов произвольного прямоугольника, стороны которого параллельны сторонам изображения. Прямоугольник может быть светлее или темнее фона. (б) Являются ли ваши маски ортогональными? (в) Сформируйте процедуру принятия решения для выделения углов и поясните, как она работает.

**57 Определение.** Кросс-корреляция изображения  $F[x, y]$  и маски  $H[x, y]$  определяется выражением

$$G[x, y] = F[x, y] \otimes H[x, y] = \sum_{i=-w/2}^{w/2} \sum_{j=-h/2}^{h/2} F[x+i, y+j]H[i, j] \quad (5.25)$$

См. рис. 5.35.

Сделаем несколько замечаний относительно организации вычислений по этой формуле:

считается, что начало координат маски  $H[x, y]$  располагается в ее центре, так что координаты могут принимать отрицательные значения;

начало координат изображения  $F[x, y]$  не обязательно должно располагаться в его центре;

когда  $H[i]$  частично попадает за границы  $F[i]$ , значения  $G[x, y]$  должны определяться по другой формуле.

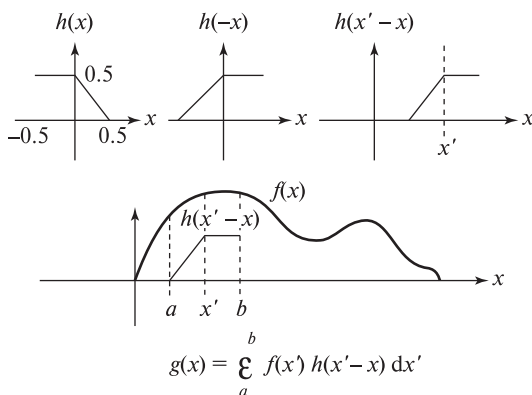
Альтернативная формулировка кросс-корреляции допускает использование маски с нечетными размерами, но в таком случае эту операцию следует рассматривать как преобразование всего изображения, а не как операцию над окрестностью с центром в пикселе  $G[x, y]$ .

$$G[x, y] = \sum_{i=0}^{w-1} \sum_{j=0}^{h-1} F[x+i, y+j]H[i, j] \quad (5.26)$$

**5.10.2 Операция свертки**

**58 Определение.** Свертка функций  $f(x, y)$  и  $h(x, y)$  определяется выражением

$$\begin{aligned} g(x, y) &= f(x, y) \star h(x, y) \\ &\equiv \int_{x'=-\infty}^{+\infty} \int_{y'=-\infty}^{+\infty} f(x', y')h(x-x', y-y') dx' dy' \quad (5.27) \end{aligned}$$



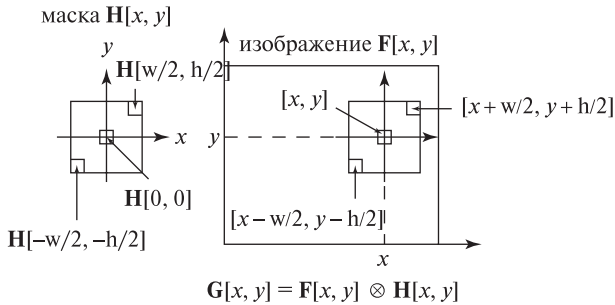
**Рис. 5.34.** Свертка  $g(x)$  сигнала  $f(x)$  с ядром  $h(x)$  обозначается  $g(x) = f(x) \star h(x)$ . Для любой точки  $x$  из области определения, ядро  $h$  сначала отражается, а затем смещается на  $x$ . Потом  $g(x)$  вычисляется путем суммирования всех произведений  $f(x)$  и значений перевернутой и смещенной функции  $h$ .

Свертка тесно связана с кросс-корреляцией и в уравнении (5.27) формально определяется с использованием непрерывных функций интенсивности. Для практического применения и чтобы интегралы имели определенные значения, функции  $f(x, y)$  и  $h(x, y)$ , представляющие двумерные изображения, выбираются такими, чтобы они были равны нулю за пределами конечной прямоугольной области на плоскости  $xy$  и чтобы объемы под определяемыми этими функциями поверхностями имели конечные значения.

Для выполнения фильтрации, функция ядра  $h(x, y)$  часто выбирается равной 0 за пределами некоторого прямоугольника, намного меньшего, чем прямоугольник, на котором ненулевые значения имеет  $f(x, y)$ . Для глобального анализа пространственных частот функции  $f$ , прямоугольник, на котором определена  $h$ , будет включать весь прямоугольник, на котором определена  $f$ . Дополнительные сведения по этому вопросу приводятся в Разделе 5.11, посвященном Фурье-анализу.

На рис. 5.34 поясняются отдельные этапы вычисления свертки  $g(x)$  двух одномерных сигналов. Функция ядра свертки  $h(x)$  сначала отражается относительно начала координат и затем смещается в точку  $x$ , в которой требуется вычислить значение  $g(x)$ . Потом значение  $g(x)$  вычисляется путем интегрирования произведений входной функции  $f(x')$  со смещенным ядром  $h(x' - x)$ . Так как функция ядра равна нулю за пределами интервала  $[a, b]$ , то интегрирование можно ограничить этим конечным интервалом. Свертку цифровых изображений легче выполнить с использованием дискретных сумм произведений, а не непрерывных интегралов из определения свертки.

При вычислении кросс-корреляции (рис. 5.35) маска или ядро смещается в точку изображения  $[x, y]$  в исходном виде, без отражения относительно вертикальной оси. В остальном выполняемые действия совпадают со сверткой.



**Рис. 5.35.** Вычисление кросс-корреляции  $G[x, y]$  изображения  $F[x, y]$  с маской  $H[x, y]$ :  $G[x, y] = F[x, y] \otimes H[x, y]$ . При вычислении  $G[x, y]$  маска  $H[x, y]$  центрируется в точке входного изображения  $F[x, y]$  для суммирования всех произведений значений изображения  $F$  и соответствующих весовых коэффициентов из наложенной маски  $H$ .

Возможно, эту операцию легче представить, так как к ней не производится переворота функции ядра, а только ее смещение в некоторое местоположение на изображении. В случае использования симметричного ядра перевернутое ядро совпадает с исходным, поэтому результаты свертки и корреляции будут одинаковыми. Например, это верно для сглаживающих масок и других изотропных операторов. В то же время многие маски для обнаружения краев являются асимметричными. Несмотря на формальное различие между сверткой и кросс-корреляцией, эти операции похожи, и в области обработки изображений часто обе эти операции называются сверткой. Многие маски, в том числе используемые в данной главе, рассчитаны на непосредственное применение к изображению без переворота.

*Нормированная кросс-корреляция* вычисляется посредством деления кросс-корреляции  $G[x, y]$  на величины  $F[x, y]$  и  $H[x, y]$ , так чтобы результат можно было интерпретировать как меру сходства структуры  $F$  со структурой  $H$ , независимую от масштабных множителей, как было рассмотрено в предыдущих разделах.

### Упражнение 5.25. Обнаружение прямоугольников

С использованием процедуры обнаружения углов из упражнения 5.24 разработайте программу для обнаружения на изображении прямоугольников. (Предполагается, что стороны прямоугольника параллельны сторонам изображения.) На первом шаге надо обнаружить кандидатов на углы прямоугольника. На втором шаге надо выделить подмножества из четырех кандидатов на углы, которые с учетом геометрических ограничений образуют допустимые прямоугольники. На дополнительном третьем шаге может выполняться дальнейшая проверка четырех углов, чтобы убедиться, что интенсивность внутри кандидата-прямоугольника однородна и контрастна с фоном. Какие

5	5	5	5	5	5
5	5	5	5	5	5
5	5	10	10	10	10
5	5	10	10	10	10
5	5	5	10	10	10
5	5	5	5	10	10

 $\cdot$ 

0	-1	0
-1	4	-1
0	-1	0

 $=$ 

0	0	0	0	0	0
0	0	-5	-5	-5	0
0	-5	10	5	5	0
0	-5	10	0	0	0
0	0	-10	10	0	0
0	0	0	0	0	0

Кросс-корреляция с ЛОГ-маской содержит точки перехода через ноль на границах

5	5	5	5	5	5
5	5	5	5	5	5
5	5	10	10	10	10
5	5	10	10	10	10
5	5	5	10	10	10
5	5	5	5	10	10

 $\cdot$ 

-1	0	11
-1	0	11
-1	0	11

 $=$ 

0	0	0	0	0	0
0	5	5	0	0	0
0	10	10	0	0	0
0	10	15	5	0	0
0	5	10	10	5	0
0	0	0	0	0	0

Кросс-корреляция с маской производной в направлении строк выделяет границы вдоль столбцов

5	5	5	5	5	5
5	5	5	5	5	5
5	5	10	10	10	10
5	5	10	10	10	10
5	5	5	10	10	10
5	5	5	5	10	10

 $\cdot$ 

+1	+1	+1
0	0	0
-1	-1	-1

 $=$ 

0	0	0	0	0	0
0	-5	-10	-15	-15	0
0	-5	-10	-15	-15	0
0	5	5	5	0	0
0	5	10	10	5	0
0	0	0	0	0	0

Кросс-корреляция с маской производной в направлении столбцов выделяет границы вдоль строк

**Рис. 5.36.** Кросс-корреляция изображения с различными масками выделяет границы областей. (*Сверху*) оператор второй производной формирует точки перехода через 0 на границах областей. (*В центре*) оператор производной в направлении строк ( $x$ ) выделяет изменения вдоль столбцов. (*Внизу*) оператор производной в направлении столбцов ( $y$ ) выделяет изменения вдоль строк.

результаты можно ожидать от вашей программы в случае обработки зашумленного изображения шахматной доски? Проверьте программу на подобном изображении шахматной доски (например, как на рис. 5.7) и на изображении здания с прямоугольными окнами (например, как на рис. 5.42).

### Упражнение 5.26

Даны маска  $\mathbf{H} =$ 

1	0	-3
0	4	0
-3	0	1

и изображение  $\mathbf{F} =$

5	5	5	5	5	5
5	5	5	5	8	8
5	5	5	8	8	8
5	5	8	8	8	8
8	8	8	8	8	8

Вычислите их кросс-корреляцию  $\mathbf{G} = \mathbf{F} \otimes \mathbf{H}$ .

### Упражнение 5.27. Размытие точки

Дано ядро  $\mathbf{H} =$ 

1	2	1
2	5	2
1	2	1

Каков будет результат свертки с этим ядром изображения  $\mathbf{F}[x, y]$ , у которого  $\mathbf{F}[x_0, y_0] = 1$ , а все остальные пиксели равны 0?

### Упражнение 5.28

Предположим, что функция  $h(x)$  принимает значение 1 при  $-1/2 \leq x \leq 1/2$  и нулевое значение за пределами этого диапазона, и что функция  $f(x)$  принимает значение 1 при  $10 \leq x \leq 20$  и равна нулю в других областях. (а) Постройте графики функций  $f$  и  $h$ . (б) Вычислите и постройте график функции  $g(x) = f(x) \star h(x)$ . (в) Вычислите и постройте график функции  $g(x) = h(x) \star h(x)$ .

### 5.10.3 Возможности параллельной реализации

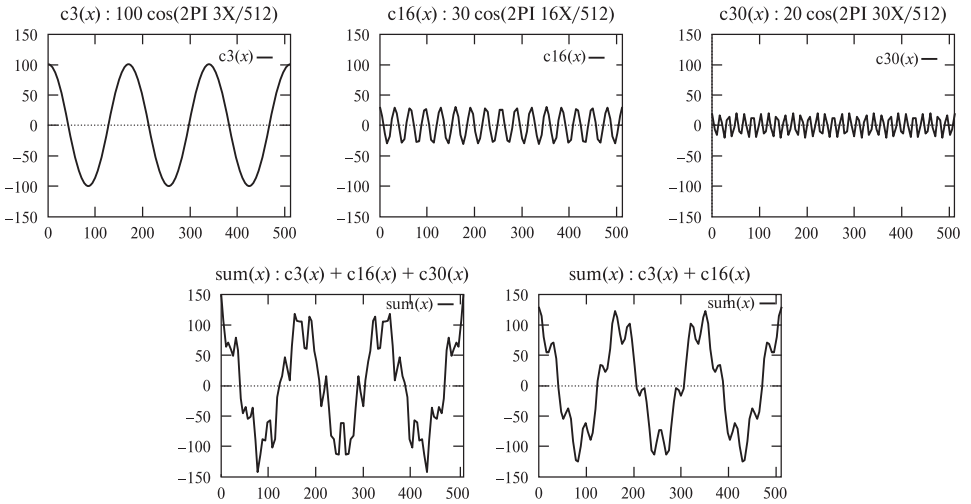
Из определения свертки видно, что  $g(x_1, y_1)$  можно вычислить независимо от  $g(x_2, y_2)$ . В действительности все операции интегрирования можно выполнять параллельно. Кроме того, при вычислении каждого интеграла для каждого отдельного значения  $g(x, y)$  можно одновременно вычислять все произведения, что делает возможным высокопараллельную обработку. Для параллельного выполнения всех или некоторых операций могут применяться различные компьютерные архитектуры.

## 5.11 АНАЛИЗ ПРОСТРАНСТВЕННЫХ ЧАСТОТ С ИСПОЛЬЗОВАНИЕМ ГАРМОНИЧЕСКИХ ФУНКЦИЙ\*

Очень важное место в области обработки сигналов занимают преобразования Фурье. Описание теоретических вопросов Фурье-анализа и методов его практического применения может занять много книг. В данном разделе будет приведено только краткое рассмотрение, в котором будут использоваться введенные ранее обозначения векторного пространства.

Французский математик Фурье представлял поверхность моря в виде суммы гармонических (синусоидальных) волн. Большие волны, вызываемые приливом или кораблями, характеризуются большими значениями длины волны (низкими частотами), а меньшие волны, вызываемые ветром, падающими предметами и т. п., имеют более короткие длины (высокие частоты). На рис. 5.37 в верхней строке представлены три гармонических колебания, содержащие 3, 16 и 30 периодов в диапазоне одномерного пространства  $x \in [0, 512]$ . В нижней строке приведены две функции, одна из которых является суммой всех трех показанных колебаний, а другая — суммой только двух первых. Похожие наборы колебаний можно использовать для формирования двумерных функций интенсивности, представляющих изображения, или даже трехмерных функций плотности.

Теория рядов Фурье в математическом анализе показывает, как большинство действительных поверхностей и функций можно представить с помощью базиса гармонических функций. Энергию, соответствующую базисным векторам, можно интерпретировать в терминах структуры представляемой



**Рис. 5.37.** (Верхняя строка) три гармонические функции:  $100 \cos(2\pi \frac{3x}{512})$ ,  $30 \cos(2\pi \frac{16x}{512})$  и  $20 \cos(2\pi \frac{30x}{512})$ . (Нижняя строка) сумма трех и сумма первых двух колебаний, показанных в верхней строке.

поверхности (функции). Этот подход оказывается наиболее полезным, когда на больших областях поверхности содержатся повторяющиеся шаблоны. Например, это могут быть прямоугольные дома на изображении аэрофотосъемки города, волны на обширной поверхности воды, текстура лесного массива или сельскохозяйственного поля. Этот подход может применяться для обработки всего изображения или его отдельных фрагментов. Он заключается в использовании базиса Фурье для последующей фильтрации изображения или для принятия решений относительно изображения на основе распределения энергии между различными базисными векторами.

Например, для удаления высокочастотного шума можно устранить из изображения все компоненты, соответствующие высокочастотным гармоническим колебаниям. Или (что то же самое), можно восстановить исходное пространственное изображение, учитывая только низкочастотные колебания и игнорируя высокочастотные.

### Упражнение 5.29

В живописи существует направление *пуантилизм*, характерное тем, что художники рисуют картины мазками точечной формы. При нанесении мазка кисть удерживается перпендикулярно холсту. Каждое касание кисти оставляет точечный след, в некотором смысле подобный одному пикселу цифрового

изображения<sup>1</sup>. Человек, рассматривающий картину с некоторого расстояния, способен воспринимать непрерывное изображение. Разработайте программу для интерактивного формирования изображений в стиле пуантилизма.

В программе должна быть предусмотрена возможность выбора цвета рисования из палитры и возможности изменения некоторых других параметров, например, размера кисти или включение/выключение режимов применения логических операций ИЛИ и ИСКЛЮЧАЮЩЕЕ ИЛИ при рисовании точек и т. п. С помощью этой программы нарисуйте изображение звездной ночи. Программа должна позволять работать с внешними файлами, чтобы можно было сохранить изображение и в дальнейшем продолжить работу с ним.

---

### Упражнение 5.30

Напишите программу для свертки маски с изображением. Программа должна загружать изображение и маску из входных файлов одинакового формата. Возможно, вы сможете проверить работу программы на изображении, полученном с помощью программы из предыдущего упражнения.

---

### Упражнение 5.31

Предположим, что в рамках программы поиска внеземных цивилизаций (SETI) выполняется анализ сигналов, приходящих из далекого космоса. Допустим, что сигнал  $\mathbf{S}$  является последовательным соединением первых 100 простых чисел в двоичной записи, а сигнал  $\mathbf{R}$  — это некоторый принятый из космоса сигнал, длина которого намного превышает длину  $\mathbf{S}$ . Будем полагать, что в сигнале  $\mathbf{R}$  присутствует шум и что он представлен в виде последовательности действительных чисел. Можно ли применить кросс-корреляцию или нормированную кросс-корреляцию для определения присутствия сигнала  $\mathbf{S}$  внутри  $\mathbf{R}$ ? Почему?

#### 5.11.1 Базис Фурье

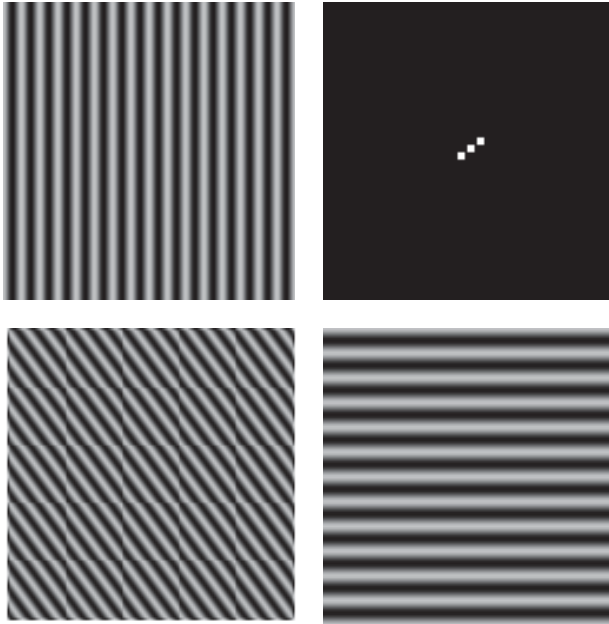
Будем полагать, что имеется ортонормированное множество гармонических базисных изображений (или функций интенсивности)  $E_k \approx E_{u,v}(x, y)$ . Здесь  $k$ ,  $u$  и  $v$  являются целыми числами, определяющими конечное множество базисных векторов. Вскоре станет очевидно, как параметры  $u$  и  $v$  определяют базисные векторы, но пока будем пользоваться только индексом  $k$ , чтобы сосредоточиться на рассмотрении основных понятий. В нижней строке на рис. 5.37 представлены два сигнала, полученные сложением трех и двух косинусоидальных колебаний, показанных в верхней строке рисунка. Если

---

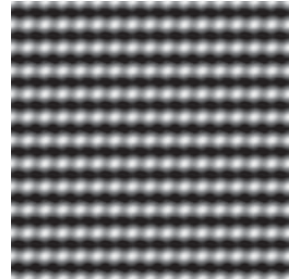
<sup>1</sup> Расстояние между центрами соседних точечных мазков может быть произвольным. — *Прим. перев.*



использовать большее количество косинусоидальных колебаний с различными частотами, то можно построить более сложные функции. На рис. 5.39 приведен результат сложения трех колебаний, показанных на рис. 5.38, взятых с различными масштабными множителями (амплитудами). В результате была получена новая функция интенсивности. С использованием базисных функций Фурье  $E_k$  любую функцию интенсивности можно представить в виде  $I[x, y] = \sum_{k=0}^{N-1} a_k E_k[x, y]$ . Как и в предыдущих разделах, каждый коэффициент  $a_k$  можно рассматривать как меру сходства между  $I[x, y]$  и  $E_k[x, y]$  или как характеристику части энергии изображения  $I[x, y]$ , приходящейся на колебательную составляющую определенной частоты. Ряд полезных операций обработки изображений можно проделать, выполняя действия со значениями  $a_k$ , а не со значениями интенсивности  $I[x, y]$ . Три операции такого типа описываются далее.



**Рис. 5.38.** Различные гармонические функции интенсивности с пространственной областью определения  $[x, y]$ . Верхнее левое изображение получено с помощью формулы  $100 \cos(2\pi(16x/512)) + 100$  и содержит 16 периодов в направлении оси  $x$ . Нижнее правое изображение было сформировано с помощью формулы  $100 \cos(2\pi(12y/512)) + 100$  и содержит 12 периодов в направлении оси  $y$ . Нижнее левое изображение было сформировано с использованием формулы  $100 \cos(2\pi(16x/512 + 12y/512)) + 100$ : обратите внимание, как колебания на нижнем левом изображении совмещены с колебаниями на верхнем левом и на нижнем правом изображениях. На верхнем правом изображении показан спектр мощности преобразования Фурье.



**Рис. 5.39.** Функция интенсивности, сформированная в виде суммы  $I[x, y] = 100E_i + 30E_j + 10E_k$  где  $E_i$  соответствует нижнему правому изображению на рис. 5.38,  $E_j$  — верхнему левому, а  $E_k$  — нижнему левому. (Возможно, это изображение напоминает фруктовый сад или пузырьки пены?)

### Важные операции обработки изображения с использованием базиса Фурье

1. Базис Фурье можно использовать для **удаления высокочастотного шума** из изображения или сигнала. Сигнал  $f$  представляется в виде  $\sum_k a_k E_k$ . Затем коэффициенты  $a_k$  для колебаний с высокими частотами  $E_k$  приравниваются 0. Новый сигнал  $\hat{f}$  вычисляется сложением оставшихся базисных функций с коэффициентами  $a_k \neq 0$ .
2. Базис Фурье может применяться для **выделения текстурных признаков**, например, для классификации типа объекта в области изображения. После представления изображения, или некоторой области изображения, с помощью базиса Фурье различные коэффициенты  $a_k$  могут использоваться для вычисления признаков, необходимых для принятия классификационного решения. Например, подобным образом могут обрабатываться области изображений с волнами на поверхности воды или сельскохозяйственными посевами, образующими строчные структуры. Коэффициенты  $a_k$  полезны как для определения частоты, так и для определения направления текстурированных областей.
3. Базис Фурье также может использоваться для **сжатия изображений**. Отправитель может послать адресату подмножество  $a_k$ , и адресат после получения этих данных сможет восстановить приближенное изображение путем сложения гармонических колебаний с известными частотами и полученными амплитудами. При необходимости значения амплитуд  $a_k$  могут передаваться по порядку возрастания энергии или в порядке роста частоты. С учетом содержания данных, полученных к текущему моменту, получатель может в любой момент инициировать прекращение передачи.

Сейчас наша цель состоит в формировании базиса, полезного для представления функций интенсивности, и в пояснении способов его практического использования. Для этого необходимы некоторые предварительные знания из математического анализа. Применительно к рассматриваемому материалу, будем полагать, что начало системы координат располагается в центре функции интенсивности, которая определена на квадратной области плоскости  $xy$ .

В дискретном случае, будем считать что изображение представлено в виде массива  $I[x, y]$  с  $N^2$  выборочными значениями интенсивности.

Во-первых, получим множество гармонических функций различной частоты в качестве ортогонального базиса для представления непрерывных сигналов  $f$ . Если  $m$  и  $n$  являются двумя любыми различными целыми числами, то два гармонических колебания с этими частотами являются ортогональными на интервале  $[-\pi, \pi]$ . Читатель может обратиться к приведенным далее упражнениям, чтобы убедиться, что множество функций  $\{1, \sin(mx), \cos(nx), \dots\}$  является ортогональным множеством функций на интервале  $[-\pi, \pi]$ . Ортогональность косинусоидальных колебаний следует из выражения (5.28), так как  $\sin(k\pi) = 0$  для всех целых чисел  $k$ .

$$\begin{aligned} \int_{-\pi}^{\pi} \cos(m\theta) \cos(n\theta) d\theta &= \frac{\sin(m-n)\pi}{2(m-n)} + \frac{\sin(m+n)(-\pi)}{2(m+n)} \\ &= 0 \quad m^2 \neq n^2 \end{aligned} \quad (5.28)$$

$N$  выборочных значений этих косинусоидальных функций, взятых с равными промежутками, будут образовывать множество векторов. Это множество векторов является ортогональным в соответствии с приведенным выше определением на основе скалярного произведения.

### 5.11.2 Двумерные функции интенсивности

**59 Определение.** Комплекснозначная функция интенсивности имеет вид

$$\begin{aligned} E_{u,v}(\mathbf{x}, \mathbf{y}) &\equiv e^{-j 2\pi(ux+vy)} \\ &= \cos(2\pi(ux + vy)) - j \sin(2\pi(ux + vy)) \end{aligned} \quad (5.29)$$

где  $u$  и  $v$  являются параметрами пространственной частоты (см. рис. 5.38), а  $j = \sqrt{-1}$ .

---

#### Упражнение 5.32

Рассмотрим множество всех непрерывных функций  $f$ , определенных на интервале  $x \in [x_1, x_2]$ . Покажите, что это множество функций  $f, g, h, \dots$ , вместе с действительными числами  $a, b, c, \dots$  образует векторное пространство. Для этого убедитесь в выполнении следующих свойств:

$$\begin{aligned} f \oplus g &= g \oplus f & (f \oplus g) \oplus h &= f \oplus (g \oplus h) \\ c(f \oplus g) &= cf \oplus cg & (a + b)f &= af \oplus bf \\ (ab)f &= a(bf) & 1f &= f \\ 0f &= \mathbf{0} \end{aligned}$$

**Упражнение 5.33**

В пространстве непрерывных функций с областью определения  $x \in [x_1, x_2]$ , которое рассматривалось в предыдущем упражнении, определим скалярное произведение и соответствующую норму:

$$f \circ g = \int_a^b f(x)g(x) dx; \quad \|f\| = \sqrt{f \circ f} \quad (5.30)$$

Докажите для этого векторного пространства выполнимость следующих свойств скалярного произведения.

$$\begin{aligned} (f \oplus g) \circ h &= (f \circ g) + (g \circ h) \\ f \circ f &\geq 0 \\ f \circ f = 0 &\iff f = \mathbf{0} \\ f \circ g &= g \circ f \\ (cf) \circ g &= c(f \circ g) \end{aligned}$$

**Упражнение 5.34.** Четные и нечетные функции

Функция называется *четной функцией*, если  $f(-x) = f(x)$ , и *нечетной*, если  $f(-x) = -f(x)$ . (а) Покажите, что  $\cos(mx)$  является четной функцией, а  $\sin(nx)$  — нечетной. Здесь  $m$  и  $n$  являются ненулевыми целыми числами. (б) Пусть  $f$  и  $g$  являются, соответственно, четной и нечетной функциями, заданными на интервале  $[-L, L]$ . Докажите, что  $\int_{-L}^L f(x)g(x)dx = 0$ .

**Упражнение 5.35**

С учетом определения скалярного произведения из упражнения 5.33 покажите, что заданное далее множество синусоидальных функций  $f_k$  является ортогональным на интервале  $[-\pi, \pi]$ .

$$f_0(x) = 1; \quad f_1(x) = \sin(x); \quad f_2(x) = \cos(x); \quad f_3(x) = \sin(2x); \quad f_4(x) = \cos(2x); \\ f_5(x) = \sin(3x); \quad f_6(x) = \cos(3x); \dots$$

Применение комплексных чисел удобно тем, что позволяет по отдельности учитывать косинусоидальные и синусоидальные колебания одинаковой частоты. Синусоидальное колебание имеет ту же структуру, что и косинусоидальное, только сдвинуто по фазе на  $1/4$  длины волны относительно косинусоидального колебания.

Когда одна из этих базисных функций сильно коррелирует с функцией интенсивности, то это значит, что функция интенсивности имеет большую составляющую энергии на частоте  $u, v$ . *Преобразование Фурье* преобразует

функцию интенсивности в массив таких значений корреляции. Сначала рассмотрим преобразование Фурье в интегральной форме, а затем перейдем к дискретным суммам, пригодным для обработки цифровых изображений.

**60 Определение.** Двумерное преобразование Фурье преобразует функцию пространственных координат  $f(x, y)$  в область частот  $u, v$ .

$$\begin{aligned} F(u, v) &\equiv \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) E_{u,v}(x, y) dx dy \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j 2\pi(ux+vy)} dx dy \end{aligned} \quad (5.31)$$

Если функция  $f$  обладает перечисленными далее свойствами, необходимыми для применения приведенной формулы, то будем называть эту функцию «хорошей». Интеграл  $\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} |f(x, y)| dx dy$  должен быть конечным. Функция  $f(x, y)$  должна быть равна 0 за пределами некоторого прямоугольника  $R$ , так что бесконечные пределы интегрирования могут быть заменены конечными пределами, соответствующими границам области  $R$ . Также  $f$  на области  $R$  должна иметь конечное количество точек экстремума и не должна содержать бесконечных разрывов.

---

### Упражнение 5.36

Каков специальный смысл  $\mathbf{F}(0, 0)$ , где  $\mathbf{F}(u, v)$  обозначает преобразование Фурье функции интенсивности  $f(x, y)$ ?

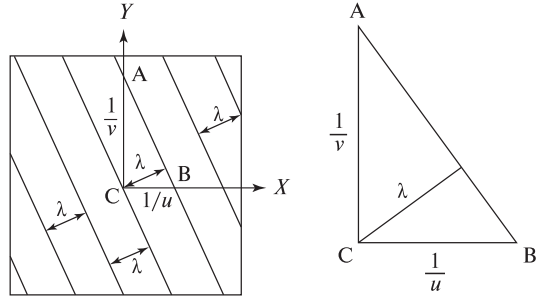
Часто требуется работать со спектром мощности, в котором сочетается энергия, приходящаяся на синусоидальные и косинусоидальные колебания одной и той же частотной компоненты  $u, v$ . Пример спектра мощности показан на верхнем правом изображении на рис. 5.38.

**61 Определение.** Спектр мощности преобразования Фурье вычисляется по формуле:

$$P(u, v) \equiv (\operatorname{Re}(F(u, v))^2 + \operatorname{Im}(F(u, v))^2)^{1/2} \quad (5.32)$$

На рис. 5.40 показано, как длина волны двумерной синусоиды связана с длинами волн, спроектированными на каждую координатную ось. Параметр  $u$  представляет собой частоту в направлении оси  $X$  (в периодах на единицу длины), а величина  $1/u$  равна длине волны. Параметр  $v$  является частотой в направлении оси  $Y$ , а величина  $1/v$  — длиной волны. Величина  $\lambda$  равна длине волны синусоиды в направлении ее естественной оси, или, направления распространения волны. Вычисляя площадь треугольника, показанного на рис. 5.40 (справа), двумя различными способами, можно получить следующую

**Рис. 5.40.** Взаимосвязь между длинами волн  $1/u_0$  и  $1/v_0$  синусоиды в проекции на оси  $X$  и  $Y$  и длиной волны  $\lambda$  двумерной волны.



формулу, которая поможет интерпретировать спектр мощности с точки зрения анализа частотного содержания исходного изображения.

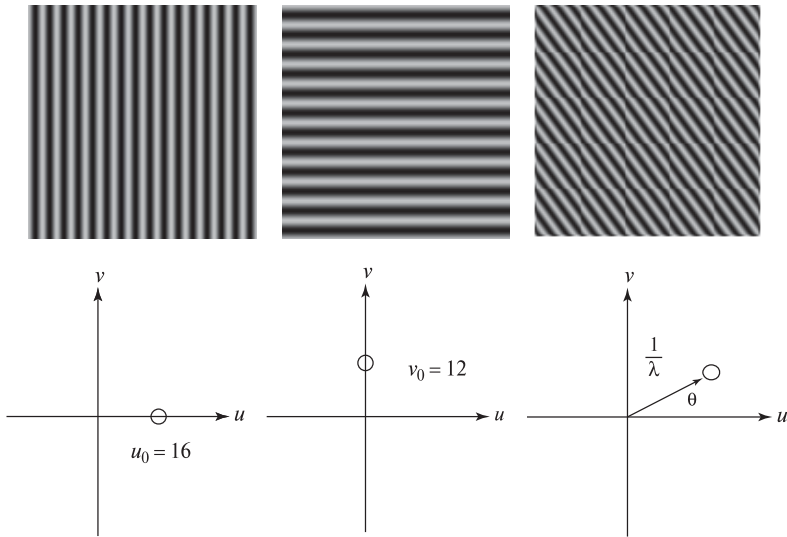
$$\lambda \sqrt{(1/u)^2 + (1/v)^2} = (1/u)(1/v)$$

$$\lambda = \frac{1}{\sqrt{u^2 + v^2}} \quad (5.33)$$

Будем полагать, что ширина и высота рассматриваемого изображения равны 1. На рис. 5.38 имеется  $u = 16$  периодов на ширине изображения, так что длина волны  $1/u = 1/16$ . Аналогично, получим  $1/v = 1/12$ . Применяя формулу (5.33), получим  $\lambda = 1/20$ . Подсчитывая количество колебаний на нижнем левом изображении на рис. 5.38, вдоль диагонали длиной 1.4 можно увидеть 27 колебаний. Это дает ожидаемую частоту  $27/1.4 \approx 20$  вдоль естественного направления двумерной волны.

На рис. 5.41 показан главный отклик в спектре мощности трех гармонических функций интенсивности, представленных ранее на рис. 5.38. Спектр мощности в верхней правой части на рис. 5.38 в действительности показывает существенные отклики не в одной, а в трех точках. Во-первых, отметим, что  $F(0, 0)$  представляет просто полную энергию функции  $f(x, y)$ . Так как среднее значение каждой гармонической функции на рис. 5.38 равно 100, а не 0, то эти функции имеют существенную среднюю энергию на нулевой частоте. Во-вторых, из определения следует, что  $P(-u, -v) = P(u, v)$ , так что спектр мощности симметричен относительно начала координат  $u = 0, v = 0$ . На рис. 5.42 показаны спектры мощности для четырех естественных изображений.

Спектр мощности можно рассматривать как двумерное графическое представление мощности исходного изображения в зависимости от частотных параметров  $u$  и  $v$ . В действительности, это преобразование можно выполнить с помощью оптической системы и получить результат в виде физического изображения. В гл. 2 кратко рассматривался датчик, у которого массив элементов был разделен на сектора и кольца (конфигурация ROSA на рис. 2.4 (с) гл. 2). Так как спектр мощности симметричен относительно поворота на  $\pi$ , как показано на рис. 5.42, то сектора можно использовать для оценки мощности в определенном направлении, а кольца — для выборки частотных полос безотносительно к направлению. Такая выборка может также выполняться



**Рис. 5.41.** (Верхняя строка) три гармонические функции интенсивности и их главный отклик в спектре мощности (нижняя строка).

программным образом. В любом случае, если для выборки используются  $n_r$  колец и  $n_s$  секторов, то можно получить  $n_r + n_s$  признаков, которые могут применяться для классификации той окрестности изображения, для которой они были получены.

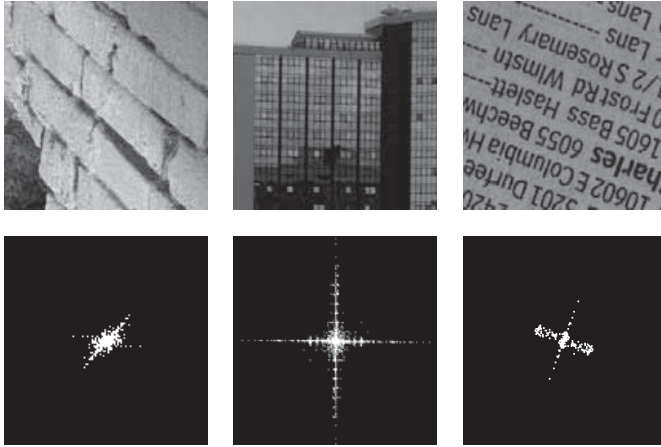
### 5.11.3 Дискретное преобразование Фурье

Дискретное преобразование Фурье (ДПФ), которое непосредственно применимо к цифровым изображениям, определяется выражением (5.34). Как отмечалось выше, базис множества изображений  $N \times N$  с действительными значениями интенсивности должен содержать  $N^2$  базисных векторов. Каждый из них определяется парой параметров частоты  $u, v$ , которые лежат в диапазоне от 0 до  $N - 1$  (эти диапазоны используются в приведенных далее выражениях).

**62 Определение.** Дискретное преобразование Фурье (ДПФ) преобразует изображение, состоящее из  $N \times N$  пространственных выборочных значений  $\mathbf{I}[x, y]$ , в массив  $\mathbf{F}[u, v]$  из  $N \times N$  коэффициентов, образующих частотное представление этого изображения.

$$F[u, v] \equiv \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} I[x, y] e^{-\frac{2\pi}{N} j (xu+yv)} \quad (5.34)$$

Для вычисления одного элемента (*пиксела*) в частотной области  $\mathbf{F}[u, v]$  надо просто вычислить скалярное произведение всего изображения  $\mathbf{I}[x, y]$  и маски  $E_{u,v}[x, y]$ . Эта маска в явном представлении обычно не применяется, а при



**Рис. 5.42.** Три изображения (наверху) и их спектры мощности (внизу). Спектр мощности кирпичной текстуры показывает, что энергия распределена по многим гармоническим функциям на многих частотах, но доминирующее направление перпендикулярно к 6 темным швам, расположенным под углом примерно 45 градусов относительно оси  $X$ . Существенная энергия наблюдается в направлении 90 градусов относительно оси  $X$ , что объясняется присутствием нескольких коротких вертикальных швов. Спектр мощности изображения здания показывает наличие энергии на высоких частотах в направлении осей  $X$  и  $Y$ . Справа показано изображение страниц из телефонного справочника. В спектре мощности есть компоненты на высокой частоте в направлении примерно 60 градусов относительно оси  $X$ . Они соответствуют текстуре строк текста. Энергия более широко распределена в перпендикулярном направлении, что соответствует расположению символов и промежутков между ними. (С любезного разрешения MIT Media Lab VisTex и Ida Stockman)

необходимости вычисляется неявно в зависимости от  $u, v$  и с использованием функций  $\cos$  и  $\sin$ . Также определим обратное преобразование, преобразующее представление  $\mathbf{F}[u, v]$  из частотной области обратно в изображение  $\mathbf{I}[x, y]$  в пространственной области. Хотя иногда может оказаться полезным представить преобразование  $\mathbf{F}$  в виде двумерного изображения, следует отметить, что в действительности это НЕ изображение. Для обозначения  $\mathbf{F}[u, v]$  будем использовать термин *частотное представление*.

**63 Определение.** Обратное дискретное преобразование Фурье (ОДПФ) преобразует частотное представление  $\mathbf{F}[u, v]$ , имеющее размеры  $N \times N$ , в изображение  $\mathbf{I}[x, y]$  с такими же размерами  $N \times N$ , состоящее из выборочных значений интенсивности в пространственной области.

$$I[x, y] \equiv \frac{1}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F[u, v] e^{+2\pi \frac{j}{N}(ux+vy)} \quad (5.35)$$



Если  $\mathbf{F}[u, v]$  было вычислено как преобразование Фурье  $\mathbf{I}[x, y]$ , то может потребоваться выполнить обратное преобразование, чтобы вернуться к исходному изображению. Прямое и обратное преобразование Фурье представлены в приведенных выше определениях. Доказательство обратимости преобразования оставлено для упражнений. Сначала остановимся на обсуждении практического использования ДПФ и ОДПФ. Изображение может потребоваться преобразовать в частотное представление для хранения или для передачи по телекоммуникационным каналам. Для восстановления исходного изображения потребуется применение обратного преобразования. В обработке изображений некоторые операции повышения качества выполняются над частотным представлением, а затем с помощью обратного преобразования производится возврат к изображению в пространственной области. Например, высокие частоты можно уменьшить или даже удалить путем уменьшения или обнуления тех элементов  $\mathbf{F}[u, v]$ , которые соответствуют высокочастотным колебаниям. Теорема о свертке в Разделе 5.11 позволяет дать элегантную теоретическую интерпретацию этому интуитивно понятному процессу.

---

### Упражнение 5.37. Некоторые важные свойства комплексных чисел

Для выполнения этого упражнения воспользуйтесь определением  $e^{j\omega} = \cos \omega + j \sin \omega$ . (а) Покажите, что  $(e^{j\omega})^n = \cos(n\omega) + j \sin(n\omega)$ . (б) Покажите, что  $x = e^{j\frac{2\pi k}{N}}$  является решением уравнения  $x^N - 1 = 0$  для  $k = 0, 1, \dots, N - 1$ . (в) Пусть  $x_0 = 1 = e^{j\frac{2\pi \cdot 0}{N}}, \dots, x_k = e^{j\frac{2\pi k}{N}}$  представляют  $N$  корней уравнения  $x^N - 1 = 0$ . Покажите, что  $x_1 + x_2 + x_3 + \dots + x_{N-1} = 0$ .

---

### Упражнение 5.38. Доказательство обратимости преобразований ДПФ/ОДПФ

Требуется доказать, что подстановка  $\mathbf{F}[u, v]$  из уравнения (5.34) в уравнение (5.35) приводит к точному исходному значению  $\mathbf{I}[x, y]$ . Рассмотрим следующую сумму, в которой  $x, y, s, t$  являются целочисленными параметрами в диапазоне  $[0, N - 1]$ , используемом в определениях преобразований Фурье:

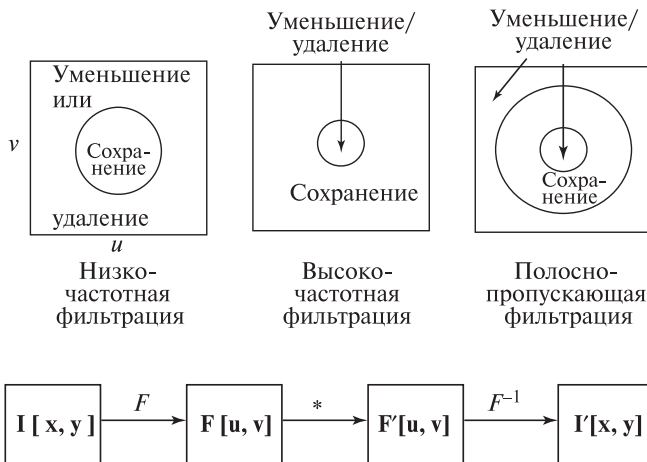
$$G(x, y, s, t) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} e^{j\frac{2\pi}{N}((x-s)u + (y-t)v)}.$$

(а) Покажите, что если  $s = x$  и  $t = y$ , то  $G(x, y, s, t) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} 1 = N^2$ . (б) Покажите, что если  $s \neq x$  или  $t \neq y$ , то  $G(x, y, s, t) = 0$ . (в) Теперь докажите основное утверждение о том, что применение обратного преобразования к результату преобразования Фурье приводит к получению исходного изображения.

### 5.11.4 Полоснопропускающая фильтрация

*Полоснопропускающая фильтрация* (bandpass filtering) является распространенной операцией обработки изображений, выполняемой в частотной области. Эта операция поясняется на рис. 5.43. ДПФ используется для преобразования изображения в его частотное представление. В этом представлении коэффициенты для некоторых частот уменьшаются (возможно, до нуля), а другие остаются неизменными. Схема *низкочастотного фильтра* приведена на рис. 5.43, слева. Как можно было предположить из интуитивных соображений, коэффициенты высокочастотных компонент удаляются и затем измененное частотное представление с помощью обратного преобразования (5.35) преобразуется для получения сглаженной версии исходного изображения.

Вместо *удаления* элементов из частотного представления можно взять скалярное произведение  $\mathbf{F}[u, v]$  с двумерным гауссианом, что позволит вычислить взвешенное среднее от низкочастотных и высокочастотных компонент. В результате коэффициенты низкочастотных компонент будут увеличены, а высокочастотных — уменьшены. На рис. 5.43 показаны способы изменения частотного представления при высокочастотной и полоснопропускающей фильтрации. Теоретическое рассмотрение свертки в Разделе 5.11.6 обеспечит более ясное понимание этих операций.



**Рис. 5.43.** Полоснопропускающая фильтрация может быть выполнена путем преобразования Фурье в частотную область (для получения  $\mathbf{F}[u, v]$ ) и последующего умножения (\*) на полоснопропускающий фильтр. Это умножение может присвоить коэффициентам  $u, v$  различных частот нулевые значения, как показано в верхней строке. Модифицированное частотное представление затем преобразуется обратно в пространственную область для получения отфильтрованного изображения  $\mathbf{I}'[x, y]$ .

### 5.11.5 Обсуждение преобразования Фурье

*Быстрое преобразование Фурье* сокращает время вычислений за счет однократного выполнения общих операций для различных пар  $u, v$ . Это преобразование обычно применяется для обработки квадратных изображений размерами  $2^m \times 2^m$ . Несмотря на широкое применение в области обработки изображений, преобразование Фурье способно привести к нежелательному подавлению локальных признаков изображений, так как оно является *глобальным преобразованием* и для вычисления каждого значения  $\mathbf{F}[u, v]$  использует все пиксели входного изображения. Например, высокочастотные компоненты должны использоваться для представления волос, травы или каких-либо других мелких деталей. Во-первых, такие компоненты могут быть отфильтрованы как шумовые. Но даже если они не были отфильтрованы, то высокочастотный отклик  $\mathbf{F}[u_h, v_h]$  был бы вычислен в виде скалярного произведения высокочастотного колебания со всем изображением. Если область волос или травы занимает малую область всего изображения, то значение скалярного произведения, большое или малое, будет зависеть от содержания остальной части изображения.

В последние десять лет большое внимание уделяется возможности использования *вейвлетов* вместо колебаний размерами с целое изображение. Вейвлеты обладают большей чувствительностью к локальным изменениям изображения, но вместе с тем имеют ряд преимуществ глобальных гармонических функций. Файловый формат JPEG и некоторые другие методы сжатия изображений основаны на использовании дискретного косинусного преобразования фрагментов изображений с целью уменьшения размера данных изображения. Иногда такие методы сжатия должны отключаться для сохранения важных локальных деталей изображения.

### 5.11.6 Теорема о свертке\*

В данном разделе в упрощенном виде приводится доказательство важной *теоремы о свертке*. Согласно этой теореме, свертка двух функций в пространственной области эквивалентна поточечному умножению их частотных представлений. Эта теорема имеет большое практическое значение.

#### Теорема о свертке:

Если  $f(x, y)$  и  $h(x, y)$  являются двумя хорошими функциями от пространственных параметров  $x, y$ , то  $\mathbf{F}(f(x, y) \star h(x, y)) \equiv \mathbf{F}((f \star h)(x, y)) = \mathbf{F}(f(x, y))\mathbf{F}(h(x, y)) = \mathbf{F}(u, v)\mathbf{H}(u, v)$ , где  $\mathbf{F}$  есть оператор преобразования Фурье, а  $\star$  — оператор свертки.

Перед рассмотрением доказательства в одномерном случае приведем интерпретацию, широко используемую в области обработки сигналов: свертка может быть выполнена без применения маски  $h(x, y)$  ко всем точкам изображения  $f(x, y)$ .

Для часто используемых фильтров  $h$  преобразование  $H$ , скорее всего, будет известно в аналитической форме или будет храниться в памяти в виде массива.

Это позволяет устранить шаг (2) из алгоритма фильтрации. В учебниках по обработке сигналов обычно приводятся таблицы *par преобразования*  $\langle h, H \rangle$  с описанием в графической и в аналитической форме, так что пользователь может выбрать фильтр с подходящими свойствами. Теперь рассмотрим доказательство теоремы о свертке в одномерном случае. Аналогичные рассуждения могут быть проведены и в двумерном случае. В качестве вспомогательного шага рассмотрим, что происходит с преобразованием при смещении функции.

**Теорема запаздывания:**  $\mathbf{F}(f(x - x_0)) = e^{-j 2\pi u x_0} \mathbf{F}(f(x))$

По определению  $\mathbf{F}(f(x - x_0)) \equiv \int_{-\infty}^{+\infty} f(x - x_0) e^{-j 2\pi u x} dx$ .  
Выполним замену переменной  $x' = x - x_0$  и получим

$$\begin{aligned} \mathbf{F}(f(x - x_0)) &= \int_{-\infty}^{+\infty} f(x') e^{-j 2\pi u (x' + x_0)} dx' \\ &= \int_{-\infty}^{+\infty} e^{-j 2\pi u x_0} f(x') e^{-j 2\pi u x'} dx' \\ &= e^{-j 2\pi u x_0} \mathbf{F}(f(x)), \end{aligned} \quad (5.36)$$

так как первый множитель является константой по отношению к переменной интегрирования  $x'$ . Отметим, что

$$|e^{-j 2\pi u x_0}|^2 = \cos^2(2\pi u x_0) + \sin^2(2\pi u x_0) = 1, \quad (5.37)$$

так что смещение функции не меняет энергии  $f(x)$  и  $\mathbf{F}(u)$ .

Полученный результат можно использовать для доказательства теоремы о свертке.

$$\mathbf{F}((f \star h)(x)) \equiv \int_{x=-\infty}^{x=+\infty} \left( \int_{t=-\infty}^{t=+\infty} f(t) h(x - t) dt \right) e^{-j 2\pi u x} dx. \quad (5.38)$$

### Фильтрация изображения $f(x, y)$ с маской $h(x, y)$

- (1) Преобразование Фурье изображения  $f(x, y)$  для получения его частотного представления  $F(u, v)$ .
- (2) Преобразование Фурье маски  $h(x, y)$  для получения ее частотного представления  $H(u, v)$ .
- (3) Поточечное умножение  $F(u, v)$  и  $H(u, v)$  для получения  $F'(u, v)$ .
- (4) Применение обратного преобразования Фурье к  $F'(u, v)$  для получения отфильтрованного изображения  $f'(x, y)$ .

**Алгоритм 5.3.** Фильтрация изображения  $f(x, y)$  с маской  $h(x, y)$  с использованием преобразования Фурье.

Предположение о том, что функции  $f$  и  $h$  являются хорошими, позволяет изменить порядок интегрирования.

$$\mathbf{F}((f \star h)(x)) = \int_{t=-\infty}^{t=+\infty} f(t) \left( \int_{x=-\infty}^{x=+\infty} h(x-t) e^{-j 2\pi u x} dx \right) dt \quad (5.39)$$

С использованием теоремы запаздывания,

$$\int_{x=-\infty}^{x=+\infty} h(x-t) e^{-j 2\pi u x} dx = e^{-j 2\pi u t} \mathbf{H}(u), \quad (5.40)$$

где  $\mathbf{H}(u)$  является преобразованием Фурье функции  $h(x)$ . Теперь имеем

$$\begin{aligned} \mathbf{F}((f \star h)(x)) &= \int_{t=-\infty}^{t=+\infty} f(t) (e^{-j 2\pi u t} \mathbf{H}(u)) dt \\ &= \mathbf{H}(u) \int_{t=-\infty}^{t=+\infty} f(t) e^{-j 2\pi u t} dt \\ &= \mathbf{H}(u) \mathbf{F}(u) = \mathbf{F}(u) \mathbf{H}(u) \end{aligned} \quad (5.41)$$

и теорема о свертке в одномерном случае доказана.

### Упражнение 5.39

Ознакомьтесь с доказательством теоремы запаздывания и теоремы о свертке и распространите их на двумерный случай.

## 5.12 ИТОГИ И ОБСУЖДЕНИЕ

В этой довольно большой главе были рассмотрены несколько методов и большое количество примеров: еще раз отметим основные моменты. Представленные методы позволяют улучшить качество изображений как для восприятия человеком, так и для автоматической обработки. Были описаны несколько методов изменения тонового распределения значений интенсивности изображения для улучшения вида объектов сцены; было показано, что при этом некоторые области изображения часто удается улучшить за счет ухудшения других. Представленные методы были рассчитаны только на полутоновые изображения; однако большинство из них могут быть расширены для обработки цветных изображений (возможно, вы уже знаете об этом из опыта работы с различными программными пакетами обработки изображений). Выделение краев было рассмотрено в качестве метода обработки изображений для восприятия человеком. Также надеемся, что описанные методы позволили расширить набор художественных приемов работы с изображениями.

Наиболее важная идея данной главы состоит в том, что для обработки целого изображения можно использовать маску или ядро, задающую локальную структуру. Свертка и кросс-корреляция являются двумя очень мощными и

взаимосвязанными операциями, которые вычисляют результат в пикселе  $I[x, y]$  путем сложения поточечных произведений значений интенсивности входного полутонового изображения и соответствующих значений маски. Эти линейные операции широко распространены как в теории, так и на практике. В данном контексте было показано, что отклик на конкретную маску в конкретной точке изображения (*корреляция*) является мерой сходства структуры маски со структурой окрестности изображения. Это дает практический способ проектирования масок, или фильтров, для большого количества задач, таких, как сглаживание, выделение краев, выделение углов, или даже выделение текстуры.

По теме выделения краев имеется обширная литература. В данной главе были представлены несколько различных способов. Очевидно, что выходные данные специфического детектора краев могут оказаться очень полезными в конкретных прикладных задачах машинного зрения. Однако так и не была достигнута мечта многих исследователей — одно универсальное решение задачи низкоуровневого зрения о представлении существенных границ объекта с помощью некоторого описания, основанного на выделенных краях. Возможно, подобные ожидания нереалистичны. Например, если есть изображение автомобиля, как низкоуровневая система узнает, есть ли действительно на нем автомобиль; является ли он объектом интереса; движется он или стоит; интересуемся ли мы поиском царапин на его поверхности или точностью закрытия дверей, а не просто распознаванием автомобиля?

Многие изображения, полученные в наших примерах в результате выделения краев, кажутся многообещающими для различных приложений. Действительно, они таковыми и являются — многие методы последующих глав будут основываться на представлениях, построенных с использованием краев. Однако читатель не должен быть излишне оптимистичен, так как человек воспринимает изображения этой главы с использованием высокоразвитой зрительной системы и знаний об окружающем мире. Высокоуровневые методы компьютерной обработки должны быть устойчивыми к потере данных, так как на изображениях с выделенными краями на границах появляются промежутки, шумы и многоуровневые структурные элементы. Все это усложняет разработку алгоритмов для выделения информации из изображений с использованием краевых изображений в качестве входных данных.

## 5.13 ЛИТЕРАТУРА

В то время, когда выполнялись первые работы по обработке изображений и компьютерному зрению, существовало мало известных сейчас специализированных журналов. Поэтому исследовательские статьи публиковались во многих различных журналах. Ларри Робертс (Larry Roberts) опубликовал свою известную диссертационную работу в области распознавания трехмерных объектов с плоскими гранями в 1965 г. Часть этой работы была посвящена

выделению краев с использованием оператора, который сейчас носит его имя. Вероятно, Робертс не ожидал, что вскоре появится большое количество работ по выделению краев. Другие ранние работы опубликовали Превитт [Prewitt, 1970] и Кирш [Kirsch, 1971]. Маски Кирша теперь обычно рассматриваются как детекторы углов, а не краев. Недавняя работа Шина [Shin и др., 1998] подтвердила популярность детектора краев Кэнни [Canny, 1986]. В ней показано, что детектор Кэнни является одним из наилучших как по производительности, так и по эффективности, по крайней мере, в задачах восстановления структуры по данным о движении. В статье [Huertas, Medioni, 1986] представлено глубокое практическое рассмотрение ЛОГ-фильтров и показано, как их использовать для определения местоположения края с субпиксельной точностью. Операция выделения краев легко может быть расширена на трехмерные объемные изображения, что было сделано в работе [Zucker, Hummel, 1981].

В книге [Kreider и др., 1966] содержится базовый материал по линейной алгебре и ее применению к анализу функций, например, таких как наши функции интенсивности. Наиболее важными в этой книге являются гл. 1, 2 и 7. В гл. 9 и 10 приведены основные сведения по аппроксимации одномерных сигналов рядами Фурье и содержится полезный материал по поводу применения Фурье-анализа к двумерным функциям интенсивности. В учебниках по оптике [Hecht, Zajac, 1974] можно узнать о применении преобразований Фурье для интерпретации изображений как суперпозиции (сложения) колебаний. В книге [Cormen и др., 1990] по алгоритмам представлены два алгоритма для выбора *i*-го наименьшего из *n* чисел: временная сложность одного из них равна  $O(n)$ , что помещает медианную фильтрацию в один теоретический класс сложности вместе с усредняющей фильтрацией.

1. Canny, J. 1986. A computational approach to edge detection. *IEEE Trans. Pattern Anal. and Machine Intelligence*, 8(6):679–698.
2. Cormen, T., C. Leiserson, and R. Rivest. 1990. *Introduction to Algorithms*. MIT Press, Cambridge, MA.
3. Duda, R., P. Hart, and D. Stark. 2000. *Pattern Classification*, 2nd ed. John Wiley & Sons, New York.
4. Frei, W., and C-C. Chen. 1977. Fast boundary detection: a generalization and new algorithm. *IEEE Trans. Comput.*, C-26(10):988–998.
5. Hecht, E., and A. Zajac. 1974. *Optics*. Addison-Wesley, New York.
6. Huertas, A., and G. Medioni. 1986. Detection of intensity changes with subpixel accuracy using Laplacian-Gaussian masks. *IEEE-T-PAMI*, v. 8(5):651–664.
7. Kirsch, R. 1971. Computer determination of the constituent structure of biological images. *Comput. Biomed. Res.*, v. 4(3):315–328.
8. Kreider, D., R. Kuller, D. Ostberg, and F. Perkins. 1966. *An Introduction to Linear Analysis*. Addison-Wesley, New York.

9. Prewitt, J. 1970. Object enhancement and extraction. In *Picture Processing and Psychopictorics*, B. Lipkin and A. Rosenfeld, eds. Academic Press, New York, 75–149.
10. Roberts, L. 1965. Machine perception of three-dimensional solids. In *Optical and Electro-Optical Information Processing*, J. Tippett and others, eds. MIT Press, Cambridge, MA, 159–197.
11. Shin, M., D. Goldgof, and K. Bowyer. 1998. An objective comparison methodology of edge detection algorithms using a structure from motion task. In *Empirical Evaluation Techniques in Computer Vision*, K. Bowyer and P. Philips, eds. IEEE Computer Society Press, Los Alamitos, CA.
12. Zucker, S., and R. Hummel. 1981. A three-dimensional edge operator. *IEEE-T-PAMI*, v. 3:324–331.



## Цвет и освещенность

Для людей зрительное восприятие цвета имеет очень большое значение. Цветовые ощущения зависят и от физических свойств света, и от психофизиологических особенностей сложной системы глаз-мозг, в которой при обработке зрительной информации совместно учитываются и сигналы возбуждения рецепторов сетчатки, и накопленный человеком опыт. На основе цветовой информации люди способны различать объекты, материалы и даже время дня. На рис. 6.1 одна и та же сцена представлена в различных цветах. Хотя форма образов не меняется, но эти изображения сильно отличаются, и наблюдатель может интерпретировать правое изображение как сцену с домашней кошкой в комнате, а не как фотографию тигра в траве.

Благодаря технологическому прогрессу относительно недавно стали общедоступны устройства для работы с цветными изображениями. Разработаны недорогие цветные камеры, цветные дисплеи и программное обеспечение



**Рис. 6.1.** (Слева) изображение тигра в траве в естественных цветах. (Справа) изображение с измененными цветами. На этом изображении узнать тигра сложнее — может быть, это домашняя кошка на ковре? (С любезного разрешения Corel Stock Photos.)

для обработки цветных изображений. Все эти факторы обусловили широкое распространение приложений машинной обработки цветных изображений. В технических системах цветовая информация может использоваться в тех же целях, что и при восприятии человеком. Цветовые данные для каждого пиксела изображения содержат несколько значений измерений. В ряде задач на основе этих данных удается классифицировать пиксела без сложной пространственной обработки изображения.

Подробное описание физических свойств и особенностей цветового восприятия заняло бы много страниц. В этой главе приведены только основные сведения, которые помогут читателю начать разработку программ для обработки цветных изображений. Представленный материал также можно рассматривать в качестве руководства по выбору дополнительной литературы. В данной главе сначала описывается ряд основных физических понятий, связанных с цветом, а также практические методы представления цветовой информации на изображениях. Затем обсуждаются несколько примеров и методы использования цветowych данных в задачах распознавания объектов и сегментации изображений.

В главе рассмотрены вопросы, связанные с освещенностью объектов. Освещенность зависит не только от цвета объектов и свойств источников света, но и от многих других факторов. Например, от шероховатости поверхностей, от ориентации поверхности относительно источников света и наблюдателя, от расстояния между поверхностью и источниками света и наблюдателем. Цвет и освещенность — важнейшие компоненты произведений искусства на протяжении столетий — также очень важны для интерпретации сцен с помощью алгоритмов компьютерного зрения.

## 6.1 ФИЗИЧЕСКИЕ СВОЙСТВА ЦВЕТА

Электромагнитное излучение с длиной волны  $\lambda$  в диапазоне примерно от 400 до 700 нанометров возбуждает рецепторы зрительной системы человека и порождает цветовые ощущения (рис. 6.2). Нанометр равен  $10^{-9}$  метра и иногда называется миллимикрон. Длина волны синего света равна  $400 \text{ нм}$  ( $400 \times 10^{-9} \text{ м}$ ). Это значит, что на метре укладывается  $2.5 \times 10^6$  периодов волны, а на сантиметре — 25000 периодов. Свет в вакууме распространяется



Рис. 6.2. Видимый диапазон электромагнитного спектра.

со скоростью  $3 \times 10^8$  м/с. С учетом этого можно вычислить частоту синего света: она равна  $0.75 \times 10^{15}$  Гц. Частота рентгеновского излучения в тысячу раз больше, а частота радиоволн — в миллиард раз меньше.

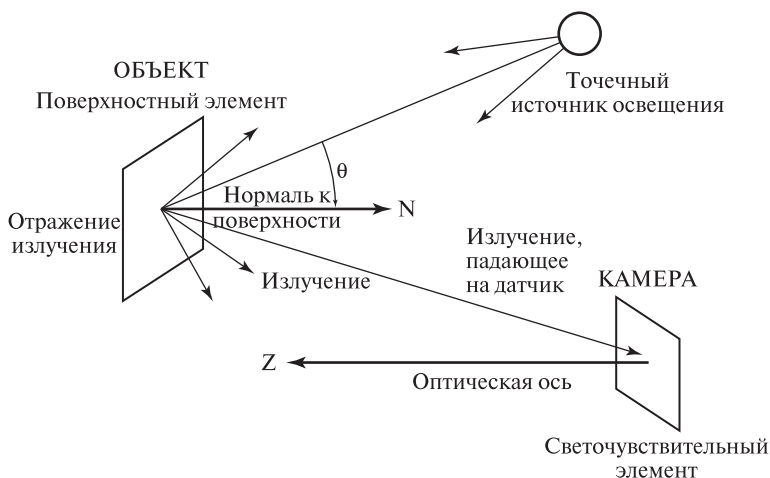
В оставшейся части данной главы длина или частота световых волн будет рассматриваться только в контексте качественного цветового восприятия. Существуют датчики для обнаружения излучения за пределами видимого диапазона, например, разработаны устройства для детектирования коротковолнового ультрафиолетового и даже рентгеновского излучения. Длинноволновое инфракрасное излучение можно обнаружить с помощью многих твердотельных камер, а еще более длинные радиоволны — с помощью радиоприемника. В научных и промышленных целях было сконструировано много различных устройств для восприятия и преобразования в видимый диапазон результатов измерений, выполненных за пределами видимого диапазона. Примерами таких устройств являются рентгеновский аппарат и спутниковый метеорологический инфракрасный сканер.

## Упражнение 6.1

Толщина листа бумаги равна 0.1 мм. Сколько длин волн синего света соответствует такому расстоянию?

### 6.1.1 Восприятие освещенных объектов

На рис. 6.3 иллюстрируется освещение поверхности объекта точечным источником света. В результате взаимодействия падающего света с атомами



**Рис. 6.3.** Свет от источника отражается от поверхности объекта и попадает на элемент датчика.

объекта часть света отражается от поверхности и попадает на элемент датчика камеры или на рецептор человеческого глаза. Возникающее цветовое ощущение зависит от трех основных факторов:

- Энергетический спектр. Описывает распределение падающей на поверхность объекта световой энергии по длинам волн.
- Спектральная отражательная способность поверхности объекта. Определяет преобразование спектра падающего света в спектр отраженного.
- Спектральная чувствительность датчика, на который падает свет от поверхности объекта.

**64 Определение. Белый свет** представляет собой смесь световых компонент всех длин волн видимого спектра, имеющих примерно одинаковую энергию.

Поверхность объекта, который выглядит *синим*, сделана из материала, который кажется синим при освещении *белым светом*. При освещении красным светом этот объект будет выглядеть фиолетовым. Синий автомобиль при интенсивном (белом) солнечном освещении нагревается и при прикосновении кажется горячим. Автомобиль излучает в инфракрасном диапазоне. Это излучение недоступно зрению человека, но может быть обнаружено с помощью инфракрасной камеры.

### 6.1.2 Дополнительные факторы

Кроме перечисленных выше трех основных факторов, от которых зависит цветовое ощущение, существует ряд дополнительных усложняющих модель факторов. Они связаны как с физическими свойствами, так и с особенностями человеческого восприятия. Отражающие свойства поверхностей различны. Например, поверхности различаются зеркальными свойствами (поверхность больше или меньше напоминает зеркало). Матовые поверхности отражают свет равномерно по всем направлениям. Энергия (интенсивность) падающего излучения зависит от расстояния — чем дальше поверхностные элементы от источника света, тем меньше они получают энергии. В результате интенсивности пикселей изображения, соответствующие поверхностям из одного и того же материала, могут иметь неодинаковые значения из-за различий в расстоянии от пикселей датчика до поверхностей объектов вдоль проектирующих лучей. Для определения энергии, отраженной по направлению к датчику, еще большее значение, чем расстояние, имеет ориентация  $\theta$  поверхностного элемента относительно источника света. Перечисленные вопросы будут более подробно рассматриваться в конце данной главы.

---

**Упражнение 6.2.** Изменение интенсивности в зависимости от расстояния до источника света

Поместите лист белой бумаги перед камерой, подключенной к компьютеру. Лист должен быть параллелен плоскости изображения. Осветите один из углов листа бумаги лампой накаливания. Получите изображение и ознакомьтесь с распределением значений интенсивности. Насколько они различны? Заметно ли систематическое уменьшение значений интенсивности с увеличением расстояния до поверхности объекта?

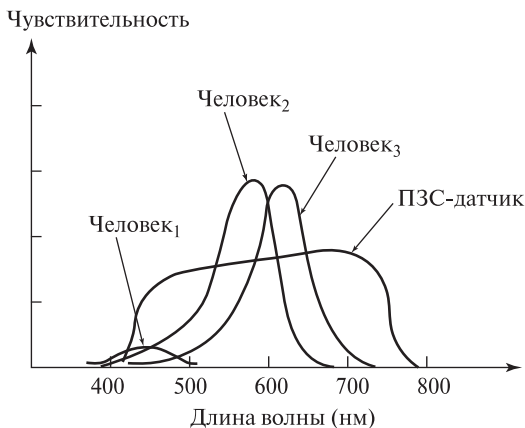
---

**Упражнение 6.3.** Изменение интенсивности в зависимости от направления нормали к поверхности

Повторите эксперимент из предыдущего упражнения, заменив плоский лист бумаги сферическим волейбольным мячом. Получите изображение и ознакомьтесь с распределением значений интенсивности. Опишите замеченные изменения и закономерности.

### 6.1.3 Чувствительность рецепторов

Все известные рецепторы чувствительны к излучению только в некотором диапазоне длин волн. Обычно чувствительность рецепторов в этом диапазоне неравномерна. Примеры кривых чувствительности приведены на рис. 6.4. Три из них соответствуют разнотипных колбочкам сетчатки человеческого глаза. В этих колбочках содержатся различные химические пигменты, чувствительные к различным длинам волн. Кривая Человек<sub>1</sub> соответствует колбочкам, которые чувствительны к синему свету с длинами волн примерно от 400 до 500 нм. Кривая Человек<sub>2</sub> построена для колбочек, наиболее восприимчивых к зеленому свету, но также обладающих меньшей чувствительностью к синему и красному свету. При формировании ощущения некоторого видимого цвета мозг учитывает отклики от окрестности из нескольких колбочек. Довольно интересно, что для этого достаточно рецепторов всего трех типов, хотя длины волн света могут иметь любые значения из видимого диапазона. Глаза многих животных содержат рецепторы только одного или двух типов. Вероятно, цветовое восприятие этих животных беднее по сравнению с человеком. Твердотельные датчики обычно характеризуются хорошей чувствительностью в диапазоне, превышающем видимый диапазон. При нагреве объектов сцены система машинного зрения иногда может формировать изображения, которые будут отличаться от результатов наблюдения оператором. Главным образом это объясняется повышенной чувствительностью ПЗС-датчиков к инфракрасному излучению.



**Рис. 6.4.** Сравнение относительной чувствительности человеческих колбочек трех типов и твердотельного датчика.

#### Упражнение 6.4. Любимый цвет

Есть ли у вас любимый цвет? Если есть, то какой именно и почему? Задайте этот вопрос трем другим людям. Предположим, что их ответы различны. Можете ли вы объяснить это различие на основе физических свойств света?

## 6.2 ЦВЕТОВАЯ СИСТЕМА RGB

С помощью рецепторов всего трех типов люди способны различать тысячи цветов. В графических системах для представления цветов широко применяется *трехцветная* цветовая система RGB (red-green-blue, красный-зеленый-синий). Для описания цвета пиксела обычно используются три байта, что дает  $(2^8)^3$  (примерно 16 миллионов) различных цветовых кодов. 16 миллионов кодов не означает «16 миллионов цветов», так как люди не способны различать такое количество цветов. Компьютеры могут отличить любые два кода, но эти коды не обязательно будут представлять реально различимые человеком цвета. Значение каждого пиксела в трехбайтной кодировке в системе RGB занимает три байта (24 бита). Для красной, зеленой и синей составляющей цвета отводится по одному байту. Порядок хранения этих компонент в памяти может различаться. Порядок хранения не имеет значения для теоретического рассмотрения, но важен для программирования. Графические дисплеи, цветовое разрешение которых соответствует человеческой зрительной системе, называются полноцветными (true color), или дисплеями с реалистичным цветовоспроизведением. При этом для представления цвета требуется не менее 16 бит. При 15-битном RGB-кодировании для составляющих R, B и G используется по 5 бит. При 16-битном кодировании учитывается повышенная чувствительность человека к зеленому цвету, поэтому для зеленой составляющей резервируется 6 бит.

	RGB	CMY	HSI
КРАСНЫЙ	(255, 0, 0)	(0, 255, 255)	(0.0, 1.0, 255)
ЖЕЛТЫЙ	(255, 255, 0)	(0, 0, 255)	(1.05, 1.0, 255)
	(100, 100, 50)	(155, 155, 205)	(1.05, 0.5, 100)
ЗЕЛЕНый	(0, 255, 0)	(255, 0, 255)	(2.09, 1.0, 255)
СИНИЙ	(0, 0, 255)	(255, 255, 0)	(4.19, 1.0, 255)
БЕЛый	(255, 255, 255)	(0, 0, 0)	(-1.0, 0.0, 255)
СЕРый	(192, 192, 192)	(63, 63, 63)	(-1.0, 0.0, 192)
	(127, 127, 127)	(128, 128, 128)	(-1.0, 0.0, 127)
	(63, 63, 63)	(192, 192, 192)	(-1.0, 0.0, 63)
	...		
ЧЕРный	(0, 0, 0)	(255, 255, 255)	(-1.0, 0.0, 0)

**Рис. 6.5.** Представление цветов в различных трехцветных системах. Часто оказывается удобно масштабировать значения интенсивности для попадания в диапазон  $[0, 1]$ . Компоненты цвета в системе HSI можно вычислить по значениям RGB с помощью алгоритма 6.1:  $H \in [0.0, 2\pi]$ ,  $S \in [0.0, 1.0]$  и  $I \in [0, 255]$ .

Комбинация трех основных цветов (RGB) позволяет представить произвольный цвет видимого спектра (рис. 6.5). При смешении равных частей красного (255, 0, 0) и зеленого (0, 255, 0) цвета получается желтый цвет (255, 255, 0). Количество каждого основного цвета соответствует интенсивности соответствующей световой волны. Три компоненты максимальной интенсивности образуют белый цвет. Смесь цветовых компонент равной интенсивности, меньшей максимальной, соответствует оттенкам серого цвета. Цвет  $(c, c, c)$  при  $0 < c < 255$  с уменьшением  $c$  стремится к черному цвету (0, 0, 0). Часто бывает удобно масштабировать значения интенсивности так, чтобы они попадали в диапазон  $[0, 1]$ , а не  $[0, 255]$ . Это упрощает разработку аппаратно-независимых алгоритмов обработки цветных изображений.

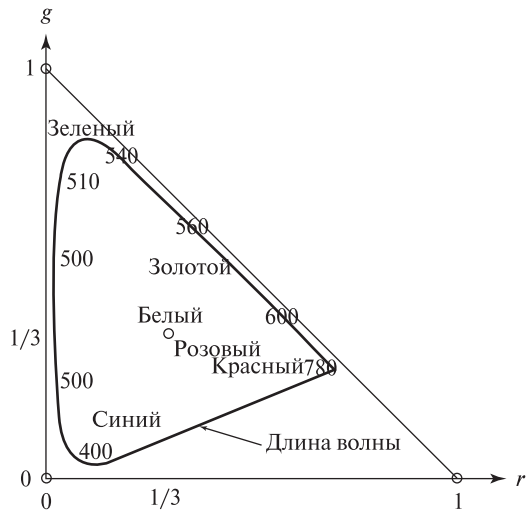
Система RGB называется *аддитивной цветовой системой*, так как цвета представляются посредством сложения основных цветов с черным цветом (0, 0, 0). Эта система хорошо подходит для RGB-дисплеев, в которых свет излучается участками люминофора трех типов. Три соседних элемента люминофора образуют один пиксел. На эти элементы падают электронные пучки с интенсивностями  $c_1$ ,  $c_2$  и  $c_3$ . Человеческий глаз воспринимает излучение от близко расположенных элементов совместно как цвет  $(c_1, c_2, c_3)$ . Следовательно, свет трех длин волн от небольшого участка экрана электронно-лучевой трубки, смешивается и воспринимается совместно.

Предположим, что цветной датчик кодирует пиксел цифрового изображения тройкой чисел  $(R, G, B)$ , где каждая координата принадлежит диапазону  $[0, 255]$ . Уравнения 6.1 описывают один из способов нормировки цветовых

данных изображения для последующей интерпретации компьютерными программами или человеком, а также для преобразования в другие цветовые системы (см. далее). Допустим, что цветная камера формирует изображение сцены с неоднородным освещением. Например, точки поверхности объекта расположены на различных расстояниях от источников света или находятся в тени некоторых источников. В таких условиях алгоритм поиска зеленых пикселей, соответствующих образу автомобиля, мог бы дать плохие результаты, если только сначала не была выполнена нормировка интенсивности пикселей.

$$\begin{aligned}
 \text{интенсивность } I &= (R + G + B)/3 \\
 \text{нормированный красный } r &= R/(R + G + B) \\
 \text{нормированный зеленый } g &= G/(R + G + B) \\
 \text{нормированный синий } b &= B/(R + G + B)
 \end{aligned}
 \tag{6.1}$$

После применения уравнений (6.1) сумма нормированных цветовых составляющих пиксела всегда будет равна 1. Известны и другие способы нормировки. Например, в знаменателе вместо суммы цветовых компонент можно было бы использовать  $\max(R, G, B)$ . При выполнении условия  $r + g + b = 1$  между цветовыми компонентами и двумерными координатами существует соответствие, позволяющее представить цвета в виде точек на двумерном графике (рис. 6.6). Чистые цвета изображаются точками вблизи углов треугольника. Например, ярко-красные цвета располагаются вблизи нижнего правого угла с координатами  $(1, 0)$ , а ярко-зеленые — в окрестности верхнего угла с координатами  $(0, 1)$ . Белый цвет находится в центре тяжести  $(1/3, 1/3)$ . На рис. 6.6 ось синего цвета  $(b)$  направлена перпендикулярно плоскости графика, т. е. осям  $r$  и  $g$ . Фактически этот треугольник представляет собой срез трехмерного цветового



**Рис. 6.6.** Цветовой треугольник в нормированных координатах RGB. Ось синего цвета  $(b)$  направлена перпендикулярно плоскости графика.



пространства в плоскости, проходящей через точки  $[1, 0, 0]$ ,  $[0, 1, 0]$  и  $[0, 0, 1]$ . Для любой пары значений  $r-g$ , показанной на графике внутри треугольника, можно вычислить значение синей составляющей  $b = 1 - r - g$ .

### Упражнение 6.5. Эксперименты с цветовыми кодами

Получите цветное изображение в системе RGB и просмотрите его с помощью какой-нибудь графической утилиты. Поменяйте местами зеленые и синие цветовые компоненты и посмотрите на получившееся изображение. Удвойте значения всех цветовых компонент, а затем только синие компоненты с небольшими значениями и опишите полученные результаты.

## 6.3 ДРУГИЕ ЦВЕТОВЫЕ СИСТЕМЫ

Были разработаны несколько цветовых систем, обладающих преимуществами применительно к различным цветным устройствам или человеческому восприятию. Некоторые, но не все, цветовые системы связаны линейными преобразованиями.

### 6.3.1 Субтрактивная цветовая система CMY

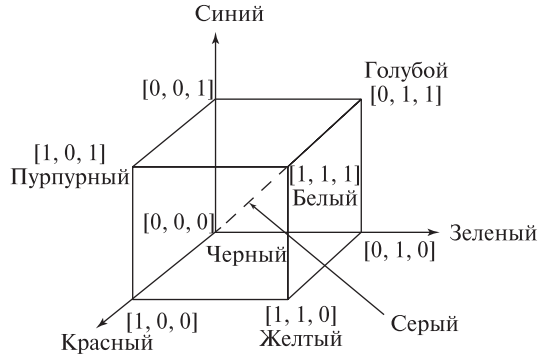
Цветовую систему CMY можно рассматривать как модель печати на белой бумаге, при которой цвета формируются путем вычитания основных цветов из белого цвета. Этим система CMY отличается от системы RGB, в которой выполняется сложение основных цветов с черным цветом. Примеры цветовых кодов CMY приведены на рис. 6.5 после кодов RGB.

Сокращение CMY образовано первыми буквами названий трех основных цветов данной системы *Cyan-Magenta-Yellow* (голубой-пурпурный-желтый). Эти цвета соответствуют разноцветным печатным краскам, поглощающим свет одного из основных цветов системы RGB. Голубая краска поглощает красный свет, пурпурная поглощает зеленый, а желтая — синий. Таким образом при печати формируются необходимые отражательные свойства, и отпечатанное изображение корректно воспринимается при освещении белым светом. Цветовая система CMY называется субтрактивной, так как значения кодов характеризуют интенсивность поглощенного излучения. Приведем несколько примеров трехцветных кодов в субтрактивной системе: белый —  $(0, 0, 0)$ , так как белый свет не должен поглощаться, черный —  $(255, 255, 255)$ , так как все компоненты белого света должны быть поглощены, желтый —  $(0, 0, 255)$ , так как синяя компонента падающего белого света должна быть поглощена, а красная и зеленая должны отразиться и сформировать ощущение желтого цвета.

### 6.3.2 Цветовая система HSI: Тон-Насыщенность-Интенсивность (Hue-Saturation-Intensity)

В системе HSI цветной пиксел кодируется одним значением общей интенсивности света  $I$  и двумя значениями *цветности* — тоном  $H$  и насыщенностью  $S$ .

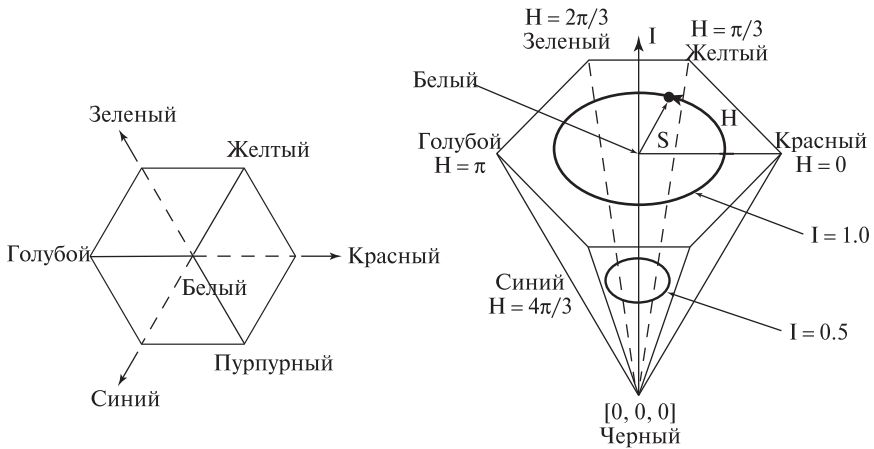
**Рис. 6.7.** Цветовой куб в нормированных координатах RGB. Треугольник на рис. 6.6 является проекцией куба на плоскость, проходящую через точки  $[1, 0, 0]$ ,  $[0, 1, 0]$ , и  $[0, 0, 1]$ .



На рис. 6.7 показан цветовой куб, связанный с треугольником системы RGB на рис. 6.6. При представлении цветового пространства RGB в виде куба каждое значение  $r$ ,  $g$  и  $b$  может меняться независимо в диапазоне  $[0.0, 1.0]$ . Плоская проекция цветового куба в направлении главной диагонали имеет вид шестиугольника (рис. 6.8, слева). В этом представлении все оттенки серого, ранее располагавшиеся вдоль диагонали цветового куба, проектируются в центральную белую точку шестиугольника, красная точка  $[1, 0, 0]$  проектируется в правый угол, а зеленая точка  $[0, 1, 0]$  — в верхний левый угол шестиугольника. С таким способом представления цветового пространства связана еще одна трехмерная модель в виде шестигранного конуса (см. рис. 6.8, справа). В этой модели диагональ куба изображается в виде вертикальной оси интенсивности  $I$ . Тон  $H$  определяется как угол от  $0$  до  $2\pi$  относительно оси красного цвета. Чистому красному цвету соответствует угол  $0$ , чистому зеленому —  $2\pi/3$ , а чистому синему —  $4\pi/3$ . Для задания точки в цветовом пространстве необходима третья координата — насыщенность  $S$ . Насыщенность характеризует чистоту цвета (цветового тона). Значение  $1$  соответствует идеально чистому (насыщенному) цвету, а  $0$  — совершенно ненасыщенному цвету, т. е. некоторому оттенку серого цвета.

Система HSI иногда называется системой HSV, когда вместо *интенсивности* используется термин *светлота* (color value). Система HSI удобна многим графическим дизайнерам, так как в ней обеспечивается непосредственный контроль яркости и тона. Пастельные цвета располагаются в центре цветового пространства вблизи оси  $I$ , а глубокие цвета — в периферийных областях шестигранного конуса. HSI лучше остальных цветовых систем подходит для некоторых алгоритмов компьютерного зрения, так как допускает простую нормировку на интенсивность света и позволяет сосредоточиться на двух параметрах цветности, которые больше связаны со свойствами поверхности, а не источника света.

Алгоритм 6.1 предназначен для вычисления цветовых координат HSI по координатам RGB. Входными данными являются координаты  $(r, g, b)$ , определяющие точку внутри цветового куба. Эти значения могут быть нормированы по уравнениям (6.1) или принадлежать диапазону  $[0, 255]$  (см. левый столбец



**Рис. 6.8.** Представление цветового пространства системы HSI в виде шестигранного конуса. Слева приведена проекция куба RGB на плоскость, перпендикулярную диагонали между вершинами  $[0, 0, 0]$  и  $[1, 1, 1]$ . Названия цветов указаны рядом с вершинами шестиугольника. Шестигранный конус является моделью цветового пространства HSI. Вертикальная ось соответствует интенсивности ( $I$ ); тон ( $H$ ) определяется как угол от  $0$  до  $2\pi$  с осью красного цвета; насыщенность ( $S$ ) лежит в диапазоне от  $0$  до  $1$  и указывает на чистоту цвета («непохожесть» на белый цвет). При  $S = 0.0$  цвет лежит на оси  $I$ .

на рис. 6.5). Вычисленное значение интенсивности  $I$  принадлежит тому же диапазону, что и входные значения. При  $I = 0$  значение насыщенности  $S$  не определено. Значение  $H$  принадлежит диапазону  $[0, 2\pi]$  и не определено при  $S = 0$ . Вместо вычисления квадратного корня и арккосинуса, входящих в математические выражения формул преобразования, в Алгоритме 6.1 применяются очень простые операции, позволяющие эффективно преобразовать весь массив пикселей изображения. Примеры выходных данных алгоритма 6.1 приведены в правом столбце на рис. 6.5.

### Упражнение 6.6

С помощью алгоритма 6.1 преобразуйте цветовые RGB-коды  $(100, 150, 200)$  и  $(0.0, 1.0, 0.0)$  в коды системы HSI.

Обратите внимание на связь координат HSI с цветовым треугольником (рис. 6.6). Тон связан с преобладающей длиной световой волны. Цвета с большими значениями тона лежат вблизи сторон треугольника. Наименьшие длины волн видимого диапазона  $\lambda$  равны примерно  $400$  нм. Соответствующие точки лежат вблизи начала координат. С увеличением длины волн точки располагаются выше по оси зеленого цвета ( $g$ ) примерно до  $520$  нм. С дальнейшим увеличением  $\lambda$  до  $800$  нм точки будут попадать все ниже на

### Преобразование цветовых координат из системы RGB в систему HSI

$R, G, B$  : все входные значения RGB принадлежат диапазону  $[0, 1]$  или  $[0, 255]$ ;

$I$  : выходное значение интенсивности (в том же диапазоне, что и входные данные);

$S$  : выходное значение насыщенности в диапазоне  $[0, 1]$ ;

$H$  : выходное значение цветового тона в диапазоне  $[0, 2\pi)$ , или  $-1$  при  $S = 0$ ;

Все переменные  $R, G, B, H, S, I$  являются числами с плавающей точкой;

```
procedure RGB_to_HSI(in R,G,B; out H,S,I)
```

```
{
```

```
  I := max (R, G, B);
```

```
  min := min (R, G, B);
```

```
  if (I ≥ 0.0) then S := (I - min)/I else S := 0.0;
```

```
  if (S ≤ 0.0) then {H := -1.0; return;}
```

```
    “Вычисление цветового тона на основе отношения значений RGB”
```

```
  diff := I - min;
```

```
  “Лежит ли точка в пределах  $\pm 60$  градусов от оси красного цвета?”
```

```
  if (R = I) then H :=  $(\pi/3) * (G - B) / \text{diff}$ ;
```

```
  “Лежит ли точка в пределах  $\pm 60$  градусов от оси зеленого цвета?”
```

```
  else if (G = I) then H :=  $(2 * \pi/3) + \pi/3 * (B - R) / \text{diff}$ ;
```

```
  “Лежит ли точка в пределах  $\pm 60$  градусов от оси синего цвета?”
```

```
  else if (B = I) then H :=  $(4 * \pi/3) + \pi/3 * (R - G) / \text{diff}$ ;
```

```
  if (H ≤ 0.0) H := H +  $2\pi$ ;
```

```
}
```

**Алгоритм 6.1.** Преобразование цветовых координат из системы RGB в HSI.

гипотенузу треугольника. Значение тона равно углу между горизонтальной осью, проходящей через центр тяжести треугольника (точку белого цвета), и направлением на данную точку  $(r, g)$  на стороне треугольника. Значения  $H$  и  $S$  для золотого цвета с насыщенностью 50% лежат на рис. 6.6 посередине между точками белого и золотого цвета. Треугольник на рис. 6.6 отдаленно напоминает палитру художника.

На рис. 6.9 демонстрируется преобразование изображения путем изменения насыщенности. Исходное входное изображение показано слева. Изображение в центре получено после увеличения насыщенности  $S$  всех пикселей на 40%, а изображение справа — после уменьшения  $S$  на 20%. С учетом нашего опыта, цвета на центральном изображении кажутся чрезмерно утрированными, а



**Рис. 6.9.** (Слева) входное RGB-изображение. (В центре) насыщенность  $S$  увеличена на 40%. (Справа) насыщенность  $S$  уменьшена на 20%. (С любезного разрешения Frank Biocca.) См. цветную вклейку.

изображение справа кажется размытым. Цветовой тон  $H$  на всех трех изображениях постояен. Поэтому цветовой тон в данном случае мог бы оказаться надежным признаком для цветовой сегментации. Цветовой тон не зависит от изменения интенсивности белого света, что обычно происходит в сценах, наблюдаемых системами машинного зрения.

### Упражнение 6.7

Разработайте программу для преобразования цветковых координат  $r, g, b$ , принадлежащих диапазону  $[0, 1]$ , в координаты  $H, S, I$ . Примените описанный далее прием на основе аналитической геометрии. Опустите перпендикуляр из точки  $[r, g, b]$  на диагональ цветового куба (проходящую через вершины  $[0, 0, 0]$  и  $[1, 1, 1]$ ) и вычислите соответствующие координаты  $H, S, I$ .

### 6.3.3 Телевизионные цветковые системы YIQ и YUV

В телевизионном стандарте NTSC цветковая информация представляется с помощью одного яркостного сигнала  $Y$  и двух цветоразностных сигналов  $I$  и  $Q$ . Черно-белые телевизоры используют только яркостный сигнал, а цветные — все три сигнала. Приближенное линейное преобразование для перехода из системы RGB в YIQ описывается уравнениями (6.2). На практике значения  $Y$  обычно кодируются большим количеством бит, чем значения  $I$  и  $Q$ . Это объясняется тем, что человеческое зрение более чувствительно к световой яркости (интенсивности), а не цвету.

$$\begin{aligned}
 \text{яркость } Y &= 0.30R + 0.59G + 0.11B \\
 \text{красный-голубой } I &= 0.60R - 0.28G - 0.32B \\
 \text{пурпурный-зеленый } Q &= 0.21R - 0.52G + 0.31B
 \end{aligned} \tag{6.2}$$

Система YUV применяется в некоторых цифровых видеосистемах и в алгоритмах сжатия (например, JPEG и MPEG). Преобразование из системы

RGB в YUV выполняется с помощью следующих выражений:

$$\begin{aligned} Y &= 0.30R + 0.59G + 0.11B \\ U &= 0.493 * (B - Y) \\ V &= 0.877 * (R - Y) \end{aligned} \tag{6.3}$$

Цветовые системы YIQ и YUV хорошо подходят для сжатия цифровых изображений и видеоданных. В этом отношении они превосходят многие другие форматы, так как яркость и цветность в них может быть представлена с использованием различного количества бит, чего невозможно достичь в системе RGB.

---

### Упражнение 6.8. Преобразование цветовых кодов

Предположим, что с помощью цветной камеры для заданного пиксела получен RGB-код (200, 50, 100). Максимальное значение интенсивности (соответствующее наибольшей воспринимаемой энергии падающего на датчик света) равно 255. (а) Какие значения имеет соответствующая тройка координат в системе HSI? (б) В системе YIQ?

---

### Упражнение 6.9

Обратимо ли преобразование из системы RGB в YIQ? Если обратимо, то поучите выражение обратного преобразования.

---

### Упражнение 6.10. Перекодировка изображений

Будем полагать, что доступны графический дисплей и программное обеспечение для просмотра RGB-изображений. Проведите следующий эксперимент. Во-первых, сформируйте HSI-изображение, на котором правая верхняя четверть заполнена насыщенным красным цветом, левая нижняя четверть — насыщенным желтым, левая верхняя четверть — синим цветом с насыщенностью 50%, а правая нижняя четверть — зеленым цветом с насыщенностью 50%. Конвертируйте изображение из системы HSI в RGB с помощью преобразования, обратного алгоритму 6.1. Выведите изображение на экран и выясните значения цветов в четырех частях изображения.

## 6.3.4 Классификация с использованием цветовой информации

Во многих приложениях при классификации пикселей можно учитывать их цвет. В разд. 6.5 описывается цветовая модель для представления цвета человеческой кожи. Эта модель была сформирована для решения задачи обнаружения человеческих лиц на цветных изображениях. Однако при цветовой

классификации возможны неточности. Например, пикселы образа коричневой картонной коробки могут успешно пройти проверку на цвет кожи. Для предотвращения ошибок далее может потребоваться проверка формы области, которая позволит отличить многогранную область коробки от овальной формы образа человеческого лица.

На рис. 6.10 показаны белые области изображения, которые были обнаружены после выборки пикселей, значения которых близки к эталонным белым пикселям. На этапе обучения эталонные значения были определены по пикселям в областях символов на изображении знака. На рис. 6.10 (справа) несколько лишних областей образовались из-за других белых объектов и зеркальных бликов. На полученном изображении алгоритмы распознавания символов смогли бы обнаружить большинство символов и отбросить большинство лишних компонент.

Результаты интерпретации цвета отдельных пикселей сильно подвержены ошибкам. Изображение слева на рис. 6.9 было получено путем фотографирования со вспышкой. Поэтому из-за зеркального отражения (обсуждается в разд. 6.6.3) некоторые грани ломтиков ананаса выглядят белыми. Классификатор, считающий такие пиксели желтыми, вероятно, будет также относить к желтым областям зеркальные блики на синей чашке и т. п.

Проблемы интерпретации также возникают в некоторых областях цветового пространства. При близких к 0 значениях насыщенности вычисление и интерпретация цветового тона выполняются ненадежно. При малых значениях интенсивности также ненадежной оказывается интерпретация насыщенности.



**Рис. 6.10.** Белые пикселы, выделенные в результате сегментации левого цветного изображения. Отдельные связные компоненты белых пикселей маркированы с помощью алгоритма маркировки из гл. 3. (С любезного разрешения David Moore.) См. цветную вклейку.

### Упражнение 6.11

Продемонстрируйте нестабильность преобразования из системы RGB в HSI при близких к 0 значениях насыщенности или интенсивности. Для этого выполните следующие эксперименты. Напишите программу с реализацией алгоритма 6.1.

(а) Преобразуйте RGB-коды  $(L + \Delta L_R, L + \Delta L_G, L + \Delta L_B)$  в систему HSI при больших значениях  $L$  и  $\Delta L_X \in \{-2, -1, 1, 2\}$ . Состоятельны ли значения  $H$ ?

(б) Повторите этот эксперимент при малом значении  $L$  (порядка 10) и тех же значениях  $\Delta$ . Состоятельны ли значения  $S$ ?

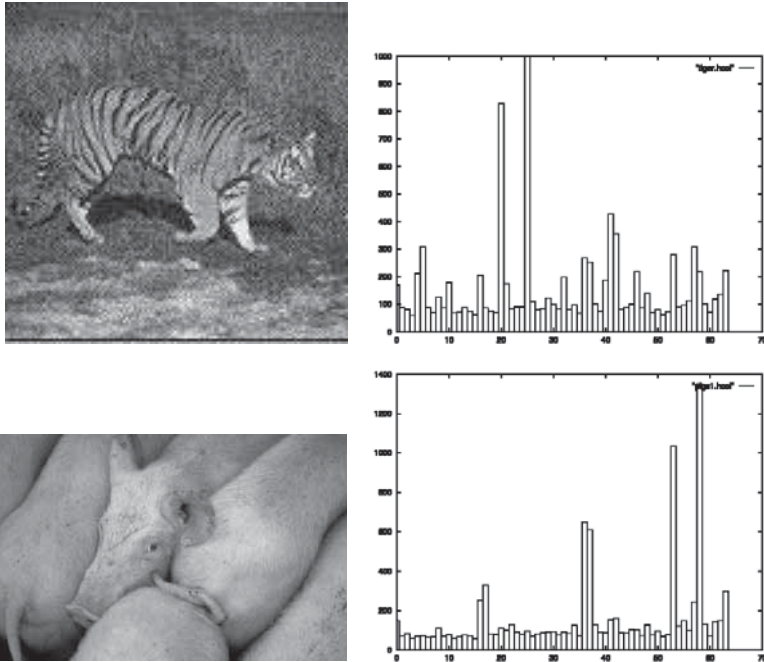
## 6.4 ЦВЕТОВЫЕ ГИСТОГРАММЫ

Гистограмма цветного изображения может применяться в качестве представления изображения в задачах поиска в базах данных или при распознавании объектов. Для построения гистограммы надо подсчитать на изображении количество пикселей определенного типа. Эта операция выполняется путем однократного чтения каждого пиксела изображения и увеличения значения в соответствующем кармане гистограммы. Применение цветовых гистограмм для поиска изображений в базах данных описывается в гл. 8. Цветовые гистограммы относительно мало зависят от преобразований переноса, поворота и малых наклонов относительно оптической оси, масштабирования и частичного перекрытия объектов. В данном разделе рассматривается метод сопоставления цветовых гистограмм, разработанный в 1991 г. Свейном и Баллардом (Swain, Ballard) для задач распознавания объектов.

Простой приближенный метод построения гистограммы цветного изображения предполагает объединение двух старших бит от каждой цветовой составляющей трехбайтного значения цвета RGB. По этим 6-битным кодам строится гистограмма, содержащая  $2^6 = 64$  разрядов (карманов). Также возможно независимое вычисление трех гистограмм, по одной для каждого цвета, и последующее их объединение в одну гистограмму. Например, отдельные 16-разрядные RGB-гистограммы можно было бы объединить в одну 48-разрядную гистограмму [Jain, Vailaya, 1996]. На рис. 6.11 приведены два примера цветных изображений и построенных для них гистограмм.

Пересечением (intersection) гистограммы изображения  $h(I)$  и модельной гистограммы  $h(M)$  будем называть сумму минимальных значений среди всех  $K$  пар соответствующих карманов этих гистограмм (см. выражение (6.4)). Значение пересечения гистограмм можно нормировать, разделив это значение на количество пикселей модельного изображения. Полученную величину будем называть сходством (match value). Эта характеристика служит мерой близости цветового содержания данного изображения и модельного (эталонного) изображения. Если фоновые пиксели данного изображения отсутствуют в модели, то значение сходства при этом не уменьшается. Для сравнения изображений могут быть определены и другие меры. Например, гистограммы можно было





**Рис. 6.11.** Цветные изображения и их 64-разрядные гистограммы. (С любезного разрешения А. Vailaya, Corel Stock Photos и Corbis.) См. цветную вклейку.

бы нормировать для преобразования в гистограммы частот путем деления счетчика пикселей в каждом кармане на общее количество пикселей. Затем два изображения можно было бы сравнить по критерию на основе евклидова расстояния.

$$\begin{aligned}
 \text{intersection}(h(I), h(M)) &= \sum_{j=1}^K \min\{h(I)[j], h(M)[j]\} \\
 \text{match}(h(I), h(M)) &= \frac{\sum_{j=1}^K \min\{h(I)[j], h(M)[j]\}}{\sum_{j=1}^K h(M)[j]}
 \end{aligned} \tag{6.4}$$

Эксперименты показали, что сходство гистограмм может успешно использоваться в качестве признака сходства изображений при наличии факторов изменчивости, перечисленных выше, а также при использовании различных параметров пространственной дискретизации изображений. Свейн (Swain) и Баллард (Ballard) также разработали алгоритм *обратного проектирования* (backprojection algorithm). Этот алгоритм обнаруживает на изображении область размерами примерно с модельный объект, для которой обеспечивается наилучшее совпадение гистограмм изображения и модели. Таким образом, они разработали два алгоритма обработки цветных изображений. Один из них

предназначен для проверки наличия известного объекта на изображении, а второй — для определения местоположения этого объекта. Если изображения были получены в условиях различного освещения, то в начале обработки необходимо исключить влияние интенсивности.

Альтернативный метод сравнения изображений предполагает сопоставление интегральных функций распределения, а не гистограмм частот.

---

**Упражнение 6.12.** Сопоставление изображений с перемещенными пикселями

Допустим, что по данному изображению  $A$  посредством случайных перестановок пикселей было построено изображение  $B$ . (Эту операцию можно выполнить следующим образом. Во-первых, скопируем изображение  $A$  в  $B$ . Затем для каждого пиксела  $I[r, c]$  изображения  $B$  случайным образом выберем пиксел  $I[x, y]$  и обменяем значения  $I[r, c]$  и  $I[x, y]$ .) Какие результаты можно ожидать при сопоставлении гистограмм изображений  $A$  и  $B$ ?

---

**Упражнение 6.13.** Распознавание фруктов и овощей

Возьмите 3 банана, 3 апельсина, 3 красных яблока, 3 зеленых яблока, 3 зеленых перца и 3 красных помидора. Для каждого из этих 6 наборов объектов получите три изображения, каждый раз меняя расположение трех объектов набора. В результате будут получены 18 изображений. Постройте цветовую гистограмму для каждого изображения. Первую гистограмму каждого набора используйте в качестве модели (всего будет 6 модельных гистограмм). Затем вычислите величину сходства каждой из 6 моделей и 12 другими гистограммами. Проанализируйте полученные результаты. Позволяют ли они разработать систему распознавания для использования в супермаркетах для распознавания продуктов на столе кассира?

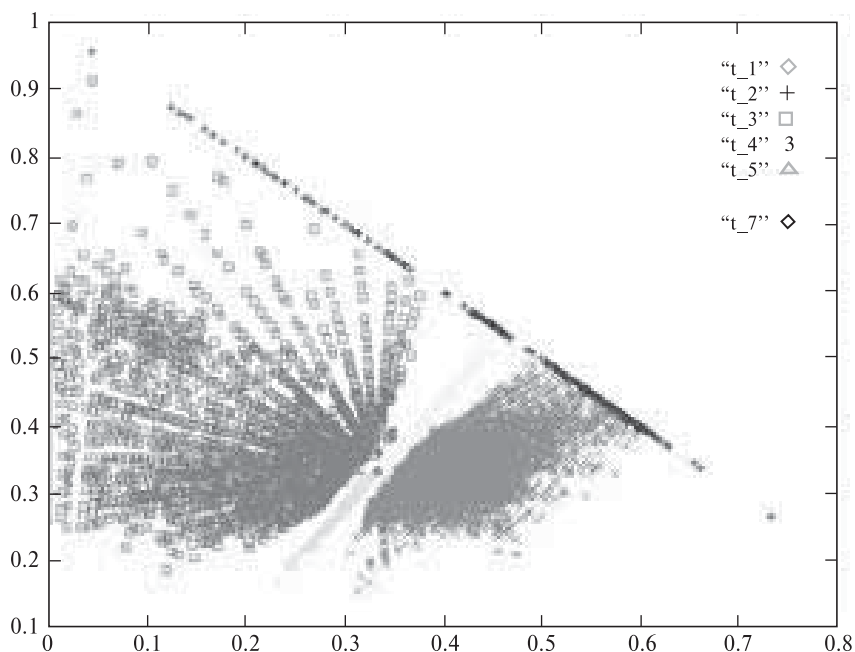
## 6.5 СЕГМЕНТАЦИЯ ЦВЕТНЫХ ИЗОБРАЖЕНИЙ

В данном разделе описывается задача поиска образа человеческого лица на цветном изображении. Для получения изображений используется малогабаритная камера, подключенная к персональному компьютеру. Результаты решения данной задачи могут применяться при проектировании новых человеко-машинных интерфейсов.

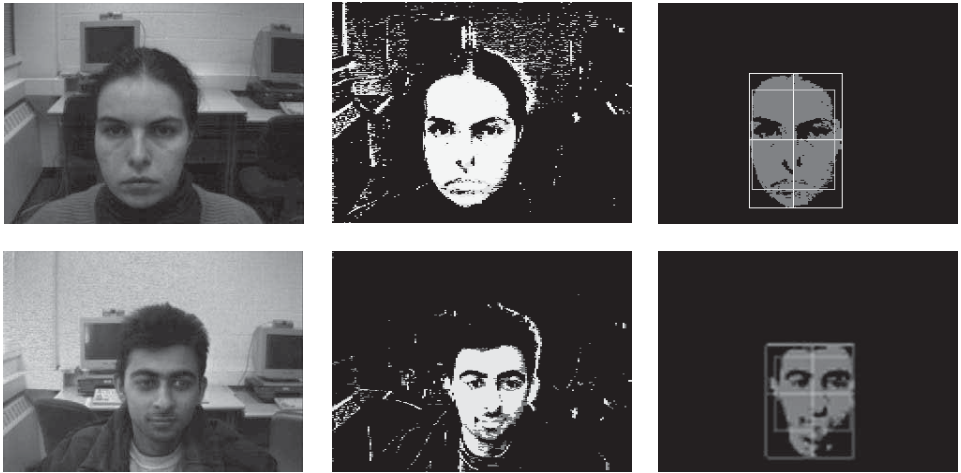
Далее в общих чертах рассматривается алгоритм поиска основной области изображения, соответствующей образу человеческого лица. Во-первых, для определения свойств пикселей образа лица на этапе обучения системы анализируются образцы, полученные из изображений лиц различных людей. Во-вторых, пиксели лица обнаруживаются на новом изображении по цветовым

значениям  $(r, g)$ , которые должны согласовываться с эталонными данными. На рис. 6.12 показан график с отмеченными точками  $(r, g)$ , взятыми с нескольких изображений человеческих лиц. Нормированные красная и зеленая цветовые координаты вычислены с помощью уравнений (6.1). На графике легко выделить 6 классов пикселей, границы между которыми будут определять границы принятия решений (с помощью методов, описанных в гл. 4). Три класса содержат пиксели образов лиц — один из этих классов соответствует основному цвету лица, а два других — теням или областям усов и бороды.

Операции по обнаружению области лица разделены на три основных этапа. На первом этапе в качестве входных данных используется маркированное изображение. Пикселям этого изображения метки  $1, 2, \dots, 7$  присвоены алгоритмом классификации, который учитывает эталонные образцы, запомненные на этапе обучения. (Метка 7 назначается пикселям, которые не принадлежат ни одному из 6 известных классов.) В центре на рис. 6.13 показаны маркированные изображения с лицами двух разных людей. На этих изображениях корректно



**Рис. 6.12.** Цветовые кластеры для образов человеческих лиц, сформированные на этапе обучения по эталонным данным. По горизонтальной оси  $R_{norm}$  измеряются нормированные красные координаты, а по вертикальной оси  $G_{norm}$  — нормированные зеленые координаты. Кластер с меткой  $t_4$  соответствует основному цвету образов человеческих лиц. Кластеры  $t_5$  и  $t_6$  представляют второстепенные цвета, связанные с затененными участками или областями усов и бороды. (С любезного разрешения V. Vakis.) См. цветную вклейку.



**Рис. 6.13.** Примеры обнаружения образов лиц. (Слева) входное изображение. (В центре) маркированное изображение. (Справа) границы выделенных областей, соответствующих образам лиц. (С любезного разрешения V. Vakis.) См. цветную вклейку.

маркированы большинство пикселей образов лиц и большинство пикселей фона. Тем не менее имеются небольшие области с ошибочными метками. При обработке компонент маркированного изображения эти компоненты объединяются или удаляются в соответствии с их размерами и расположением относительно основной области лица. В процессе обработки связанных компонент (см. гл. 3) учитываются только пиксели переднего плана с метками 4, 5 или 6.

На втором этапе в качестве образа лица выбирается наиболее подходящая компонента. На этом этапе также удаляются компоненты со слишком малыми или большими размерами. При оценке размеров применяются эвристические правила, сформулированные в результате анализа многих тестовых изображений. Обычно на изображении остается менее 100 компонент, большинство из которых принадлежат классам теневых областей.

На третьем этапе оставшиеся компоненты удаляются или объединяются с выбранной ранее областью лица. Применяются несколько эвристических правил относительно структуры образа лица. Также предполагается, что на изображении может находиться только одно лицо. Результаты обработки двух изображений показаны справа на рис. 6.13.

Программа, реализующая описанный алгоритм, обрабатывает до 30 изображений в секунду (выполняет обработку в режиме реального времени). В процессе обработки также определяется местоположение областей, соответствующих глазам и носу (в данном разделе эти алгоритмы не рассматриваются).

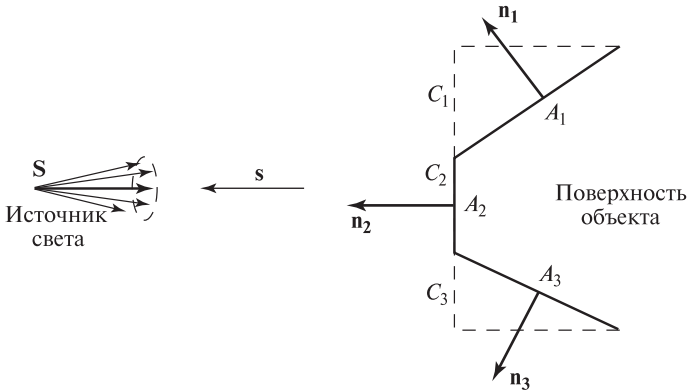
Представленный пример позволяет сделать выводы, применимые и ко многим другим задачам. Ключевой операцией является кластеризация тысяч цветковых кодов исходного изображения для получения маркированного изображения, содержащего всего несколько различных меток. В задаче обнаружения образов человеческих лиц кластеры были сформированы вручную на этапе обучения системы. В ряде задач кластеризация выполняется автоматически. Сегментация будет подробно рассматриваться в гл. 10.

## 6.6 ОСВЕЩЕННОСТЬ

Существует ряд факторов, усложняющих рассмотрение как физических моделей освещения, так и особенностей человеческого зрительного восприятия. Поверхности различаются зеркальными свойствами (т. е. насколько отражение света от этих поверхностей похоже на отражение от зеркала). Зеркальные поверхности отражают падающий свет в направлении, определяемом по закону зеркального отражения. Матовые поверхности отражают падающий свет равномерно по всем направлениям. Следовательно, поверхность характеризуется не только спектральным составом отраженного света, но и направлением отражения. Кроме того, энергия (интенсивность) излучения, падающего на поверхность, зависит от расстояния до источника света. Участки поверхности, расположенные дальше от точечного источника, будут получать меньше энергии, чем более близко расположенные участки. Аналогичное заключение можно сделать относительно светоизлучающего объекта и светочувствительных элементов датчика. В результате, значения интенсивности пикселей изображения будут распределены неравномерно из-за неодинаковых расстояний вдоль проектирующих лучей от элементов датчика до участков поверхностей. Ориентация поверхностного участка относительно источника света также имеет большое значение.

### 6.6.1 Излучение одиночного источника света

Рассмотрим световое излучение от одиночного удаленного источника света. Схема освещения поверхности объекта приведена на рис. 6.14. На этой схеме не указана точка наблюдения, с которой можно было бы видеть поверхность. Пока будем рассматривать лишь свойства света, падающего на поверхность. Будем полагать, что источник света располагается достаточно далеко и направление от всех поверхностных элементов освещаемого объекта до источника света можно представить одним единичным вектором  $\mathbf{s}$ . Световая энергия на единицу площади (*интенсивность  $i$* ), падающая на каждый поверхностный элемент  $A_j$ , пропорциональна площади проекции этого элемента на плоскость, перпендикулярную направлению падающего света  $\mathbf{s}$ . Для вычисления площади этой проекции надо умножить площадь элемента  $A_j$  на косинус угла между элементом и вектором  $\mathbf{s}$ . Косинус угла равен скалярному произведению  $\mathbf{n}\mathbf{s}$ , где  $\mathbf{n}$  — это единичный вектор нормали к плоскому поверхностному элементу  $A_j$ .



**Рис. 6.14.** На поверхностные элементы  $A_j$  падает свет от источника  $S$ . Попадающая на поверхностный элемент световая энергия пропорциональна проекции этого элемента, перпендикулярной направлению освещения. Площадь этой проекции равна  $C_j = A_j \cos \theta_j$ . Следовательно, интенсивность падающего света  $i \sim \mathbf{n} \circ \mathbf{s}$  где  $\mathbf{n}$  — единичный вектор нормали к поверхности, а  $\mathbf{s}$  — единичный вектор в направлении распространения света. Угол  $\theta_j$  равен углу между нормалью к поверхностному элементу  $\mathbf{n}_j$  и вектором  $\mathbf{s}$ .

Таким образом, интенсивность света, падающего на поверхностный элемент, можно представить в виде:

$$i \sim \mathbf{n} \circ \mathbf{s} \quad (6.5)$$

Энергия падающего на поверхность света прямо пропорциональна мощности источника света. Мощность источника света не всегда бывает известна. Свет от источника может распространяться по всем направлениям или как у прожектора, только в небольшом световом конусе. В любом случае, мощность источника света измеряется в ваттах настерадиан, или как энергия, проходящая через единичную площадь основания конического сектора единичной сферы, в центре которой располагается источник света.

Рассмотренная простая модель освещения плоских поверхностных элементов легко расширяется на случай криволинейных поверхностей. Для этого достаточно представить криволинейную поверхность в виде множества плоских элементов достаточно малого размера. Доля падающего излучения, которая отражается поверхностным элементом, называется *альбедо* этого элемента.

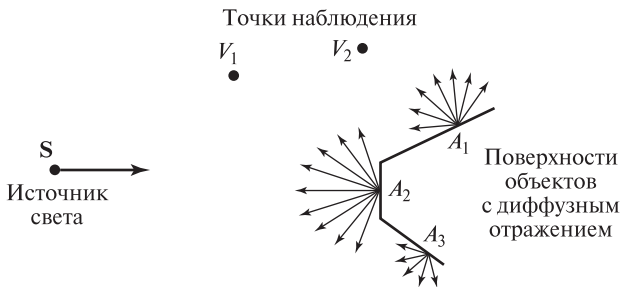
**65 Определение.** Альбедо поверхностного элемента равно отношению энергии отраженного света к энергии света, падающего на поверхность.

Не для всех поверхностей альбедо является их внутренним свойством. У некоторых поверхностей доля отраженного света зависит от угла между направлением на источник света и нормалью к поверхности.

### 6.6.2 Диффузное отражение

Теперь учтем в нашей модели освещения отражение света от поверхностей объектов и оценим вид поверхностных элементов при наблюдении из некоторой точки  $V$ . На рис. 6.15 поясняется *диффузное отражение*, или *отражение по закону Ламберта*. Свет, падающий на поверхностный элемент, равномерно отражается по всем направлениям в пределах полусферы, в центре которой находится данный поверхностный элемент. Диффузное отражение наблюдается у поверхностей, размеры рельефных шероховатостей на которых существенно больше длины световой волны. Интенсивность отраженного света пропорциональна интенсивности падающего света. Коэффициент пропорциональности равен альбедо поверхности. Альбедо темных поверхностей меньше, чем у светлых.

$$\text{интенсивность диффузно отраженного света } i \sim \mathbf{n}_j \circ \mathbf{s} \quad (6.6)$$



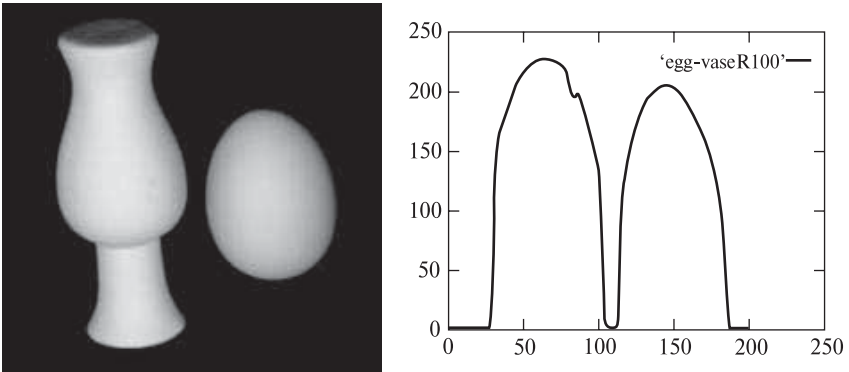
**Рис. 6.15.** При диффузном (ламбертовом) отражении свет отражается равномерно по всем направлениям внутри полусферы с центром в данном поверхностном элементе. Следовательно, весь этот элемент с любой точки наблюдения будет выглядеть как участок постоянной яркости.

**66 Определение.** Поверхности с **диффузным** отражением отражают свет равномерно по всем направлениям. Поэтому такие поверхности выглядят одинаково яркими со всех точек наблюдения.

Особое свойство диффузного отражения заключается в том, что поверхностный элемент выглядит одинаково ярким при наблюдении с любого направления, попадающего в полусферу. На рис. 6.15 яркости поверхностных элементов  $A_1$  и  $A_2$  будут одинаковыми при наблюдении из точек  $V_1$  и  $V_2$ . Если все три поверхностных элемента сделаны из одинакового материала, то их альбедо также будет иметь равные значения. Следовательно, элемент  $A_2$  будет выглядеть ярче, чем  $A_1$ , а элемент  $A_1$  будет ярче  $A_3$ . Различия в яркости объясняются значениями углов между этими элементами и направлением освещения. Поверхностный элемент  $A_3$  из точек наблюдения  $V_1$  и  $V_2$  вообще

не будет виден. (Элемент не виден, если  $\mathbf{n} \cdot \mathbf{v} < 0$ , где единичный вектор  $\mathbf{v}$  задает направление взгляда наблюдателя.)

Характерный пример диффузного отражения приведен на рис. 6.16. На этом рисунке показана интенсивность света, отраженного от матовой поверхности яйца и от белой керамической вазы. График интенсивностей пикселей из одной строки изображения напоминает график косинуса. Этот пример демонстрирует, что форма поверхности объекта тесно связана с интенсивностью отраженного света (это следует из уравнения (6.6)).



**Рис. 6.16.** Диффузное отражение от ламбертовых объектов — вазы и яйца. Справа приведен график интенсивностей пикселей вдоль выделенной строки. Форма кривой интенсивности связана с формой объекта. (С любезного разрешения Deborah Trytten.)

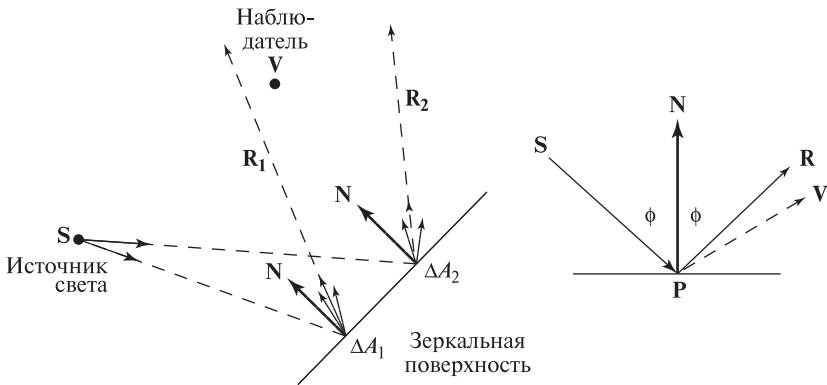
### Упражнение 6.14

Рассмотрим многогранный объект из диффузно отражающего материала. Одна из граней объекта  $F$  расположена прямо напротив удаленного источника света  $S$ . Другая грань  $A$ , смежная с гранью  $F$ , выглядит наполовину менее яркой, чем  $F$ . Чему равен угол между нормальными к граням  $A$  и  $F$ ?

### 6.6.3 Зеркальное отражение

Многие гладкие поверхности напоминают зеркало, отражая большую часть падающего света в направлении зеркального отражения (см. рис. 6.17). Зеркально отраженный луч ( $\mathbf{R}$ ) лежит в одной плоскости с нормалью ( $\mathbf{N}$ ) к поверхности и с падающим световым лучом ( $\mathbf{S}$ ). Углы между нормалью и лучами  $\mathbf{R}$  и  $\mathbf{S}$  равны. Идеальное зеркало будет отражать весь падающий свет от источника  $\mathbf{S}$  в направлении луча  $\mathbf{R}$ . Спектральный состав отраженного света будет совпадать с падающим, т. е. он не зависит от цвета поверхности объекта.





**Рис. 6.17.** При зеркальном отражении свет отражается в узком конусе вокруг луча зеркального отражения  $\mathbf{R}$ . В точку  $\mathbf{V}$  попадет некоторая часть отраженного света от поверхностного элемента  $\Delta A_1$ , и очень малая часть отраженного света от элемента  $\Delta A_2$ . Интенсивность отраженного света в точке  $\mathbf{V}$  выражается как  $e_i \sim (\mathbf{R} \circ \mathbf{V})^\alpha$ , где  $\mathbf{R}$  — направление зеркально отраженного луча,  $\mathbf{V}$  — направление от поверхностного элемента к точке наблюдения,  $\alpha$  — параметр, называемый *блеском* поверхности.

Например, на красном яблоке при освещении белым светом будет виден белый блик. Уравнение (6.7) является математическим выражением интенсивности зеркально отраженного света. Оно часто применяется в компьютерной графике.

Уравнение (6.8) позволяет вычислить направление отраженного луча  $\mathbf{R}$  по нормали к поверхности и направлению падающего луча. Параметр  $\alpha$ , называемый *блеском* (shininess) поверхности, для очень блестящих поверхностей имеет значение 100 и более. При увеличении  $\alpha$  функция  $\cos^\alpha \phi$  будет убывать сильнее по мере удаления  $\phi$  от 0.

$$\text{интенсивность зеркально отраженного света } i \sim (\mathbf{R} \circ \mathbf{V})^\alpha \quad (6.7)$$

$$\mathbf{R} = 2\mathbf{N}(\mathbf{N} \circ (-\mathbf{S})) \oplus \mathbf{S} \quad (6.8)$$

**67 Определение.** **Зеркальное отражение** характеризует зеркальные свойства поверхности. Зеркально отраженный от поверхности свет распространяется в узком конусе вокруг луча зеркального отражения. Спектральный состав отраженного и падающего света одинаков и не зависит от цвета поверхности.

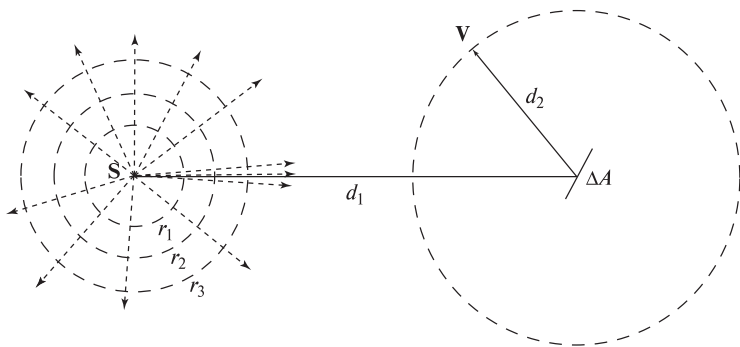
**68 Определение.** **Бликом** на поверхности объекта называется яркое пятно, образовавшееся вследствие зеркального отражения света от источника. Блики характерны для металлических, стеклянных, восковых и др. объектов.

### 6.6.4 Потемнение при увеличении расстояния

Интенсивность падающего на поверхность света уменьшается при увеличении расстояния между источником света и поверхностью. Очевидно, что Земля получает меньше солнечной световой энергии, чем Меркурий. На рис. 6.18 показана иллюстрирующая данный факт модель. Будем полагать, что источник излучает постоянный световой поток, т. е. одинаковое количество энергии в единицу времени. Площадь сферической поверхности растет пропорционально квадрату радиуса. Тогда количество энергии, приходящейся на единичную площадку сферической поверхности, уменьшается пропорционально квадрату радиуса. Следовательно, интенсивность света, падающего на поверхность объекта, будет уменьшаться пропорционально квадрату расстояния от источника света. На рис. 6.18 это расстояние обозначено  $d_1$ . Эта модель также применима к свету, отраженному от элементов поверхности объекта. Следовательно, наблюдатель в точке  $V$  будет видеть поверхность яркости ( $d_2$ ), которая обратно пропорциональна квадрату расстояния от поверхности. Данная модель часто используется в компьютерной графике для вычисления закраски поверхностей, чтобы отображаемые поверхности изображались менее яркими при увеличении глубины их элементов (расстояния от наблюдателя).

#### Упражнение 6.15

Некий изобретатель предложил дорожной полиции приобрести устройство для определения скорости автомобилей ночью. Устройство генерирует световые вспышки в моменты времени  $t_1$  и  $t_2$  и регистрирует отраженный от автомобиля свет. По величине интенсивности отраженного света устройство вычисляет



**Рис. 6.18.** Поток энергии излучения точечного источника света через любую сферическую поверхность вокруг источника имеет одинаковое значение. Следовательно, количество энергии на единичный элемент поверхности (интенсивность) должно быть обратно пропорционально квадрату радиуса сферы ( $d_1$ ). Аналогично, интенсивность света, отраженного от поверхностного элемента, убывает пропорционально квадрату расстояния ( $d_2$ ) от элемента до наблюдателя.

расстояния  $d_1$  и  $d_2$  для двух моментов времени  $t_1$  и  $t_2$  (с помощью модели, аналогичной рис. 6.18). По этим данным вычисляется скорость автомобиля. Сделайте замечания относительно этого устройства. Работоспособен ли данный подход?

### 6.6.5 Дополнительные факторы

При построении модели освещения для большинства поверхностей необходимо учитывать и диффузную, и зеркальную компоненты. Если осветить яблоко карманным фонарем, то мы увидим красный объект с белым бликом. Красный свет отражается диффузно, а блик образуется за счет зеркального отражения. Если при моделировании сцены с яблоком учитывалось бы только зеркальное отражение, то большая часть поверхности яблока была бы не видна.

Часто сцена освещается несколькими источниками света и содержит большое количество поверхностных элементов, отражающих свет от источников. Обмен световой энергией между всеми поверхностными элементами учесть сложно или даже невозможно. Многократно отраженный свет можно смоделировать за счет добавления в модель освещения *рассеянного света*. В компьютерной графике применение рассеянной компоненты освещения является стандартным приемом, используемым при закраске поверхностей.

**69 Определение.** **Рассеянный свет** представляет свет постоянной интенсивности в любой области сцены. Рассеянный свет учитывается при наличии нескольких источников света и многократных переотражений света от различных поверхностей.

Некоторые объекты сами излучают свет. Например, это электрическая лампочка или объект, поглощающий электромагнитное излучение невидимого диапазона и излучающий видимый свет. Такие объекты и отражают свет, и излучают. Характеристики излучения и отражения зависят от длины волны. Источники обычно излучают в некотором спектре длин волн (за исключением монохроматических источников, например, лазеров). Поверхности отражают или поглощают свет также преимущественно на некоторых определенных длинах волн. Возможна разработка датчиков, учитывающих такие особенности. Например, многоспектральные сканеры способны генерировать 200 значений интенсивности при восприятии света, отраженного от одиночного поверхностного элемента. Однако для просмотра человеком результаты восприятия видимого света можно представить с помощью всего трех значений (например, RGB или HSI). Модели освещения в компьютерной обычно описываются в системе RGB.

---

### Упражнение 6.16

Начинающий фотограф сфотографировал друзей на фоне Большого Каньона сразу после захода солнца. Хотя он пользовался вспышкой и друзья на фото-

графией получились хорошо, но стена каньона на фоне фотографии оказалась почти черной. Почему?

### 6.6.6 Модель освещения Фонга\*

Модель освещения Фонга часто применяется в компьютерной графике для закраски поверхностей. В ней учитываются несколько факторов: (а) рассеянный свет, (б) диффузное отражение, (в) зеркальное отражение и (г) потемнение при увеличении расстояния. Компоненты (б), (в) и (г) вычисляются для каждого источника света и суммируются. Будем полагать, что нам известны характеристики поверхностного элемента, изображаемого в пикселе  $\mathbf{I}[x, y]$ , а также местоположение и свойства всех источников света. Отражающие свойства поверхностного элемента представляются параметрами диффузного и зеркального отражения —  $K_{d\lambda}$  и  $K_{s\lambda}$ .  $K_{q\lambda}$  представляет собой вектор коэффициентов отражения для различных длин  $\lambda$  (обычно используются три коэффициента для основных цветов RGB).

$$I_{\lambda}[x, y] = I_{a\lambda}K_{d\lambda} + \sum_{m=1}^M \left( \frac{1}{cd_m^2} I_{m\lambda} [K_{d\lambda}(\mathbf{n} \circ \mathbf{s}) + K_{s\lambda}(\mathbf{R}_m \circ \mathbf{V})^{\alpha}] \right) \quad (6.9)$$

В уравнение (6.9) входит интенсивность рассеянного света  $I_{a\lambda}$  и интенсивности  $I_{m\lambda}$  для каждого из  $M$  источников света. Это уравнение можно считать векторным уравнением, которое можно одинаковым образом записать для каждой длины волны  $\lambda$ . Значение  $I_{a\lambda}$  равно интенсивности рассеянного света на длине волны  $\lambda$ , а  $I_{m\lambda}$  равно интенсивности  $m$ -го источника света на длине волны  $\lambda$ . Расстояние между  $m$ -м источником света и поверхностным элементом равно  $d_m$ . Направление луча  $m$ -го источника света, отраженного от данного поверхностного элемента, определяется вектором  $\mathbf{R}_m$ .

### 6.6.7 Учет освещенности зрительной системой человека

Нет сомнений, что человеческое восприятие формы трехмерных объектов связано с восприятием освещенности поверхностей. Выше были рассмотрены упрощенные модели освещения и отражения. Тем не менее описанные явления играют важнейшую роль при зрительном восприятии освещенности объектов. Упрощенные модели освещения очень важны для компьютерной графики. Для ускорения расчета изображений трехмерных освещенных сцен широко используются различные приближения.

В системах компьютерного зрения, наблюдающих сцены в условиях управляемого освещения, с помощью приведенных выше формул можно реализовать расчет формы поверхности по ее освещенности. Эти методы рассматриваются в гл. 13. Например, если бы мы определили коэффициенты пропорциональности в приведенных формулах, то смогли бы вычислить нормали к поверхностям

в точках, показанных на рис. 6.16. В сценах с неконтролируемым освещением, например, в природных условиях, учесть различные факторы освещения оказывается намного сложнее.

## **6.7 СМЕЖНЫЕ ТЕМЫ\***

### **6.7.1 Приложения**

Цветовые признаки упрощают решение ряда задач распознавания по сравнению с использованием только данных об интенсивности, текстуре или форме областей. Измерения цвета выполняются локально и в некоторых задачах устраняют необходимость применения методов накопления данных и анализа формы областей. В Упражнении 6.13 описан пример использования пиксельной цветовой информации для классификации фруктов и овощей. Алгоритм обнаружения лиц, рассмотренный в разд. 6.5, сначала обнаруживает пиксели телесного цвета по соответствию с эталонными данными. После этого выделяются области обнаруженных пикселей и вычисляются геометрические отношения между ними. Цветовая информация применяется для организации доступа к базам данных изображений (см. гл. 8) и при обработке биологических изображений, полученных с помощью микроскопа.

### **6.7.2 Человеческое цветовое восприятие**

Свойства человеческого цветового восприятия важны по двум основным причинам. Во-первых, зрительную систему человека часто можно рассматривать в качестве эффективной системы для изучения и моделирования. Во-вторых, основная цель применения компьютерной графики и графических дисплеев заключается в организации интерфейса между компьютерами и людьми. Специалисты по машинному зрению часто интересуются возможностями воспроизведения или замены отдельных способностей человека, а графические дизайнеры изучают способы наилучшего представления информации.

В целом, люди обладают субъективным цветовым восприятием. Например, для окраски стен помещений обычно выбираются ненасыщенные пастельные цвета; красный цвет стимулирует внимание, а голубой цвет на многих людей действует успокаивающе. Примерно 8 процентов людей страдают одной из разновидностей цветовой слепоты. Поэтому при проектировании интерфейсов следует аккуратно выбирать комбинации цветов.

На сетчатке человеческого глаза количество рецепторов, чувствительных к красному и зеленому свету, намного превышает количество синих рецепторов. Это особенно заметно в области фовеа, в которой синих рецепторов очень мало. В результате основная обработка цветовой информации выполняется нейронами, интегрирующими сигналы от рецепторов. Были предложены несколько теорий, объясняющих обработку цвета на основе нейронной обработки. В настоящее время пока не достигнуто удовлетворительного понимания этой

высокоуровневой обработки. Человеческое зрительное восприятие остается предметом непрекращающихся исследований.

Цвет отдельных пикселей дисплея человеком точно не воспринимается. Но люди способны устойчиво определять цвет участков поверхности даже при изменении условий освещенности, в том числе при освещении светом, содержащим только два основных цвета.

Во многих задачах методы обработки краев на изображении интенсивностей (гл. 5) позволяют распознать объект быстрее, чем методы обработки с использованием цветовой информации. В теоретических моделях зрительного восприятия цветовая обработка часто рассматривается как следующий уровень на базе более примитивной обработки значений интенсивности. Читатель может получить представление об обширной области исследований человеческого зрительного восприятия с помощью дополнительной литературы.

### 6.7.3 Многоспектральные изображения

Датчик, который выполняет три цветовых измерения для каждого пиксела, относится к категории многоспектральных датчиков (гл. 2). Однако интенсивность излучения можно измерять и за пределами видимого диапазона, например в инфракрасной области спектра.

На инфракрасных компонентах спутниковых фотографий нагретые асфальтовые дороги будут выглядеть яркими объектами, а холодные водные массивы — темными. Несколько результатов измерений для одного пиксела часто упрощают разработку процедур классификации изображений поверхностей. Аппаратная часть многоспектральной сканирующей системы может оказаться весьма дорогостоящей. Она требует тщательного проектирования и точного изготовления, чтобы от каждого наблюдаемого поверхностного элемента собиралось излучение из нескольких спектральных диапазонов.

Настройка параметров ЯМР-сканирования (см. гл. 2) позволяет получить набор трехмерных изображений, из которых для каждого воксела сканируемого объема можно извлечь  $m$  значений интенсивности. Эти измерения можно использовать для классификации материала воксела как жировой ткани, крови, мышечной ткани, и т. п. Получение трехмерного массива данных на ЯМР-томографе может занять около часа, поэтому в данных будет присутствовать шум из-за движения, в частности, вблизи границ различных тканей. В этих областях с течением времени датчик может сканировать различные ткани из-за их смещения вследствие дыхания пациента или движения крови.

### 6.7.4 Классификационные изображения

На *классификационных изображениях* (thematic images) свойства материала или объемных элементов сцены представляются с помощью *псевдоцветов*. Например, пиксеты карты или спутникового изображения для просмотра человеком могут быть маркированы так, чтобы реки были синими, городские

территории — пурпурными, а дороги — красными. Это не естественные цвета, зарегистрированные датчиком, а цвета, выбранные для удобства восприятия изображений человеком. На картах погоды высокие значения температуры часто отмечаются красным цветом, а низкие — синим. Аналогичным образом на классификационных изображениях можно представлять высоту поверхности, локальные геометрические свойства поверхностных элементов или их ориентацию, текстуру, плотность некоторой величины или любые другие скалярные измеряемые величины или результаты классификации.

В центре на рис. 6.13 показаны два классификационных изображения. Желтый, синий и пурпурный цвета используются в качестве меток трех кластеров в цветовом пространстве. Важно отметить, что на классификационных изображениях показываются не действительные физические данные, зарегистрированные датчиком, а преобразованные или классифицированные данные, подготовленные для просмотра человеком.

## 6.8 ЛИТЕРАТУРА

Подробное описание оптики и световых свойств можно найти в учебнике [Hecht, Zajac, 1974]. Практические вопросы, связанные с цифровым представлением цвета, рассматриваются в книге [Murray, VanRyper, 1994]. В этой книге дано детальное рассмотрение многих файловых форматов, применяемых для хранения цифровых изображений. Подробности устройства цветных дисплеев, особенно, масочных кинескопов, описаны в книге [Foley и др., 1996]. В книге [Levine, 1985] обсуждается несколько различных биологических зрительных систем. Более подробное описание приведено в книге [Overington, 1992], в которой принят подход на основе методов обработки сигналов. В качестве начального материала по психологии можно выбрать статью [Livingstone, 1988]. Приведенный в данной главе метод сопоставления цветовых гистограмм был опубликован в работе [Swain, Ballard, 1991] и Джейн и Вэйлая [Jain, Vailaya, 1996]. Подробное описание работы по обнаружению образов человеческих лиц содержится в отчете [Bakic, Stockman, 1999]. Статья [Taxt, Lundervold, 1994] посвящена многоспектральному анализу изображений мозга, полученных методом ЯМР-томографии.

1. Bakic, V., and G. Stockman. 1999. Menu selection by facial aspect. *Proc. Vision Interface '99* (19–21 May 1999). Trois Rivieres, Quebec.
2. Fleck, M., D. Forsyth, and C. Pregler. 1966. Finding naked people. *Proc. Euro. Conf. Comput. Vision*. Springer-Verlag, New York, 593–602.
3. Foley, J., A. van Dam, S. Feiner, and J. Hughes. 1996. *Computer Graphics: Principles and Practice*, 2nd Ed in C. Addison-Wesley, New York.
4. Hecht, E., and A. Zajac. 1974. *Optics*. Addison-Wesley, New York.
5. Jain, A., and A. Vailaya. 1996. Image retrieval using color and shape. *Pattern Recog.*, v. 29(8):1233–1244.

6. Levine, M. 1985. *Vision in Man and Machine*. McGraw-Hill, New York.
7. Livingstone, M. 1988. Art, illusion and the visual system. *Sci. Am.* (Jan. 1988), 78–85.
8. Murray, J., and W. VanRyper. 1994. *Encyclopedia of Graphical File Formats*. O'Reilly and Associates, Sebastopol, CA.
9. Overington, I. 1992. *Computer Vision: A Unified, Biologically-Inspired Approach*. Elsevier, Amsterdam.
10. Swain, M., and D. Ballard. 1991. Color indexing. *Inter. J. Comput. Vision*, v. 7(1): 11–32.
11. Taxt, T., and A. Lundervold. 1994. Multispectral analysis of the brain in magnetic resonance imaging. *Proc. IEEE Workshop on Biomed. Image Anal.* (24–25 June 1994), Seattle, 33–42.



Текстура является еще одним характерным признаком, применяемым для сегментации изображений на области интереса и для классификации этих областей. На изображениях некоторых типов текстура может определять свойства областей, критически важные для корректного выполнения анализа. Изображение на рис. 7.1 содержит три сильно отличающиеся текстуры, соответствующие образам тигра, джунглей и воды. Эти текстуры можно обнаружить и использовать для идентификации классов объектов, представленных данными текстурами.

Текстура описывает пространственное распределение цветов или значений интенсивности на изображении. Допустим, что гистограмма некоторой области показывает наличие в данной области 50% белых пикселей и 50% черных. На рис. 7.2 приведены изображения трех различных областей с таким распределением значений интенсивности. Эти области могли бы рассматриваться в качестве трех различных текстур. Левое изображение состоит из двух больших блоков — белого и черного. На изображении в центре есть восемнадцать маленьких белых клеток и восемнадцать черных клеток, расположенных в шахматном порядке. На правом изображении содержатся 6 блоков в виде чередующихся белых и черных полос.

Изображения мозаик из черных и белых прямоугольников на рис. 7.2 были созданы искусственно. Текстуры часто присутствуют на изображениях естественных сцен, содержащих как природные, так и изготовленные человеком объекты. Песок, камни, трава, листва, кирпичи и многие другие объекты формируют текстурное содержание изображения. На рис. 7.3 приведены примеры нескольких природных текстур. Отметим, что две показанные текстуры кирпичей и две текстуры листвы сильно различаются между собой. Следовательно, описание текстур не ограничивается задачами классификации объектов.

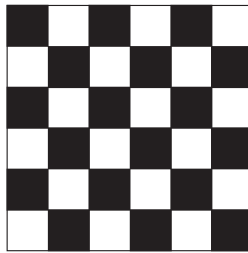
В данной главе обсуждается понятие текстуры, способы представления и вычисления текстурных характеристик, а также их применение в задачах анализа изображений.



**Рис. 7.1.** Изображение, содержащее несколько различных областей с разными текстурами. (С любезного разрешения Corel Stock Photos.)



Два блока



Шахматная доска



Чередующиеся полосы

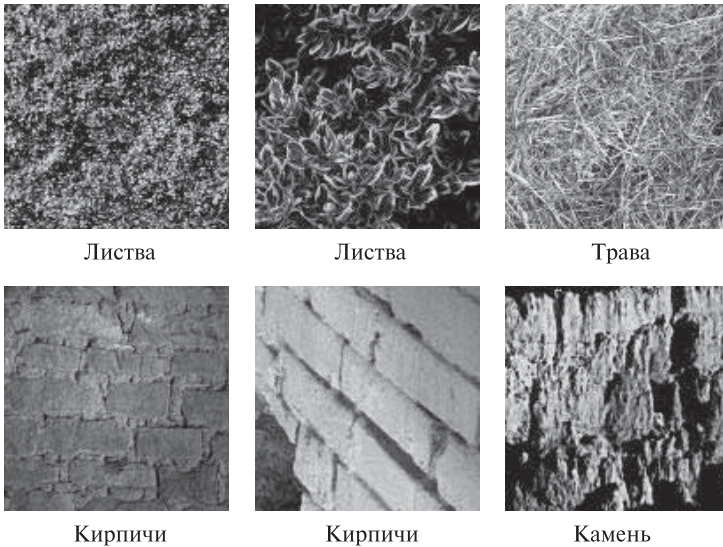
**Рис. 7.2.** Три разных текстуры с одинаковым частотным распределением черных и белых пикселов.

## 7.1 ТЕКСТУРА, ТЕКСЕЛЫ И ТЕКСТУРНЫЕ СТАТИСТИКИ

Искусственные текстуры на рис. 7.2 состоят из прямоугольных областей черного или белого цвета. На шахматной текстуре области выглядят как маленькие квадраты, расположенные с чередованием цвета на двумерной сетке. Области полосатой текстуры имеют вид длинных полос с чередованием цвета в вертикальном направлении. Эти одноцветные области легко сегментировать и затем произвести распознавание соответствующих простых образов.

Теперь рассмотрим две текстуры листвы на рис. 7.3. На первой текстуре присутствует большое количество маленьких, округлых листьев, а на второй — меньшее число крупных заостренных листьев. Нерегулярное пространственное расположение листьев сложно описать словами. Однако на изображениях заметны некоторые свойства, которые позволяют говорить о наличии характерного порядка на каждом изображении.

Одна из задач текстурного анализа заключается в точном определении понятия текстуры. Известны два основных подхода:



**Рис. 7.3.** Природные текстуры. (С любезного разрешения MIT Media Lab VisTex: <http://vismod.www.media.mit.edu/vismod/imagery/VisionTexture/vistex.html>.)

1. **Структурный подход:** Текстура представляет собой множество примитивных *текселов*, расположенных в некотором регулярном или повторяющемся порядке.
2. **Статистический подход:** Текстура является количественной характеристикой распределения значений интенсивности в области изображения.

Первый подход может показаться привлекательным и в действительности оказывается применимым для регулярных искусственных образов. Но на практике чаще применяется более общий второй подход. Он также оказывается более простым с точки зрения организации вычислений.

## 7.2 ТЕКСЕЛЬНОЕ ОПИСАНИЕ ТЕКСТУР

Текстуру можно рассматривать как множество элементарных текселов (texel), расположенных в заданном пространственном порядке. Тогда структурное описание текстуры могло бы состоять из описания текселов и их пространственной взаимосвязи. Конечно, текселы должны поддаваться сегментации, и должен быть известен вычислительный метод для определения пространственной взаимосвязи текселов.

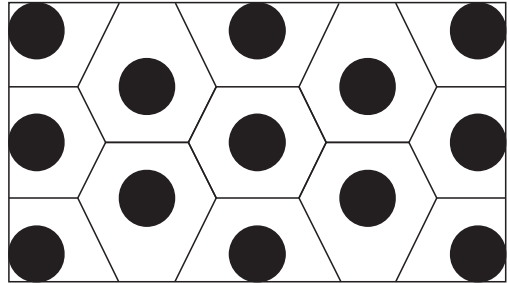
Один из вариантов структурного геометрического описания текстур был предложен Тусерьяном и Джейн [Tusseryan, Jain, 1990]. В качестве текселов выбираются области изображений, которые можно обнаружить с помощью про-

стных операций обработки наподобие пороговой бинаризации. Характеристики пространственной взаимосвязи текселов определяются на основе построенной для них диаграммы Вороного (см. далее).

Предположим, что имеется множество обнаруженных текселов, и каждый тексел можно представить одной характерной точкой, например, центром тяжести тексела. Обозначим множество этих точек символом  $S$ . Для любой пары точек  $P$  и  $Q$  из множества  $S$  можно построить перпендикуляр, делящий пополам отрезок  $PQ$ . Этот серединный перпендикуляр разбивает плоскость на две полуплоскости. Точки одной полуплоскости лежат ближе к точке  $P$ , а точки другой — ближе к точке  $Q$ . Обозначим через  $H^Q(P)$  полуплоскость точек, более близких к точке  $P$  по отношению к серединному перпендикуляру отрезка  $PQ$ . Описанную операцию можно повторить для каждой точки  $Q$  из множества  $S$ . *Многоугольником Вороного* для точки  $P$  называется многоугольник, содержащий все точки, более близкие к  $P$ , чем к любой другой точке множества  $S$ . Многоугольник Вороного определяется выражением:

$$V(P) = \bigcap_{Q \in S, Q \neq P} H^Q(P)$$

На рис. 7.4 показаны многоугольники Вороного для множества круговых текселов. Многоугольники Вороного внутренних текселов имеют вид шестиугольников, а для граничных текселов имеют иную форму.



**Рис. 7.4.** Диаграмма Вороного для множества круговых текселов.

После того как на изображении были выделены текселы и для них определены многоугольники Вороного, можно вычислить признаки формы многоугольников и использовать эти свойства для группировки многоугольников в кластеры. Кластеры будут задавать области изображения с однородной текстурой. Шаблон, подобный показанному на рис. 7.4, в случае распространения на изображение большего размера, мог бы соответствовать некоторой области с однородной текстурой, для описания которой применимы признаки формы правильных шестиугольников.

### Упражнение 7.1. Тексельное описание текстур

Получите или найдите готовый набор из пяти изображений, содержащих текстуры с очевидно заметными текселями, которые можно обнаружить с помощью простых операций наподобие пороговой бинаризации в полутоновом или цветовом диапазоне. Попытайтесь найти как минимум одну текстуру, содержащую текселя нескольких типов. Постройте диаграмму Вороного для небольшой области этого изображения.

## 7.3 КОЛИЧЕСТВЕННЫЕ ХАРАКТЕРИСТИКИ ТЕКСТУР

Обнаружить текселя на естественных изображениях обычно намного сложнее, чем на искусственно сгенерированных образах. Вместо обнаружения текселов по пиксельным данным полутоновых (или цветных) изображений можно вычислить численные характеристики текстур (текстурные статистики). По сравнению со структурным подходом данный подход менее нагляден, но обладает вычислительными преимуществами и может применяться как для сегментации, так и для классификации текстур.

### 7.3.1 Плотность и направление краев

Выделение краев представляет собой хорошо известный и простой в применении способ обнаружения характерных признаков. Поэтому естественно было бы попробовать использовать детектор краев на начальном этапе текстурного анализа. Количество краев в заданной области фиксированного размера является некоторой характеристикой заполнения этой области. В процессе выделения краев обычно вычисляются их направления. Эти значения также могут применяться для описания образа текстуры.

Рассмотрим область из  $N$  пикселей. Предположим, что в результате обработки этой области градиентным детектором краев для каждого пиксела  $p$  были вычислены два значения: 1) модуль (величина) градиента  $Mag(p)$  и 2) направление градиента  $Dir(p)$  (см. гл. 5). Одним из простейших характерных признаков текстуры является *плотность краев* (количество краев на единицу площади):

$$F_{edgeness} = \frac{|\{p \mid Mag(p) \geq T\}|}{N} \quad (7.1)$$

где  $T$  — некоторое пороговое значение. Плотность краев характеризует заполнение текстуры, но не ее ориентацию.

Полученную текстурную характеристику можно расширить, чтобы учесть и заполнение текстуры, и свойства ее ориентации. Для этого воспользуемся гистограммами модуля и направления градиента. Обозначим  $H_{mag}(R)$  нормированную гистограмму, построенную по значениям модуля градиента в пикселах области  $R$ , а  $H_{dir}$  — нормированную гистограмму направления градиента. Обе

гистограммы обычно содержат небольшое фиксированное количество разрядов (порядка 10), соответствующих диапазонам значений модуля и направления градиента. Обе гистограммы нормируются посредством деления на размер  $N_R$  области  $R$ . Тогда количественное описание текстуры области  $R$  имеет вид пары гистограмм:

$$F_{mag\ dir} = (H_{mag}(R), H_{dir}(R)) \quad (7.2)$$

Рассмотрим два изображения размерами  $5 \times 5$  на рис. 7.5. Изображение слева заполнено сильнее, чем изображение справа. На левом изображении края присутствуют во всех 25 пикселах, так что плотность краев равна 1.0. Правое изображение содержит 6 краев на 25 пикселах, таким образом плотность краев равна 0.24.

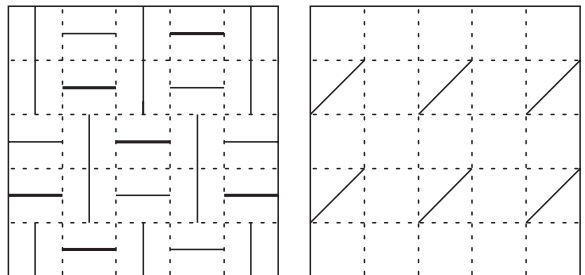
Будем строить двухразрядные гистограммы модуля градиента. Два разряда соответствуют темным и светлым краям. На гистограммах направления градиента выберем три разряда для горизонтальных, вертикальных и диагональных краев. На изображении слева присутствуют 6 темных и 19 светлых краев. Нормированная гистограмма модуля градиента для этого изображения имеет вид  $(0.24, 0.76)$ . Это значит, что 24% краев являются темными, а 76% — светлыми. На левом изображении есть 12 горизонтальных и 13 вертикальных краев, а диагональных краев нет. Нормированная гистограмма направления градиента будет иметь вид  $(0.48, 0.52, 0.0)$  — 48% горизонтальных краев, 52% вертикальных, и 0% диагональных.

На правом изображении содержится 6 светлых краев, а темных краев нет. Нормированная гистограмма модуля градиента равна  $(0.0, 0.24)$ . На этом изображении есть только диагональные края, поэтому гистограмма направления градиента имеет вид  $(0.0, 0.0, 0.24)$ .

В рассмотренном примере для различения двух изображений было бы достаточно признака плотности краев, но гистограммные признаки являются существенно более мощным методом описания текстур.

Для сравнения двух  $n$ -разрядных гистограмм  $H_1$  и  $H_2$  можно вычислить расстояние  $L_1$  между гистограммами:

$$L_1(H_1, H_2) = \sum_{i=1}^n |H_1[i] - H_2[i]| \quad (7.3)$$



**Рис. 7.5.** Два изображения с различными текстурными характеристиками плотности и направления краев.

### Упражнение 7.2. Краевые характеристики текстуры

Получите набор изображений, содержащих достаточное количество искусственных структур с явно заметными краями. Разработайте программу для вычисления текстурной характеристики  $F_{mag\ dir}$  согласно уравнению (7.2). Вычислите эту характеристику для каждого изображения. Сравните изображения по расстоянию  $L_1$  из уравнения (7.3).

### 7.3.2 Локальное двоичное разбиение

Локальное двоичное разбиение (local binary partition, LBP) представляет собой еще одну простую, но практически полезную текстурную характеристику. На изображении выполним сравнение значения интенсивности каждого пиксела  $p$  и восьми его соседей. Результаты проверки 8 соседей можно представить в виде 8-разрядного двоичного числа  $b_1b_2b_3b_4b_5b_6b_7b_8$ . Бит  $b_i = 0$  если интенсивность  $i$ -го соседа меньше или равна интенсивности  $p$ . В противном случае  $b_i = 1$ . По этим числам можно построить гистограмму и использовать ее в качестве описания текстуры изображения. Для сравнения двух изображений или областей подходит критерий на основе расстояния  $L_1$  между двумя гистограммами (см. выше).

### Упражнение 7.3. Текстурная характеристика «локальное двоичное разбиение»

Напишите программу для вычисления текстурной характеристики локального двоичного разбиения. Обработайте с помощью этой программы изображения из предыдущего упражнения. Вычислите расстояния  $L_1$  между парами изображений с использованием вычисленных характеристик. Сравните полученные результаты с результатами из предыдущего упражнения.

### 7.3.3 Вычисление характерных признаков на основе матрицы вхождений

*Матрица вхождений* (co-occurrence matrix) — это двумерный массив  $\mathbf{C}$ , в котором индексы строк и столбцов образуют множество  $\mathbf{V}$  допустимых на изображении значений пикселей. Например, для полутоновых изображений  $\mathbf{V}$  может быть множеством допустимых значений интенсивности (яркости), а для цветных изображений  $\mathbf{V}$  может быть множеством возможных цветов. Значение  $\mathbf{C}(\mathbf{i}, \mathbf{j})$  указывает, сколько раз значение  $\mathbf{i}$  встречалось на изображении в некотором заданном пространственном отношении со значением  $\mathbf{j}$ . Например, в качестве пространственного отношения может быть выбрано «значение  $\mathbf{i}$  является правым соседом для значения  $\mathbf{j}$ ».

Рассмотрим поясняющий пример, в котором множество  $\mathbf{V}$  представляет собой множество значений яркости, а пространственное отношение описывается

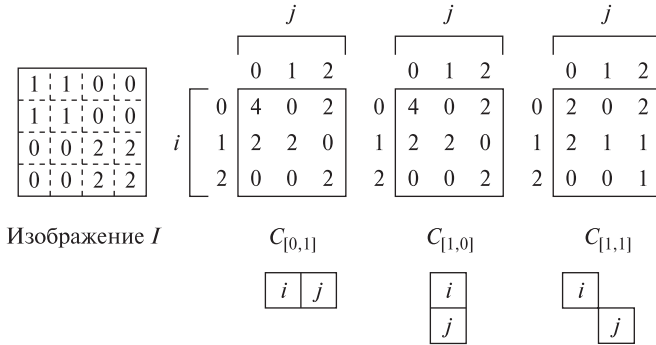


Рис. 7.6. Три различные матрицы вхождений для полутонового изображения.

вектором  $\mathbf{d}$ . Этот вектор задает смещение между пикселом с яркостью  $\mathbf{i}$  и пикселом с яркостью  $\mathbf{j}$ .

Обозначим через  $\mathbf{d}$  вектор перемещения  $(\mathbf{dr}, \mathbf{dc})$ , где  $\mathbf{dr}$  соответствует перемещению в направлении строк (вниз по изображению), а  $\mathbf{dc}$  — перемещение в направлении столбцов (вправо по изображению). Обозначим через  $\mathbf{V}$  множество значений яркости. Полутоновая матрица вхождений  $\mathbf{C}_{\mathbf{d}}$  для изображения  $\mathbf{I}$  определяется выражением:

$$C_{\mathbf{d}}[i, j] = |\{[r, c] \mid I[r, c] = i \text{ и } I[r + \mathbf{dr}, c + \mathbf{dc}] = j\}| \quad (7.4)$$

На рис. 7.6 показаны три различные матрицы вхождений  $\mathbf{C}_{[0,1]}$ ,  $\mathbf{C}_{[1,0]}$  и  $\mathbf{C}_{[1,1]}$  для одного и того же изображения  $\mathbf{I}$  размерами  $4 \times 4$ .

В матрице  $\mathbf{C}_{[0,1]}$  элемент  $[1, 0]$  равен 2. Это значит, что на изображении значение  $\mathbf{j} = 0$  дважды встречается в качестве правого соседа значения  $\mathbf{i} = 1$ . Элемент  $[0, 1]$  равен 0. Это значит, что значение  $\mathbf{j} = 1$  ни разу не встречается в качестве соседа пиксела  $\mathbf{i} = 0$ . Наибольшее количество вхождений, равное 4, хранится в элементе  $[0, 0]$ . Следовательно, на изображении нулевой пиксел является правым соседом другого нулевого пиксела в четырех различных местах.

#### Упражнение 7.4. Матрицы вхождений

Постройте полутоновые матрицы вхождений  $\mathbf{C}_{[1,2]}$ ,  $\mathbf{C}_{[2,2]}$  и  $\mathbf{C}_{[2,3]}$  для изображения, показанного на рис. 7.6.

Существуют две основные разновидности стандартной полутоновой матрицы вхождений. Первая называется *нормированной* полутоновой матрицей вхождений  $\mathbf{N}_{\mathbf{d}}$  и определяется выражением:

$$N_{\mathbf{d}}[i, j] = \frac{C_{\mathbf{d}}[i, j]}{\sum_i \sum_j C_{\mathbf{d}}[i, j]} \quad (7.5)$$



Вычисленные, согласно этому выражению значения вхождений принадлежат диапазону  $[0, 1]$ . Их можно рассматривать в качестве значений вероятности в матрице большего размера.

Вторая разновидность называется *симметричной* полутоновой матрицей вхождений  $S_d$ . В этой матрице группируются пары симметричных отношений смежности:

$$S_d[i, j] = C_d[i, j] + C_{-d}[i, j] \quad (7.6)$$

---

### Упражнение 7.5. Нормированная матрица вхождений

Вычислите нормированную матрицу вхождений  $N_{[1,1]}$  для изображения, показанного на рис. 7.7. Черным пикселям соответствуют значения яркости 0, серым пикселям — значения 1, а белым — значения 2. Каким образом полученная матрица может использоваться в качестве представления текстурного образа?

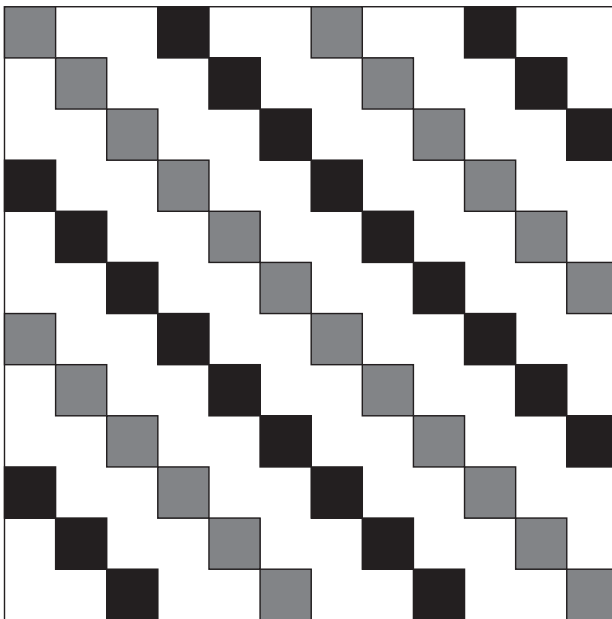


Рис. 7.7. Изображение с диагональной текстурой.

Матрицы вхождений представляют свойства текстуры, но они не удобны для непосредственного применения при анализе изображений, например, для сравнения двух текстур. Вместо этого матрицы вхождений используются для вычисления числовых характерных признаков, которые могут служить более

компактным представлением текстуры. На основе нормированной матрицы вхождений можно вычислить следующие характерные признаки:

$$\text{Энергия} = \sum_i \sum_j N_d^2[i, j] \quad (7.7)$$

$$\text{Энтропия} = - \sum_i \sum_j N_d[i, j] \log_2 N_d[i, j] \quad (7.8)$$

$$\text{Контраст} = \sum_i \sum_j (i - j)^2 N_d[i, j] \quad (7.9)$$

$$\text{Однородность} = \sum_i \sum_j \frac{N_d[i, j]}{1 + |i - j|} \quad (7.10)$$

$$\text{Корреляция} = \frac{\sum_i \sum_j (i - \mu_i)(j - \mu_j) N_d[i, j]}{\sigma_i \sigma_j} \quad (7.11)$$

где  $\mu_i, \mu_j$  — средние значения, а  $\sigma_i, \sigma_j$  — среднеквадратические ожидания сумм  $\mathbf{N}_d[\mathbf{i}]$  и  $\mathbf{N}_d[\mathbf{j}]$ , вычисленных в направлении строк и столбцов изображения:

$$N_d[i] = \sum_j N_d[i, j]$$

$$N_d[j] = \sum_i N_d[i, j]$$

Одна из проблем при вычислении текстурных характеристик на основе матриц смежности заключается в выборе вектора смещения  $\mathbf{d}$ . В работе [Zucker, Terzopoulos, 1980] было предложено использовать статистическую проверку  $\chi^2$  (хи-квадрат) для выбора значения (или значений)  $\mathbf{d}$ , соответствующих наиболее выраженной структуре текстурного образа. Эти значения соответствуют максимуму величины

$$\chi^2(d) = \left( \sum_i \sum_j \frac{N_d^2[i, j]}{N_d[i] N_d[j]} - 1 \right)$$

### 7.3.4 Энергетические текстурные характеристики Лавса

Еще один подход к вычислению текстурных характеристик предполагает обнаружение различных типов текстур с помощью локальных масок. Лавс (Laws) разработал энергетический подход, в котором оценивается изменение содержания текстуры в пределах окна фиксированного размера. Для вычисления энергетических характеристик используется набор из девяти масок размерами  $5 \times 5$ . Затем энергетические характеристики каждого пиксела анализируемого изображения представляются в виде вектора из 9 чисел. Для вычисления

масок используются следующие векторы (похожие на те, что рассматривались в гл. 5):

$$\begin{aligned} L5 \text{ (Level)} &= [1 \quad 4 \quad 6 \quad 4 \quad 1] \\ E5 \text{ (Edge)} &= [-1 \quad -2 \quad 0 \quad 2 \quad 1] \\ S5 \text{ (Spot)} &= [-1 \quad 0 \quad 2 \quad 0 \quad -1] \\ R5 \text{ (Ripple)} &= [1 \quad -4 \quad 6 \quad -4 \quad 1] \end{aligned}$$

Названия векторов описывают их назначение. Вектор L5 предназначен для вычисления симметричного взвешенного локального среднего значения. Вектор E5 предназначен для обнаружения краев, S5 — для обнаружения пятен (spot), а R5 — для обнаружения образа в виде ряби (ripples). Двумерные маски вычисляются умножением пар векторов. Например, для получения маски E5L5 надо умножить вектор E5 на L5:

$$\begin{bmatrix} -1 \\ -2 \\ 0 \\ 2 \\ 1 \end{bmatrix} \times [1 \quad 4 \quad 6 \quad 4 \quad 1] = \begin{bmatrix} -1 & -4 & -6 & -4 & -1 \\ -2 & -8 & -12 & -8 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 8 & 12 & 8 & 2 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

На первом шаге в методе Лавса [Laws, 1980] устраняется влияние интенсивности освещения. Для этого по изображению перемещается небольшое окно и из значения каждого пиксела вычитается локальное среднее значение. В результате данной предварительной обработки формируется изображение, на котором средняя интенсивность каждой пиксельной окрестности близка к 0. Размер окна зависит от типа изображений. Для естественных сцен было выбрано окно размерами  $15 \times 15$ .

После предварительной обработки к полученному изображению применяется каждая из шестнадцати масок  $5 \times 5$ . В результате формируются шестнадцать профильтрованных изображений. Обозначим через  $F_k[i, j]$  результат фильтрации пиксела  $[i, j]$  с использованием  $k$ -й маски. Тогда энергетическая текстурная карта  $E_k$  для фильтра  $k$  определяется выражением:

$$E_k[r, c] = \sum_{j=c-7}^{c+7} \sum_{i=r-7}^{r+7} |F_k[i, j]| \quad (7.12)$$

Каждая текстурная энергетическая карта является полноразмерным изображением, которое представляет результаты обработки входного изображения с использованием  $k$ -й маски.

После получения шестнадцати энергетических карт некоторые симметричные пары комбинируются и в результате строятся девять окончательных карт. Каждая симметричная пара карт заменяется усредненной картой. Например, маска E5L5 характеризует содержание горизонтальных краев, а L5E5 — вертикальных. Среднее двух этих карт будет характеризовать наличие на изображении краев обоих типов. Перечислим девять окончательных энергетических карт:

L5E5/E5L5	L5S5/S5L5
L5R5/R5L5	E5E5
E5S5/S5E5	E5R5/R5E5
S5S5	S5R5/R5S5
R5R5	

Девять изображений энергетических карт можно рассматривать в качестве одного изображения, в котором каждому пикселу соответствует вектор из 9 текстурных атрибутов. В табл. 7.1 показаны 9 текстурных атрибутов для основной текстуры каждого из изображений травы/камней/кирпичей на рис. 7.3. Эти текстурные атрибуты можно использовать для кластеризации изображения на области с одинаковой текстурой. На рис. 7.8 показаны результаты сегментации на кластеры нескольких многотектурных изображений.

### 7.3.5 Автокорреляция и спектр мощности

Автокорреляционную функцию изображения можно использовать для обнаружения повторяющихся образов текстурных элементов и для описания мелкозернистости/грубости текстуры. Автокорреляционная функция  $\rho(\mathbf{dr}, \mathbf{dc})$  изображения размерами  $(N + 1) \times (N + 1)$  для вектора смещения  $\mathbf{d} = (\mathbf{dr}, \mathbf{dc})$  выражается следующим образом (см. гл. 5):

$$\rho(dr, dc) = \frac{\sum_{r=0}^N \sum_{c=0}^N I[r, c] I[r + dr, c + dc]}{\sum_{r=0}^N \sum_{c=0}^N I^2[r, c]} \quad (7.13)$$

$$= \frac{I[r, c] \circ I_d[r, c]}{I[r, c] \circ I[r, c]} \quad (7.14)$$

#### Упражнение 7.6. Энергетические текстурные характеристики Лавса

Напишите программу для вычисления энергетических текстурных характеристик Лавса для полутонового изображения. Программа должна генерировать 9 изображений, по одному для каждой характеристики. Получите набор изображений с искусственными и природными текстурами и выполните с ними ряд экспериментов.

В каждом эксперименте выберите одно из изображений в качестве *тестового*, а остальные рассматривайте в качестве *базы данных изображений*. Напишите интерактивную оболочку, с помощью которой пользователь может указать некоторый пиксел на тестовом изображении. Затем должен выполняться поиск в базе данных таких изображений, которые где-либо имеют такую же текстуру, что и тестовое изображение в окрестности указанного пиксела. Для поиска используйте расстояние  $L_1$ , вычисляемое для набора 9 текстурных характеристик, известных для каждого пиксела.



(a) Исходное изображение



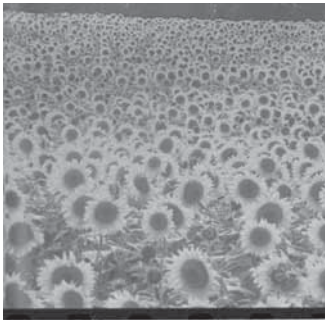
(b) Сегментация на 4 кластера



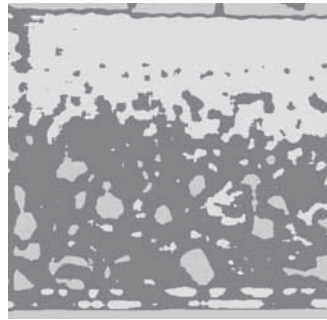
(c) Исходное изображение



(d) Сегментация на 4 кластера



(e) Исходное изображение



(f) Сегментация на 3 кластера

**Рис. 7.8.** Результаты сегментации нескольких изображений с использованием энергетических текстурных характеристик Лавса. (С любезного разрешения Corel Stock Photos и MIT Media Lab VisTex.) См. цветную вклейку.

При лобовом поиске можно просто сравнивать 9 значений выбранного пиксела тестового изображения с 9 значениями каждого пиксела всех изображений базы данных. По результатам сравнения выбирается то изображение, на котором есть пикселы с достаточно близкими к искомым текстурными характеристиками. Как реализовать поиск более эффективно?

**Таблица 7.1.** Энергетические текстурные характеристики Лавса для изображений, показанных на рис. 7.3

Изображение	E5E5	S5S5	R5R5	E5L5	S5L5	R5L5	S5E5	R5E5	R5S5
Листва1	250.9	140.0	1309.2	703.6	512.2	1516.2	187.5	568.8	430.0
Листва2	257.7	121.4	988.7	820.6	510.1	1186.4	172.9	439.6	328.0
Трава	197.8	107.2	1076.9	586.9	410.5	1208.5	144.0	444.8	338.1
Кирпичи1	128.1	60.2	512.7	442.1	273.8	724.8	86.6	248.1	176.3
Кирпичи2	72.4	28.6	214.2	263.6	130.9	271.5	43.2	93.3	68.5
Камень	224.6	103.2	766.8	812.8	506.4	1311.0	150.4	413.5	281.1

В случае грубого текстурного образа (крупнодетальной текстуры) автокорреляционная функция будет медленно убывающей. Автокорреляционная функция мелкозернистых текстур оказывается быстро убывающей. На графике автокорреляционных функций регулярных текстур присутствуют пики и впадины. Так как значения  $\mathbf{I}[\mathbf{r} + \mathbf{dr}, \mathbf{c} + \mathbf{dc}]$  вблизи краев изображения не определены, то необходимо предусмотреть способ вычисления значений этих виртуальных пикселей изображения.

Автокорреляционная функция связана со спектром мощности преобразования Фурье. Если  $\mathbf{I}[\mathbf{r}, \mathbf{c}]$  является функцией изображения, а  $F(u, v)$  — преобразованием Фурье этого изображения, то функция  $|F(u, v)|^2$  представляет собой спектр мощности. Модуль здесь означает модуль комплексного числа. Частотную область можно разделить на  $n_r$  кольцевых областей (для частотного содержания) и  $n_d$  клиновидных областей (для ориентационного содержания). Можно вычислить полную энергию каждой области и использовать полученные значения в качестве набора текстурных признаков (см. гл. 5.)

## 7.4 ТЕКСТУРНАЯ СЕГМЕНТАЦИЯ

Любая текстурная характеристика, которая описывает текстуру в окрестности каждого пиксела некоторым значением или вектором значений, может использоваться для сегментации изображения на области с похожей текстурой. Как и другие алгоритмы сегментации, алгоритмы текстурной сегментации разрабатываются на основе двух основных подходов: анализ областей и анализ границ. Алгоритмы на основе областей пытаются найти группы или кластеры пикселей с похожими текстурными свойствами. Алгоритмы на основе границ пытаются обнаружить края текстур между пикселями, которым соответствуют различные текстурные распределения. Алгоритмы сегментации подробно рассматриваются в гл. 10. На рис. 7.8 показаны результаты сегментации нескольких изображений с использованием энергетических текстурных характеристик Лавса. Сегментация выполнялась с помощью алгоритма кластеризации на основе областей.

**Таблица 7.2.** Энергетические текстурные характеристики Лавса для основных областей на рис. 7.8.

Область	E5E5	S5S5	R5R5	E5L5	S5L5	R5L5	S5E5	R5E5	R5S5
Тигр	168.1	84.0	807.7	553.7	354.4	910.6	116.3	339.2	257.4
Вода	68.5	36.9	366.8	218.7	149.3	459.4	49.6	159.1	117.3
Флаги	258.1	113.0	787.7	1057.6	702.2	2056.3	182.4	611.5	350.8
Изгородь	189.5	80.7	624.3	701.7	377.5	803.1	120.6	297.5	215.0
Трава	206.5	103.6	1031.7	625.2	428.3	1153.6	146.0	427.5	323.6
Мелкие цветы	114.9	48.6	289.1	402.6	241.3	484.3	73.6	158.2	109.3
Крупные цветы	76.7	28.8	177.1	301.5	158.4	270.0	45.6	89.7	62.9
Границы	15.3	6.4	64.4	92.3	36.3	74.5	9.3	26.1	19.5

**Таблица 7.3.** Энергетические текстурные характеристики Лавса для областей образа тигра на нескольких различных изображениях.

Изображение	E5E5	S5S5	R5R5	E5L5	S5L5	R5L5	S5E5	R5E5	R5S5
Тигр1	171.2	96.8	1156.8	599.4	378.9	1162.6	124.5	423.8	332.3
Тигр2a	146.3	79.4	801.1	441.8	302.8	996.9	106.5	345.6	256.7
Тигр2b	177.8	96.8	1177.8	531.6	358.1	1080.3	128.2	421.3	334.2
Тигр3	168.8	92.2	966.3	527.2	354.1	1072.3	124.0	389.0	289.8
Тигр4	168.1	84.0	807.7	553.7	354.4	910.6	116.3	339.2	257.4
Тигр5	146.9	80.7	868.7	474.8	326.2	1011.3	108.2	355.5	266.7
Тигр6	170.1	86.8	913.4	551.1	351.3	1180.0	119.5	412.5	295.2
Тигр7	156.3	84.8	954.0	461.8	323.8	1017.7	114.0	372.3	278.6

На рис. 7.8 (а) и (b) изображение тигра было сегментировано на области, соответствующие образу тигра, воды и некоторым другим фрагментам изображения. На рис. 7.8 (c) и (d) изображение со многими объектами было сегментировано на области, примерно соответствующие траве, двум флагам, черной сетчатой изгороди и фону. На рис. 7.8 (e) и (f) изображение поля подсолнухов было сегментировано на области с текстурами трех типов: темные границы в верхней и нижней частях изображения; небольшие, далеко расположенные подсолнухи в глубине поля зрения, и крупные, близко расположенные подсолнухи на переднем плане. В табл. 7.2 приведены средние значения характеристик Лавса для основных областей каждого из этих изображений. Таблица 7.3 позволяет сравнить характеристики Лавса для областей образов тигра на нескольких различных изображениях.

На изображении подсолнухов темные центральные части больших цветков по текстуре объединяются вместе с темными краевыми областями. Это объясняется тем, что маска для вычисления текстурных характеристик значительно

меньше центральных областей крупных подсолнухов. В целом, полученные результаты сегментации несовершенны. Они соответствуют структуре примененных текстурных операторов. Сегментация с учетом цветовых *и* текстурных характеристик могла бы дать лучшие результаты. Но в общем случае, сегментация естественных сцен пока остается нерешенной задачей. Более подробное рассмотрение общих вопросов сегментации содержится в гл. 10.

---

### Упражнение 7.7. Тектурная сегментация

С помощью своей программы для вычисления текстурных характеристик Лавса выясните, можно ли на основе этих характеристик выполнить текстурную сегментацию. Напишите еще одну интерактивную оболочку, которая бы позволяла пользователю указать на изображении прямоугольные области с однотипной текстурой (например, цветы, траву или небо). Для каждой области вычислите средние значения девяти текстурных характеристик. Постройте таблицу, в которой указывается название каждого типа текстуры и приведены соответствующие девять усредненных текстурных характеристик. Сравните эти данные для нескольких различных текстур.

## 7.5 ЛИТЕРАТУРА

Тектурный анализ является одной из старейших областей компьютерного зрения. Первые работы по текстурному анализу были выполнены при решении задач дистанционного зондирования в конце 1960-х — начале 1970-х гг. В статье [Haralick, Shanmugam, Dinstein, 1973] предложено использовать для обработки изображений в задачах дистанционного зондирования классические характерные признаки на основе матрицы вхождений. В работе [Zucker, Terzopoulos, 1980] описан статистический метод для определения наилучшего вектора смещения в методах на основе матрицы вхождений. В статье [Trivedi, 1984] описано применение полутоновых матриц вхождений в задачах обнаружения объектов.

В статье [Julesz, 1975] рассмотрен ряд широко известных в настоящее время экспериментов по человеческому восприятию текстур. В работе [Tamura, Mori, Yamawaki, 1978] разработаны текстурные характерные признаки, значимые для человеческого зрительного восприятия. В статьях [Tomita, Shirai, Tsuji, 1982] и [Tuceryan, Jain, 1990] разработаны структурные методы текстурного анализа. В статье [Wang, He, 1990] предложен спектральный подход, а в [Cross, Jain, 1983] — метод на основе марковских случайных полей. Для получения числовых текстурных признаков на практике часто применяются маски для вычисления энергетических текстурных характеристик Лавса (Laws). В обзорной статье [Weszka, Dyer, Rosenfeld, 1976] описаны первые работы в области текстурного анализа. В современной обзорной статье [Tuceryan, Jain, 1994] рассмотрены основные методы, разработанные вплоть до 1995 г. В недавней работе



[Leung, Malik, 1999] описан новый метод получения текстурных примитивов на основе эталонных образцов.

1. Cross, G. R., and A. K. Jain. 1983. Markov random field texture models. *IEEE Trans. Pattern Anal. Machine Intelligence*, v. PAMI-5:25–39.
2. Haralick, R. M., K. Shanmugam, and I. Dinstein. 1973. Textural features for image classification. *IEEE Trans. Systems, Man, and Cybernetics*, v. 3:610–621.
3. Julesz, B. 1975. Experiments in the visual perception of texture. *Sci. Am.*, 34–43.
4. Laws, K. 1980. Rapid texture identification. *SPIE Image Processing for Missile Guidance*, v. 238:376–380.
5. Leung, T., and J. Malik. 1999. Recognizing surfaces using three-dimensional textons. *Int. Conf. Comput. Vision* (Sept. 1999), 1010–1017.
6. Tamura, H., S. Mori, and T. Yamawaki. 1978. Textural features corresponding to visual perception. *IEEE Trans. Systems, Man, and Cybernetics*, v. 8(6):460–473.
7. Tomita, F., Y. Shirai, and S. Tsuji. 1982. Description of textures by a structural analysis. *IEEE Trans. Pattern Anal. Machine Intelligence*, v. PAMI-4:183–191.
8. Trivedi, M. M. 1984. Object detection based on gray level cooccurrence. *Comput. Vision, Graphics, and Image Proc.*, v. 28:199–219.
9. Tuceryan, M., and A. K. Jain. 1994. Texture analysis. In *Handbook of Pattern Recognition and Vision*, C. H. Chen, L. F. Pau, and P. S. P. Wang, Eds. World Scientific Publishing Co., Singapore, 235–276.
10. Tuceryan, M., and A. K. Jain. 1990. Texture segmentation using Voronoi polygons. *IEEE Trans. Pattern Anal. Machine Intelligence*, v. 12(2):211–216.
11. Wang, L., and D. C. He. 1990. Texture classification using texture spectrum. *Pattern Recog. Lett.*, v. 13:905–910.
12. Weszka, J., C. R. Dyer, and A. Rosenfeld. 1976. A comparative study of texture measures for terrain classification. *IEEE Trans. Systems, Man, and Cybernetics*, v. SMC-6: 269–285.
13. Zucker, S. W., and D. Terzopoulos. 1980. Finding structure in co-occurrence matrices for texture analysis. *Comput. Graphics and Image Proc.*, v. 2:286–308.

# Поиск изображений на основе содержания

Увеличение емкости и уменьшение стоимости устройств оперативной и дисковой памяти, а также постоянный рост производительности процессоров привели к тому, что базы данных изображений перешли из разряда воображаемых компьютерных приложений в разряд реально используемых. Существуют базы данных изображений произведений искусства, спутниковых и медицинских изображений, а также универсальные коллекции фотографий. Способы работы с этими базами данных зависят от приложения. Пользователи художественных коллекций могут захотеть найти работу определенного художника или выяснить, кто является автором заданной картины. Медицинскими базами данных могут пользоваться студенты, изучающие анатомию, или врачи, которым нужны примеры снимков пациентов с заданным заболеванием. К универсальным коллекциям фотографий могут обращаться художники-иллюстраторы в поисках подходящей иллюстрации для статьи или книги. Запросы к базам данных изображений могут быть весьма разнообразными. Один пользователь может захотеть найти изображения лошадей; другому могут потребоваться изображения солнечных закатов, а третьему — изображения, иллюстрирующие абстрактное понятие, например, любовь.

Базы данных изображений могут быть очень большими и содержать сотни тысяч и даже миллионы изображений. В большинстве случаев эти базы данных проиндексированы только по ключевым словам. Эти слова вводятся в базу данных оператором-человеком, который выполняет классификацию изображений по категориям. Однако изображения могут быть найдены в базе данных и по собственному содержанию. Под содержанием изображения можно понимать распределение цветов, текстуру и форму областей изображения, или классификацию объектов. В настоящее время алгоритмы сегментации и распознавания пока находятся на начальном уровне развития. Тем не менее уже разработаны и используются несколько коммерческих и исследовательских систем для поиска изображений в базе данных на основе содержания. Демонстрационные

версии этих систем часто открыты для доступа в Интернет. В данной главе описываются методы, позволяющие людям извлекать изображения из баз данных без применения ключевых слов.

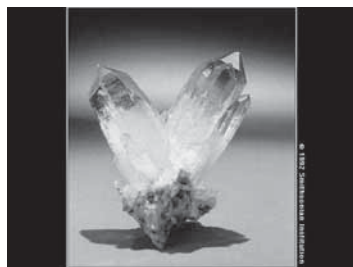
## 8.1 ПРИМЕРЫ БАЗ ДАННЫХ ИЗОБРАЖЕНИЙ

Некоторые базы данных изображений были разработаны с целью демонстрации работоспособности конкретной системы поиска. Подобным примером является система QVIC (Query by Image Content, Запрос по содержанию изображений) компании IBM. QVIC является исследовательской системой, но впоследствии она была использована для разработки коммерческого продукта, распространяемого компанией IBM. Эта система выполняет поиск изображений на основе их видимого содержания. При поиске учитываются такие свойства, как процентное отношение цветов, пространственное распределение цветов и текстура. Компания Virage, Inc. разработала конкурирующий продукт Virage Search Engine (поисковая машина Virage). Эта поисковая машина способна выполнять поиск изображений на основе цвета, композиции, текстуры и структуры. Две перечисленные и другие подобные поисковые машины для поиска изображений могут использоваться для поиска в базах данных, сформированных сторонними организациями. Например, Музей изобразительных искусств Сан-Франциско с помощью системы QVIC организовал доступ к своей базе данных Imagebase, в которой хранится коллекция оцифрованных картин. На рис. 8.1 (а) показана картина Ренуара из этой коллекции. Подобные цифровые коллекции картин были созданы в известных музеях многих городов мира.

Кроме художественных коллекций, существуют универсальные коллекции изображений, в которых отдельные изображения доступны для лицензирова-



(а) Картина Ренуара



(b) Изображение кристалла аметиста

**Рис. 8.1.** Изображения из цифровых коллекций. (Изображение картины Пьера-Огюста Ренуара, Пейзаж в Болье, 1893 г., предоставлено Музеем изобразительных искусств Сан-Франциско. Изображение аметиста предоставлено Смитсоновским институтом, 1992.) См. цветную вклейку.

ния частными покупателями для использования в рекламных целях или для иллюстрирования статей и книг. Одной из крупнейших подобных коллекций является Corbis Archive. В ней хранится более 17 миллионов изображений. Примерно один миллион из них хранится в цифровом виде. Эта коллекция постоянно растет. В ней сделана попытка представить на изображениях полный диапазон человеческих представлений и ощущений. Изображения относятся к таким категориям, как история, искусство, развлечения, наука и технология, животные и др. Corbis обеспечивает поиск изображений по ключевым словам и посредством просмотра. Другая компания, Getty Images, предоставляет в настоящее время доступ через Интернет к нескольким базам данных изображений с классификацией по категориям и с поддержкой поиска по ключевым словам.

В дополнение к базам данных художественных произведений и фотографий, разработаны базы данных медицинских и научных изображений. Национальная медицинская библиотека США подготовила базу данных рентгеновских, томографических и других медицинских изображений. Среди них есть цветные изображения поперечных сечений, снятых с небольшим шагом вдоль мужского и женского человеческого тела. Свыше 14000 изображений доступны в этой базе данных для людей, желающих использовать эти изображения в медицинских исследованиях. Национальный комитет по авиации и исследованию космического пространства США (National Aeronautics and Space Administration, NASA) сформировал большие базы данных изображений со спутников и открыл к ним платный доступ. Геологическая служба США (United States Geological Survey, USGS) предоставила возможность поиска через Интернет для пользователей, которые хотят найти и заказать наборы данных, в том числе цифровые спутниковые и авиационные изображения. Наконец, самый известный сервис Интернет — «Всемирная паутина» (World Wide Web) — является огромной базой данных, которая содержит и текст, и большое количество изображений. В настоящее время разрабатываются поисковые машины для поиска изображений в Интернете на основе ключевых слов и, в ограниченной степени, на основе содержания изображений.

## 8.2 ЗАПРОСЫ К БАЗАМ ДАННЫХ ИЗОБРАЖЕНИЙ

Для работы с базой данных изображений желательно иметь некоторый способ поиска изображений, который был бы более удобен и эффективен, чем непосредственный перебор всей базы данных. Компании, выполняющие подготовку баз данных изображений и открывающие доступ к ним клиентам, обычно выполняют два этапа обработки: на первом выбираются изображения для включения в коллекцию, а на втором этапе производится классификация выбранных изображений посредством назначения им общих категорий и ключевых слов. Изображения, доступные на страницах World Wide Web в

Интернет, обычно снабжены подписями, из которых ключевые слова можно извлечь автоматически.

С помощью реляционных систем управления базами данных сущности можно находить на основе значений их текстовых атрибутов. Для поиска изображений могут применяться такие атрибуты, как общая категория, названия присутствующих на изображении объектов и имена людей, дата создания и название источника. Для ускорения выполнения поисковых запросов изображения могут быть проиндексированы по этим атрибутам. Запросы по текстовым атрибутам можно записывать на языке запросов SQL, который в реляционных базах данных является стандартным языком запросов. Например, запрос

```
SELECT * FROM IMAGEDB
WHERE CATEGORY = 'ДРАГОЦЕННЫЕ КАМНИ'
      AND SOURCE = 'СМИТСОНИАНСКИЙ ИНСТИТУТ'
```

мог бы найти и вернуть все изображения из набора с именем IMAGEDB, у которых атрибут CATEGORY имеет значение 'ДРАГОЦЕННЫЕ КАМНИ', а атрибут SOURCE имеет значение 'СМИТСОНИАНСКИЙ ИНСТИТУТ'. Цель данного запроса могла бы заключаться в получении изображений коллекции драгоценных камней, хранящейся в Смитсоновском институте. На рис. 8.1 (b) приведено изображение аметиста из этой коллекции. Оно могло бы быть найдено наряду с изображениями других драгоценных камней. Для обеспечения более избирательных выборок можно было бы предусмотреть хранение для каждого изображения описывающего множества ключевых слов. Например, в таблице реляционной базы данных можно было бы предусмотреть атрибут KEYWORD для хранения нескольких значений для каждого изображения. Так, в качестве ключевых слов изображения аметиста могли бы быть выбраны слова 'АМЕТИСТ' и 'КРИСТАЛЛ'. Тогда это изображение можно найти с помощью запроса, в котором указано одно или оба значения ключевых слов. Например, SQL-запрос

```
SELECT * FROM IMAGEDB
WHERE CATEGORY = 'ДРАГОЦЕННЫЕ КАМНИ'
      AND SOURCE = 'СМИТСОНИАНСКИЙ ИНСТИТУТ'
      AND (KEYWORD = 'АМЕТИСТ' OR KEYWORD = 'КРИСТАЛЛ')
```

найдет все изображения в наборе с именем IMAGEDB, у которых атрибут CATEGORY имеет значение 'ДРАГОЦЕННЫЕ КАМНИ', атрибут SOURCE имеет значение 'СМИТСОНИАНСКИЙ ИНСТИТУТ', а значение атрибута KEYWORD равно 'АМЕТИСТ' или 'КРИСТАЛЛ'. Вероятно, этот запрос найдет некоторое множество изображений, а не одно изображение аметиста. Пользователь сможет просмотреть найденные изображения и выбрать среди них те, которые действительно ему нужны.

Поиск на основе ключевых слов имеет ряд ограничений. Назначение ключевых слов человеком является трудоемкой операцией и допускает неоднозначное выполнение. Разные люди могут использовать для описания изображения



(a) Изображение поросят



(b) Изображение, на котором поросят нет

**Рис. 8.2.** Изображения с поросятами, найденные в результате поиска по ключевым словам. (Изображение (a) лицензировано из архива Corbis. Credit line: \051 Clive Druett; Papilio/CORBIS.)

различные ключевые слова. Для назначения ключевых слов изображениям из Интернете может применяться автоматический анализ подписей изображений на HTML-страницах. Но возможности индексации при этом весьма ограничены. Более того, при автоматическом назначении ключевых слов некоторые из найденных изображений могут весьма сильно отличаться от ожиданий пользователя. На рис. 8.2 показаны два изображения, найденные по ключевому слову «поросята» одной из поисковых машин для World Wide Web.

Полагая, что использование ключевых слов обеспечивает недостаточно эффективный поиск изображений, далее рассмотрим ряд других методов поиска изображений. Они могут применяться вместо или в дополнение к поиску по ключевым словам.

---

### Упражнение 8.1. Запросы по ключевым словам

Запишите на языке SQL запрос, который мог бы найти изображение, показанное на рис. 8.2 (a), но при этом не извлекал бы изображение, показанное на рис. 8.2 (b). Используйте любые подходящие категории и ключевые слова.

## 8.3 ЗАПРОСЫ ПО ОБРАЗЦУ

В терминологии баз данных *запросом по образцу* (query-by-example, QBE) называется запрос, который формируется посредством заполнения значений и ограничений в некоторой табличной форме, которую затем можно конвертировать в формат языка SQL. Первая система для построения запросов по образцу была разработана компанией IBM. В настоящее время хорошим примером системы подобного типа может служить Microsoft Access. В стандартных реляционных базах данных, в которых атрибуты имеют преимущественно текстовый или числовой тип, механизм запросов по образцу фактически сводится к предоставлению пользователям удобного интерфейса для заполнения формы с параметрами запроса и не содержит каких-либо дополнительных средств.

Применение запросов по образцу в базах данных изображений открывает более широкие возможности. Вместо печати значений атрибутов запроса, пользователь базы данных изображений мог бы предъявить системе образец изображения, или в интерактивном режиме нарисовать его на экране, или схематично изобразить контур искомого объекта (эскиз). Затем система поиска должна найти похожие изображения или изображения, содержащие требуемые объекты. В этом заключается цель всех систем поиска изображений на основе содержания. В каждой из них предусмотрены собственные средства для задания запросов, для определения сходства между запросом и конкретным изображением из базы данных, и для выбора возвращаемых в результате поиска изображений.

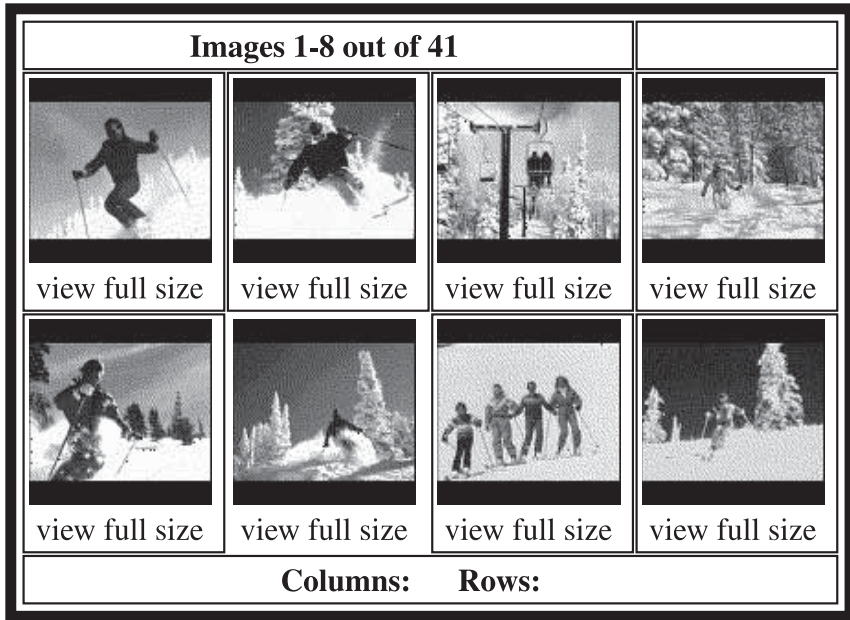
Для сохранения общего характера обсуждения будем считать, что запрос состоит из примера изображения и дополнительного набора ограничений. Изображение может быть цифровой фотографией, нарисованным пользователем грубым примером, или контурным изображением. Также может быть задано пустое изображение, тогда найденное множество изображений должно удовлетворять только набору ограничений, указанных в запросе. Ограничения могут быть заданы в виде ключевых слов, существующих в некоторой системе индексации, или могут задавать объекты, которые должны быть на изображении, или даже описывать пространственные отношения между этими объектами. В наиболее общем случае, запрос содержит цифровое изображение, которое сравнивается с изображениями базы данных согласно применяемой *мере расстояния между изображениями*. Если расстояние равно 0, то считается, что изображение точно соответствует запросу. Значения, большие 0, соответствуют различной степени сходства изображения с запросом. Поисковые машины для поиска изображений обычно возвращают набор изображений, отсортированный по значению расстояния от изображения, указанного в запросе. На рис. 8.3 показаны результаты поиска в системе QVIC с применением меры расстояния на основе цветового макета. Приведены 8 изображений, которые были выбраны системой в качестве наиболее похожих на изображение запроса. Это изображение показано в верхнем левом углу рис. 8.3, так как оно тоже хранится в базе данных и в результате поиска оказывается максимально похожим на изображение запроса, т. е. на само себя.

#### 8.4 МЕРЫ РАССТОЯНИЯ МЕЖДУ ИЗОБРАЖЕНИЯМИ

Для определения сходства изображения из базы данных с изображением, указанным в запросе, обычно применяется некоторая мера расстояния или характеристики, с помощью которых можно получить численную оценку сходства изображений. Характеристики сходства изображений можно разделить на четыре основные группы:

1. цветовое сходство;
2. текстурное сходство;





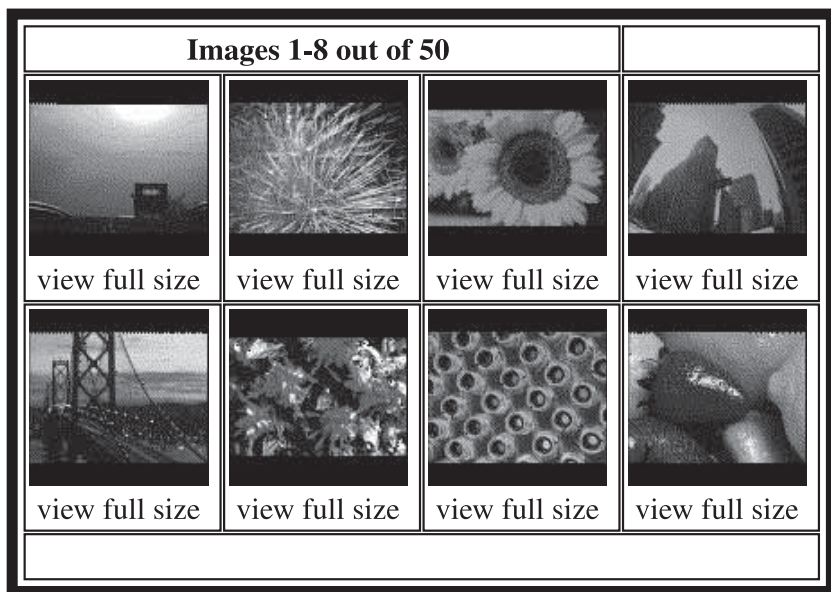
**Рис. 8.3.** Результаты поиска изображений в системе QVIC с использованием меры расстояния на основе цветового макета. В запросе было указано изображение, показанное в верхнем левом углу. (С любезного разрешения Egames.) См. цветную вклейку.

3. сходство формы;
4. сходство объектов и отношений между объектами.

### 8.4.1 Характеристики цветового сходства

Характеристики цветового сходства часто выбираются очень простыми. Они позволяют сравнить цветовое содержание одного изображения с цветовым содержанием другого изображения или с параметрами, заданными в запросе. Например, в системе QVIC пользователь может указать в запросе процентное содержание цветов в искомых изображениях. Пользователь может выбрать до 5 цветов из таблицы цветов и задать процентное содержание каждого цвета в искомых изображениях. Затем система QVIC выполняет поиск изображений, количественное распределение цветов в которых близко к параметрам запроса. При этом не учитывается пространственное распределение пикселей различных цветов на изображении. На рис. 8.4 показан набор изображений, полученных в результате выполнения запроса с указанием 40% красного, 30% желтого и 10% черного цвета. Хотя найденные изображения содержат очень похожие цвета, но смысловое содержание этих изображений существенно отличается.





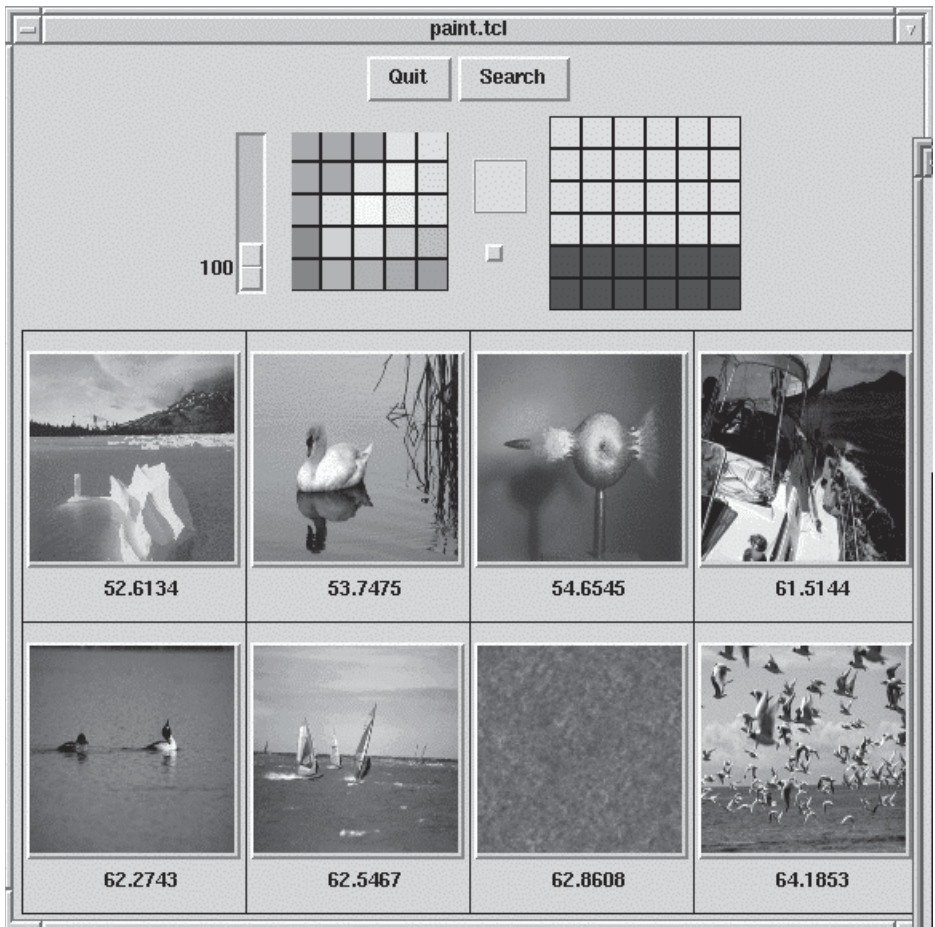
**Рис. 8.4.** Результаты поиска, выполненного системой QBIC на основе процентного содержания цветов. В запросе был задан поиск изображений, содержащих 40% красного, 30% желтого и 10% черного цвета. (С любезного разрешения Egames.) См. цветную вклейку.

Похожий способ поиска основан на сопоставлении цветовых гистограмм. Этот метод был описан в гл. 6 и используется в системе для распознавания овощей и фруктов в гл. 16. Пользователь системы поиска указывает изображение-образец и выбирает команду для поиска изображений, которые близки к изображению-образцу по расстоянию на основе цветовой гистограммы. Меры расстояния на основе цветовой гистограммы должны предусматривать оценку сходства двух различных цветов. Например, в системе QBIC расстояние на основе цветовой гистограммы определено следующим образом:

$$d_{hist}(I, Q) = (h(I) - h(Q))^T A (h(I) - h(Q)) \quad (8.1)$$

где  $h(I)$  и  $h(Q)$  являются  $K$ -разрядными гистограммами изображений  $I$  и  $Q$ , соответственно, а  $A$  — это матрица сходства размерами  $K \times K$ . В матрице сходства элементы, значения которых близки к 1, соответствуют похожим цветам, а элементы, близкие или равные 0, соответствуют сильно отличающимся цветам.

Еще одна возможная мера расстояния основана на использовании цветового макета (color layout). При формировании запроса пользователю обычно предьявляется пустая сетка. Для каждой клетки сетки пользователь может выбрать цвет из таблицы. При формировании запроса в окне на рис. 8.5 пользователь выбрал из таблицы цветов (она находится в левой верхней части окна) два



**Рис. 8.5.** Результаты поиска в базе данных изображений по запросу на основе закрашенной сетки. (Изображения получены из базы данных MIT Media Lab VisTex: <http://vismod.www.media.mit.edu/vismod/imagery/VisionTexture/vistex.html>.) См. цветную вклейку.

цвета и закрасил этими цветами сетку макета  $6 \times 6$ , которая расположена в правой верхней части окна. В результате поиска были найдены изображения, показанные в окне в уменьшенном виде. Система выбрала эти изображения в качестве наиболее хорошо удовлетворяющих запросу на основе цветового макета. Как было показано на рис. 8.3, при поиске на основе цветового макета вместо закрашенной сетки в запросе можно указать изображение-образец. Тогда система выполняет поиск изображений, у которых пространственное распределение цветов близко к распределению цветов в изображении-образце.

Характеристики сходства на основе цветового макета, в которых используется закрашенная сетка, требуют определения меры расстояния между изображениями, которая учитывала бы содержание двух закрашенных сеток. Эта мера  $\hat{d}_{color}$  должна обеспечивать сравнение каждой клетки сетки, указанной в запросе, с соответствующей клеткой сетки произвольного изображения из базы данных. Результаты сравнения всех клеток комбинируются для получения значения расстояния между изображениями:

$$d_{gridded\_color}(I, Q) = \sum_g \hat{d}_{color}(C^I(g), C^Q(g)) \quad (8.2)$$

где  $C^I(g)$  — это цвет клетки  $g$  сетки изображения  $I$  из базы данных, а  $C^Q(g)$  — цвет соответствующей клетки  $g$  сетки изображения  $Q$ , указанного в запросе. Для представления цветов клеток сетки могут использоваться различные способы. Перечислим несколько подходящих способов представления цвета:

1. среднее значение цвета в пределах клетки сетки,
2. среднее значение и среднеквадратическое отклонение цвета, и
3. многоуровневая цветовая гистограмма.

Мера расстояния между клетками сеток  $\hat{d}$  должна учитывать выбранный способ представления цвета и обеспечивать получение поддающихся интерпретации значений расстояния. Например, если среднее значение цвета представлено в виде тройки чисел  $(R, G, B)$ , то очевидным (но не обязательно наилучшим) вариантом меры могла бы быть мера вида  $\hat{d} = \|(R^Q, G^Q, B^Q) - (R^I, G^I, B^I)\|^2$ . Вместо сравнения значений  $(R, G, B)$  некоторые системы разбивают пространство цветов на множество трехмерных областей и сохраняют в таблице численные значения сходства между парами областей. Аналогичный подход использован при определении меры расстояния на основе гистограмм в системе QVIC (см. уравнение (8.1)).

---

### Упражнение 8.2. Меры расстояния на основе цветовых гистограмм

Разработайте программу сравнения изображений с использованием меры расстояния на основе цветовых гистограмм размера  $4 \times 4 \times 4$ . Входными данными для сравнения являются два изображения. Выполните сравнение изображений целиком или их избранных фрагментов. Используйте эту меру расстояния для реализации меры расстояния на основе раскрашенной сетки. Предусмотрите указание размеров сетки пользователем. В мере расстояния должны комбинироваться расстояния между всеми парами соответствующих клеток сеток сравниваемых изображений и в результате должно вычисляться одно значение расстояния (см. уравнение (8.2)). Проверьте работу программы сравнения изображений с использованием меры расстояния на основе закрашенной сетки на нескольких парах цветных изображений и с сетками размерами  $1 \times 1$ ,  $4 \times 4$  и  $8 \times 8$ .

### 8.4.2 Характеристики текстурного сходства

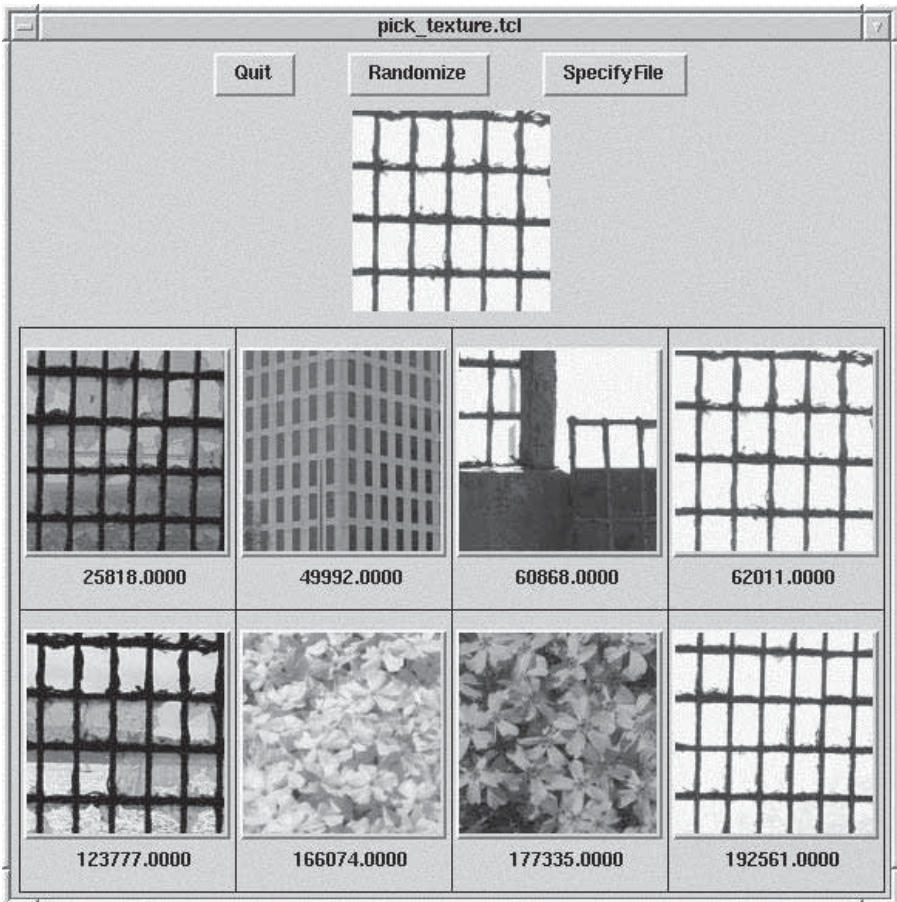
Характеристики текстурного сходства несколько сложнее характеристик цветового сходства. Изображения, близкие по характеристике текстурного сходства, должны иметь одинаковое пространственное распределение цветов (или значений яркости), но при этом значения цветов (или яркости) двух изображений могут не совпадать. Для оценки сходства двух текстур можно использовать текстурные характеристики, рассматривавшиеся в гл. 7. На рис. 8.6 показаны результаты поиска изображений по текстурному сходству с использованием меры расстояния на основе энергетических текстурных характеристик Лавса. Как видно из результатов запроса, это расстояние не зависит от цветов пикселей изображений. Однако можно построить меры расстояния, совместно учитывающие и текстурное, и цветовое сходство.

При рассмотрении характеристик текстурного сходства необходимо уточнить два аспекта:

1. представление текстуры и
2. определение сходства с учетом выбранного представления текстуры.

Один из наиболее распространенных способов представления текстуры основан на использовании *вектора текстурного описания* (texture description vector). Это вектор с числовыми компонентами, которые соответствуют обобщенным параметрам текстуры заданного изображения или некоторой области изображения. Примерами векторов текстурного описания являются пятикомпонентный вектор Харалика, в котором хранятся признаки, извлеченные из матрицы вхождений, и вектор с девятью компонентами, соответствующими энергетическим текстурным характеристикам Лавса. Вектор текстурного описания может использоваться применительно к текстуре целого изображения, но в таком случае хорошие результаты достигаются только при описании изображений с однородной текстурой. Для изображений более общего вида, векторы текстурного описания обычно вычисляются для каждого пиксела по небольшой (например,  $15 \times 15$ ) окрестности этого пиксела. Затем пикселы группируются с помощью алгоритма кластеризации, который назначает уникальную метку каждой новой обнаруженной текстурной категории.

Пикселам можно назначить вектор текстурного описания и затем произвести маркировку пикселов по их принадлежности текстурным категориям. Это позволяет определить различные текстурные меры расстояния. Простейшим вариантом подобной меры является расстояние *от выбранного образца* (pick-and-click distance). При этом пользователь выбирает текстуру щелчком мыши на произвольном пикселе внутри текстурной области изображения из запроса или выбирает образец текстуры из некоторого доступного набора. Выбранный образец текстуры представляется в виде вектора текстурного описания, который в процессе поиска сравнивается с векторами текстурного описания, соответствующими изображениям в базе данных. Мера расстояния



**Рис. 8.6.** Результаты поиска в базе данных изображений с использованием меры расстояния на основе текстурного сходства. (Изображения получены из базы данных MIT Media Lab VisTex: <http://vismod.www.media.mit.edu/vismod/imagery/VisionTexture/vistex.html>.) См. цветную вклейку.

определяется следующим выражением:

$$d_{pick\_and\_click}(I, Q) = \min_{i \in I} \|T(i) - T(Q)\|^2 \quad (8.3)$$

где  $\mathbf{T}(i)$  — вектор текстурного описания пиксела  $i$  изображения  $I$ , а  $\mathbf{T}(Q)$  — вектор текстурного описания для выбранного пользователем текстурного образца, который используется в запросе. Может показаться, что этот метод требует большого объема вычислений, но на практике этот объем можно существенно уменьшить за счет сохранения в базе данных изображений списка текстурных категорий каждого изображения. Тектурные категории определяются с помощью процедуры кластеризации. В процессе поиска для каждого



изображения из базы данных требуется выполнить сравнение вектора текстурного описания из запроса с векторами текстурного описания, хранящимися в списке для этого изображения. Для дополнительного ускорения процесса поиска может применяться индексация.

Использование меры расстояния от выбранного образца требует интерактивного указания текстуры пользователем. Эту меру расстояния затруднительно автоматически использовать с изображением, указанным в запросе. Более удобную характеристику текстурного сходства можно сформировать посредством обобщения на случай текстур характеристик на основе закрашенных сеток, которые обсуждались выше применительно к характеристикам цветового сходства. На изображение, указанное в запросе, накладывается сетка и для каждой клетки сетки вычисляется вектор текстурного описания. Аналогичная процедура выполняется для изображений в базе данных. Расстояние на основе сеточного текстурного сходства (*gridded texture distance*) вычисляется по формуле:

$$d_{gridded\_texture}(I, Q) = \sum_g \hat{d}_{texture}(T^I(g), T^Q(g)) \quad (8.4)$$

где  $\hat{d}_{texture}$  может быть евклидовым расстоянием или какой-либо другой мерой расстояния между клетками сеток.

Аналогично расстоянию между цветовыми гистограммами можно определить расстояние между текстурными гистограммами. Для каждой текстурной категории в гистограмме представлено количество пикселей, у которых вектор текстурного описания принадлежит данной категории. Интересная и легко вычисляемая характеристика на основе текстурной гистограммы была разработана в расчете на использование пар соединяющихся отрезков. Сначала к изображению для обнаружения контурных отрезков применяется определитель прямых (см. гл. 10). Затем выполняется поиск соединяющихся или «почти» соединяющихся пар отрезков и вычисляются углы между отрезками каждой такой пары. Значения углов используются для построения текстурной гистограммы, которая применяется в качестве описания текстуры изображения.

### 8.4.3 Характеристики сходства формы

Цвет и текстура могут служить глобальными атрибутами изображения. Меры расстояния на основе цвета и текстуры предназначены для определения, содержит ли некоторое изображение заданный цвет или текстуру и расположены ли области этого цвета или текстуры в тех же местах, что и на изображении, указанном в запросе. Форма не является атрибутом изображения; не имеет смысла спрашивать, какую форму имеет изображение. Понятие формы применимо к некоторой области изображения. При использовании характеристик формы необходимы дополнительные (по сравнению с характеристиками цвета и текстуры) операции обработки, так как перед вычислением характеристик

сходства формы требуется выполнить идентификацию областей. Часто это делается вручную, но в некоторых задачах удается применять автоматическую сегментацию. Сегментация представляет собой важную и еще не до конца решенную задачу, решение которой необходимо для широкого распространения методов поиска на основе формы. Сегментация будет рассмотрена в гл. 10. Методы сопоставления описаний формы описываются ниже.

---

### Упражнение 8.3. Меры расстояния на основе текстурного сходства

Выберите несколько различных текстурных характеристик из гл. 7 и реализуйте на их основе меры расстояния между изображениями. Предусмотрите сравнение текстуры фрагмента изображения из запроса с текстурой фрагмента изображения из базы данных. Затем разработайте программу с реализацией меры расстояния на основе сеточного текстурного сходства, которая бы позволяла использовать для сравнения клеток сеток различные текстурные характеристики. Оцените результаты сравнения на одном и том же наборе изображений с использованием различных текстурных характеристик и сеток разных размеров. В тестовый набор изображений включите изображения, содержащие области с различными текстурами.

Методы распознавания объектов по форме их двумерных образов играют важную роль в анализе изображений. В гл. 3 был определен ряд свойств областей изображений. Эти свойства можно считать *глобальными характеристиками формы* (global shape properties), так как они относятся к форме в целом. Сравнение формы двух областей может быть выполнено с учетом глобальных свойств формы с помощью различных методов статистического распознавания образов, которые рассматривались в гл. 4. При сопоставлении формы областей могут также использоваться структурные методы, применительно к которым форма описывается посредством перечисления примитивных элементов и пространственных отношений между ними. Так как структурное описание формы представляет собой реляционный граф, то для сопоставления описаний формы можно применять методы сопоставления графов. Сопоставление формы областей методами сопоставления графов является весьма мощным подходом, так как описание обычно включает в себя пространственные отношения, инвариантные относительно большинства двумерных преобразований. Однако сопоставление графов может выполняться очень медленно, так как временные затраты экспоненциально растут с увеличением количества элементов. В системах поиска изображений на основе содержания желательно применять методы, позволяющие быстро определить, насколько форма области изображения близка к описанию формы, указанной в запросе. Часто требуется, чтобы методы сопоставления формы были инвариантны относительно переноса и размера. Иногда также требуется инвариантность относительно поворота, чтобы область заданной формы могла быть обнаружена независимо от ориентации на изображении. Однако при поиске изображений в базах данных инвариантность

относительно поворота требуется не всегда. Существует много изображений, на которых объекты обычно встречаются в одной и той же ориентации. Примерами подобных объектов являются деревья, здания и грузовые автомобили на изображениях природных и городских сцен.

В литературе по компьютерному зрению описано много различных характеристик формы. Среди них есть грубые глобальные характеристики, которые упрощают, но не обеспечивают, распознавание объектов, а также очень детализированные характеристики, позволяющие обнаружить объекты весьма специфической формы. Примером простой характеристики могут служить гистограммы формы. Они позволяют исключать из обработки области, которые нельзя успешно сопоставить с описанием формы из запроса. Но при обработке с помощью гистограмм формы допускается много ложных срабатываний (как и при использовании цветowych гистограмм). Более избирательными являются методы сопоставления границ. Они работают с представлением формы в виде границы области. Эти методы позволяют найти области, форма которых описывается похожими границами. Эскизное сопоставление может оказаться еще более точным, так как при этом рассматривается не только граница одного объекта, а множество контуров изображения, охватывающих один или более объектов. Это множество контуров (эскиз) сопоставляется с контурным изображением запроса, которое было нарисовано или выбрано пользователем. Перечисленные характеристики формы далее будут рассмотрены подробнее.

**Гистограммы формы.** Выше было описано применение гистограмм для сопоставления изображений по цветовому и текстурному сходству. Применение гистограмм привлекательно тем, что расстояние между ними просто определяется и его можно быстро вычислять. Поэтому было бы естественно попробовать применить гистограммы для сопоставления описаний формы. Основная проблема заключается в определении переменной, по которой будет строиться гистограмма.

Рассмотрим связную область произвольной формы на бинарном изображении. Допустим, что пиксели области имеют значения 1, а все остальные пиксели изображения равны 0. Один из способов сопоставления гистограмм заключается в *проекционном сопоставлении* (projection matching) с использованием горизонтальных и вертикальных проекций области на строки и столбцы изображения. Предположим, что область занимает  $n$  строк и  $m$  столбцов. Построим гистограмму, поставив в соответствие каждой строке и каждому столбцу один разряд гистограммы. Значение разряда гистограммы равно количеству единичных пикселей, расположенных в соответствующей строке или столбце области. Таким образом можно построить гистограмму с  $n + m$  разрядами, которая пригодна для сопоставления областей одинаковых размеров.

Чтобы проекционное сопоставление не зависело от масштаба, можно зафиксировать количество разрядов, соответствующих строкам и столбцам. Если определить порядок разрядов так, чтобы они соответствовали просмотру



области в направлении от левого верхнего угла к правому нижнему, то можно обеспечить инвариантность относительно переноса. Проекционное сопоставление неинвариантно относительно поворота, но обеспечивает сравнение изображений при малых углах поворота и других незначительных геометрических искажениях.

Один из способов обеспечения инвариантности относительно поворота предполагает вычисление осей наилучшего аппроксимирующего эллипса (см. гл. 3) и поворот области до тех пор, пока большая ось ее аппроксимирующего эллипса не станет вертикальной (в пиксельной системе координат). Так как положение «верхней» части области заранее не известно, то возможны два угла поворота. Если большая и малая оси аппроксимирующего эллипса примерно равны, то потребуются рассматривать не два, а четыре возможных угла поворота. Проекционное сопоставление было успешно применено для решения задачи поиска изображений эмблем.

Еще один возможный способ описания формы заключается в построении гистограммы по значению угла наклона касательной в каждом граничном пикселе области. Эта мера по определению является инвариантной относительно масштаба и переноса, но не инвариантна относительно поворота, так как угол наклона касательной вычисляется для области с некоторой фиксированной ориентацией. Известны несколько путей решения этой проблемы. Во-первых, область можно повернуть, контролируя положение большой оси аппроксимирующего эллипса (как было описано выше). Еще один простой способ вместо поворота области предполагает циклический сдвиг гистограммы. Для  $K$ -разрядной гистограммы возможны  $K$  различных вариантов сдвига. Некорректные значения сдвига можно быстро обнаружить по сильному увеличению расстояния между гистограммами сравниваемых описаний формы. Вместо перебора всех возможных вариантов сдвига гистограммы, можно выполнить нормировку гистограмм, выбирая в качестве первого разряда разряд с наибольшим значением. При наличии шума и искажений можно пытаться производить сравнение не для одного, а для нескольких разрядов с наибольшими значениями.

---

#### Упражнение 8.4. Гистограммы формы

Напишите программу с реализацией меры расстояния на основе гистограмм формы, в которой используется угол наклона касательной в граничных пикселях области. Обеспечьте инвариантность относительно поворота за счет перебора всех возможных вариантов сдвига гистограммы, указанной в запросе. В качестве расстояния между сравниваемыми описаниями формы выберите минимальное значение среди расстояний, соответствующих всем возможным вариантам сдвига гистограммы. Используйте эту меру расстояния для сравнения описаний формы областей, которые получите из тестовых изображений с помощью пороговой бинаризации или в интерактивном режиме.

**Сопоставление границ.** Алгоритмы сопоставления границ требуют выполнения процедур выделения и формирования описания границы области, указанной в запросе, и области изображения из базы данных. Границу области можно представить в виде последовательности символов или в виде многоугольника методом полигональной аппроксимации.

Один из классических методов сравнения описаний формы в виде последовательностей символов основан на использовании рядов Фурье. В математике известны методы разложения в ряды Фурье, применимые к непрерывным и дискретным функциям. Если границу области можно описать с помощью непрерывной функции, то эту функцию можно разложить в ряд Фурье и использовать коэффициенты членов ряда в качестве описания формы области. В дискретном случае границу области можно представить в виде последовательности  $m$  точек  $\langle V_0, V_1, \dots, V_{m-1} \rangle$ . Для этой последовательности точек сформируем последовательность единичных векторов:

$$v_k = \frac{V_{k+1} - V_k}{|V_{k+1} - V_k|} \quad (8.5)$$

и последовательность кумулятивных разностей:

$$l_k = \sum_{i=1}^k |V_i - V_{i-1}|, \quad k > 0$$

$$l_0 = 0 \quad (8.6)$$

Коэффициенты Фурье  $\{a_{-M}, \dots, a_0, \dots, a_M\}$  можно приближенно представить с помощью кумулятивных разностей:

$$a_n = \frac{1}{L \left(\frac{n2\pi}{L}\right)^2} \sum_{k=1}^m (v_{k-1} - v_k) e^{-jn(2\pi/L)l_k} \quad (8.7)$$

Эти коэффициенты можно использовать для определения меры расстояния на основе сходства формы. Обозначим символом  $\mathbf{Q}$  описание формы, указанное в запросе, а символом  $\mathbf{I}$  — описание формы области изображения, которое надо сравнить с описанием  $\mathbf{Q}$ . Пусть  $\{a_n^{\mathbf{Q}}\}$  является последовательностью коэффициентов Фурье для описания формы в запросе, а  $\{a_n^{\mathbf{I}}\}$  — последовательностью коэффициентов Фурье для описания формы области изображения. Тогда меру расстояния на основе коэффициентов Фурье можно определить с помощью выражения

$$d_{\text{Fourier}}(\mathbf{I}, \mathbf{Q}) = \left[ \sum_{n=-M}^M |a_n^{\mathbf{I}} - a_n^{\mathbf{Q}}|^2 \right]^{\frac{1}{2}} \quad (8.8)$$

Эта мера расстояния инвариантна только относительно переноса. Если требуется инвариантность относительно других преобразований, то эту меру расстояния можно применять совместно с численной процедурой, определяющей параметры масштабирования, поворота и начальную точку, минимизирующую значение  $d_{\text{Fourier}}(\mathbf{I}, \mathbf{Q})$ .

Если граница области представлена в виде многоугольника (методом полигональной аппроксимации), то для описания формы можно вычислить и использовать длины сторон многоугольника и углы между ними. Форму области можно представить в виде последовательности точек стыка  $\langle X_i, Y_i, \alpha_i \rangle$ . Каждая тройка чисел  $\langle X_i, Y_i, \alpha_i \rangle$  обозначает пару отрезков, соединяющихся в точке с координатами  $(X_i, Y_i)$  под углом  $\alpha_i$ .

Допустим, что задана последовательность  $Q = Q_1, Q_2, \dots, Q_n$  точек стыка, представляющая границу области запроса  $Q$  и похожая последовательность  $I = I_1, I_2, \dots, I_m$ , представляющая границу области изображения  $I$ . Тогда для сопоставления границ необходимо найти такое отображение  $Q$  на  $I$ , которое отображает отрезки прямых из описания формы запроса на отрезки похожей длины в описании формы области изображения. Это отображение должно отображать пару смежных отрезков из описания запроса, соединяющихся под углом  $\alpha$ , на пару смежных отрезков в описании области изображения, которые тоже соединяются под близким по значению углом  $\alpha'$ .

Известен еще один метод сопоставления границ, называемый *эластичным сопоставлением* (elastic matching). При использовании этого метода область из запроса деформируется так, чтобы по форме она стала максимально похожей на область изображения. Расстояние между описанием формы из запроса и формы области изображения зависит от двух компонент: (1) энергии, необходимой для деформации области запроса до наилучшего совпадения с областью изображения, и (2) численной характеристики качества совпадения деформированной области запроса с областью изображения. На рис. 8.7 показаны результаты поиска изображений лошадей с использованием эластичного сопоставления. В запросе был указан грубо нарисованный пользователем контур искомой области.

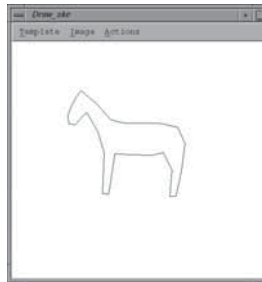
---

### Упражнение 8.5. Сопоставление границ

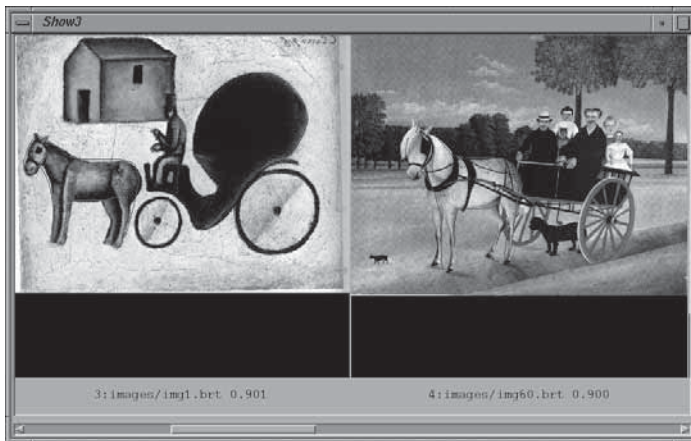
Для сопоставления граничных описаний формы областей известно много алгоритмов, но в настоящее время эти алгоритмы не получили широкого распространения в системах поиска изображений по содержанию. Как вы полагаете, чем это объясняется?

**Эскизное сопоставление** Системы эскизного сопоставления позволяют оператору нарисовать эскизное изображение, содержащее основные контуры изображения. Затем это изображение используется для поиска цветных или полутоновых изображений, содержащих похожие контуры.

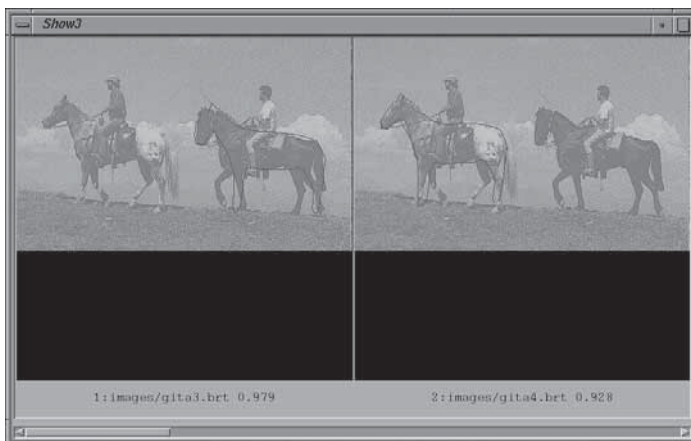
В базе данных системы ART MUSEUM хранятся цветные изображения знаменитых картин. Эти цветные изображения подвергаются предварительной обработке для получения промежуточного представления, называемого *абстрактным изображением* (abstract image). Ниже перечислены операции, выполняемые на этапе предварительной обработки.



(a) Описание формы в запросе строится на основе контура, нарисованного пользователем



(b) Два изображения из найденного набора



(c) Еще одно найденное изображение, на котором были обнаружены две лошади

**Рис. 8.7.** Поиск изображений с использованием эластичного сопоставления. (С любезного разрешения Alberto Del Vimbo.) См. цветную вклейку.

1. Уменьшение изображения до заданных размеров (например,  $64 \times 64$  пиксела) с помощью аффинного преобразования. Для удаления шума применяется медианный фильтр. Результат обработки называется нормированным изображением.
2. Обнаружение перепадов с помощью градиентного алгоритма выделения перепадов. В процессе обнаружения перепадов выполняются два основных действия: сначала производится поиск глобальных перепадов по глобальному порогу, который выбирается по среднему значению и дисперсии градиента; затем среди глобальных перепадов в соответствии с локально вычисленными порогами выбираются локальные перепады. В результате обработки формируется очищенное контурное изображение (refined edge image).
3. На очищенном контурном изображении выполняется скелетизация и удаление избыточных контуров. Этот окончательный результат называется абстрактным изображением. Оно представляет собой относительно чистый от помех эскиз, содержащий основные контуры исходного изображения.

После того как пользователь указывает в запросе грубый эскиз, этот эскиз подвергается обработке, аналогичной предварительной обработке изображений из базы данных. Для эскиза тоже выполняется нормировка размеров, бинаризация, скелетизация и удаление избыточных контуров. В результате формируется изображение, называемое линейным эскизом (linear sketch).

Линейный эскиз пригоден для сопоставления с абстрактными изображениями. Алгоритм сопоставления имеет корреляционный характер. Два изображения делятся на клетки посредством наложения сеток. Для каждой клетки сетки изображения запроса вычисляется локальная корреляция с соответствующей клеткой сетки изображения из базы данных. С целью повышения надежности локальная корреляция вычисляется несколько раз для нескольких значений сдвига положения клетки сетки на изображении из базы данных. Среди этих сдвигов выбирается максимальное значение корреляции. Оно используется в качестве результата сопоставления двух клеток изображения из базы данных и изображения из запроса. Характеристика сходства изображений имеет значение, равное сумме этих максимальных локальных корреляций. Мера расстояния обратно пропорциональна значению этой характеристики сходства. С применением использовавшихся выше обозначений, меру расстояния можно определить следующим образом:

$$d_{sketch}(I, Q) = \frac{1}{\sum_g \max_n [\hat{d}_{correlation}(shift_n(A^I(g)), L^Q(g))]} \quad (8.9)$$

где  $A^I(g)$  обозначает клетку  $g$  сетки абстрактного изображения, вычисленного для изображения из базы данных  $I$ ,  $shift(A^I(g))$  обозначает ту же самую клетку  $g$  после сдвига на том же самом абстрактном изображении, а  $L^Q(g)$  обозначает клетку  $g$  сетки на линейном эскизе для изображения  $Q$ , указанного в запросе.

---

**Упражнение 8.6.** Эскизное сопоставление

Спроектируйте и программно реализуйте меру расстояния на основе эскизного сопоставления, аналогичную мере расстояния системы ART MUSEUM. Используйте эту меру для поиска в некотором наборе известных изображений согласно эскизу, нарисованному пользователем.

**8.4.4 Характеристики сходства, учитывающие присутствующие на изображении объекты и их взаимное расположение**

Большинство мер расстояния, применявшихся в первых поисковых машинах для поиска изображений, учитывали цвет, текстуру и форму областей. Но в целом эти характеристики нельзя назвать естественными для формирования запросов пользователем. Многие пользователи поисковых машин хотели бы выполнять поиск изображений, указывая, что именно на них содержится. Возможно, это определенные объекты, например, люди или собаки. Или это могут быть абстрактные понятия, такие, как счастье или бедность. Первые системы, в которых было предусмотрено распознавание объектов, позволяли искать изображения, содержащие объекты нескольких классов, например, лошадей или человеческие лица и тела. Для прогресса в этой области необходимы дальнейшие исследования по разработке методов распознавания объектов, пригодных для использования при поиске изображений.

**Обнаружение образов человеческих лиц.** Задача обнаружения лиц имеет большое значение, так как позволяет организовать поиск изображений, содержащих образы людей. Обнаружение лиц является сложной задачей, так как образы лиц могут иметь различные размеры и располагаться в разных областях изображения; лица могут быть видны под различными углами и могут различаться по цвету. Одна из систем для поиска образов лиц была разработана в Университете Карнеги-Меллон. В этой системе цветные изображения преобразовывались в полутоновые. При этом устранялись цветовые различия, выполнялась нормировка освещенности и расширение диапазона яркостей посредством выравнивания гистограммы. Затем для распознавания лиц использовался классификатор на основе искусственной нейронной сети, которая была обучена на 16000 изображениях, как содержащих, так и не содержащих лица людей.

В качестве входных данных нейронная сеть принимала изображение, состоящее из  $20 \times 20 = 400$  значений яркости, и классифицировала это изображение как содержащее или не содержащее образ лица. Функционирование нейронной сети сложно представить в виде эквивалентного логического алгоритма. Но экспериментальный анализ чувствительности системы показал, что результаты работы сети при обработке изображений размерами  $20 \times 20$  пикселей сильнее всего зависели от образов глаз, затем от области носа, и менее сильно — от области рта. Система из Университета Карнеги-Меллон позволяла обнаруживать



**Рис. 8.8.** Лица, найденные на изображении обнаружителем лиц на основе нейронной сети. (С любезного разрешения Henry Rowley, Takeo Kanade.)

большинство, хотя и не все, фронтальных изображений лиц (см. рис. 8.8). Этот метод не удастся расширить на объекты произвольной формы, если только для них нельзя сформировать подробный и специфический шаблон, в некотором смысле похожий на шаблон, описывающий расположение глаз, носа и рта на полутонных изображениях лиц.

**Обнаружение образов человеческих тел.** Для обнаружения объекта на изображении можно выполнять обнаружение областей изображений, по цвету и текстуре похожих на области искомого объекта. Одна из первых работ в данном направлении касалась поиска на изображениях образов обнаженных людей. В качестве приложения этой задачи можно назвать фильтрацию порнографических изображений из результатов обработки поискового запроса. В методе, который предложен в статье [Fleck, Forsyth, Bregler, 1996] в 1996 г., выполнялись два основных шага: (1) обнаружение больших областей, похожих на образы человеческих тел и (2) группировка этих областей для обнаружения образов тел.

Фильтр для устранения изображений с образами тел работает на уровне отдельных пикселей. Исходное RGB-изображение преобразуется в логарифмическое цветовое пространство с помощью преобразований:

$$I = L(G) \quad (8.10)$$

$$R_g = L(R) - L(G) \quad (8.11)$$

$$B_y = L(B) - \frac{L(G) + L(R)}{2} \quad (8.12)$$



где  $L(x)$  определено как

$$L(x) = 105 \log_{10}(x + 1 + n) \quad (8.13)$$

и  $n$  является случайным значением шума в диапазоне  $[0,1]$ . Компонента  $I$  используется для генерации текстурной амплитудной карты (texture amplitude map) согласно следующему выражению:

$$texture = med_2(|I - med_1(I)|) \quad (8.14)$$

где  $med_1$  и  $med_2$  — это два отдельных медианных фильтра различных размеров (размер  $med_2$  в 1.5 раза больше размера  $med_1$ ). Тектурная амплитудная карта используется для поиска областей с незначительно выраженной текстурой. Например, области кожи на изображениях людей обычно имеют очень гладкую текстуру.

Тон (hue) и насыщенность (saturation) используются для выбора областей, цвет которых похож на цвет человеческой кожи. Изображения  $R_g$  и  $B_y$  перед использованием в операциях обработки также подвергаются медианной фильтрации. Преобразование из логарифмического цветового пространства в значения тона и насыщенности описывается выражениями:

$$hue = atan(R_g, B_y) \quad (8.15)$$

$$saturation = \sqrt{R_g^2 + B_y^2} \quad (8.16)$$

Если значение пиксела попадает в один из двух следующих диапазонов, то он помечается как пиксел области с изображением кожи человека.

1.  $texture < 5$ ,  $110 < hue < 150$ ,  $20 < saturation < 60$
2.  $texture < 5$ ,  $130 < hue < 170$ ,  $30 < saturation < 130$

Отметим, что все приведенные константы взяты из опубликованной работы и могут быть изменены при работе с различными множествами изображений или по желанию пользователя.

Пикселы, принадлежащие областям с изображением кожи, отмечаются на бинарном изображении. Для уменьшения шумовых помех к этому изображению можно применить морфологическую операцию замыкания. После обнаружения областей человеческой кожи на изображении их можно проверить по двум критериям:

1. Имеют ли эти области достаточную площадь, чтобы изображение можно было считать порнографическим (в опубликованной работе использовалось пороговое значение, равное 30% площади всего изображения).



2. Соответствует ли пространственное расположение областей кожи возможным образам частей человеческого тела.

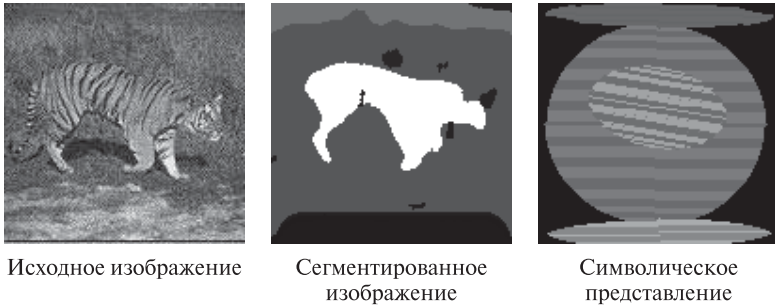
---

### Упражнение 8.7. Обнаружение образов человеческих лиц и тел

Напишите программу для обнаружения образов человеческих тел посредством поиска областей цвета человеческой кожи. Выберите области заранее заданного размера и попытайтесь найти на них характерные признаки образа лица. Это могут быть глаза, нос и рот (именно в такой последовательности). На основе найденных характерных признаков назначьте каждой области вероятность того, что она является образом человеческого лица.

**Пространственные отношения.** После распознавания объектов можно определить пространственные отношения между ними, описывающие взаимное расположение объектов на изображении. В таком случае запросы можно формулировать, задавая некоторое множество именованных объектов и пространственные отношения между ними. Подобные методы поиска изображений являются наиболее сложными. В недавней работе из Беркли [Forsyth и др., 1996] и в похожей работе из Санта-Барбары [Ma, Manjunath, 1997] исследователи продемонстрировали использование цветовых и текстурных характеристик для сегментации изображений на области, которые с большой вероятностью соответствуют объектам или фону сцены. Таким способом можно найти объекты, которые хорошо выделяются на фоне и имеют известный цвет и/или текстуру. Например, это могут быть образы тигров или зебр на изображениях природных сцен. Также этот метод позволяет изолировать хорошо заметный фон (такой, как джунгли, небо или побережье пляжа).

На рис. 8.9 приведен пример применения сегментации такого типа. Слева показано исходное цветное изображение. В центре приведены результаты сегментации изображения на области. Справа показано символическое представление этого изображения, в котором области интереса отмечены эллипсами. Это представление можно использовать для построения реляционного графа, у которого узлы содержат результаты классификации областей, а ребра представляют пространственные отношения между областями. После этого для построения реляционных мер расстояния для поиска изображений можно применять методы символического сопоставления. В описанной здесь системе сегментации такие операции не были предусмотрены. Система поиска, которая могла бы использовать такое представление в качестве входных данных, описана в статье [Del Vimbo, 1993]. Эта система позволяет оператору построить запрос, размещая выбранные пиктограммы в заданных пространственных отношениях на экране запроса. Затем система находит изображения, объекты на которых расположены в таких отношениях, что были указаны в запросе пользователем. На рис. 8.10 приведен пример ре-



**Рис. 8.9.** Объекты и пространственные отношения между ними можно выделить из изображений и использовать при поиске. (С любезного разрешения Corel Stock Photos.) См. цветную вставку.



**Рис. 8.10.** Результаты обработки запроса с указанием пространственных отношений. (С любезного разрешения Alberto Del Bimbo и IEEE. Воспроизведено из «Symbolic Description and Visual Querying of Image Sequences using Spatio-Temporal Logic,» by A. Del Bimbo, E. Vicario, D. Zingoni, *IEEE Transactions on Knowledge and Data Engineering*, vol. 7, no. 4, Aug. 1995. © 1995 IEEE.)

зультатов поиска с использованием запроса с пространственными отношениями.

**Упражнение 8.8.** Поиск с указанием объектов и отношений между ними  
 Воспользуйтесь готовой программой или напишите собственную программу для сегментации цветного изображения на области по цвету и, если возможно, по текстуре. Обработайте с помощью этой программы набор тестовых изображений. Объект каждого класса, например, тигр, небо или джунгли, должен присутствовать на нескольких изображениях тестового набора. Обучите алгоритм классификации на этих известных областях согласно цветовым и текстурным свойствам областей. Напишите программу, которая использует процедуры сегментации и классификации для формирования множества маркированных областей входного изображения и затем вычисляет пространственные отноше-

ния *выше*, *ниже*, *слева-от*, *справа-от* и *смежный-с* для пар областей. Затем разработайте интерактивную оболочку, которая позволяет оператору ввести структуру в виде графа, в котором узлы представляют объекты из тестового набора, а ребра — требуемые пространственные отношения. Программа должна находить в базе данных все изображения, соответствующие запросу пользователя.

## 8.5 ОРГАНИЗАЦИЯ БАЗЫ ДАННЫХ

Большие базы данных изображений, как и другие крупные базы данных, слишком велики для того, чтобы при обработке поискового запроса можно было бы перебирать все изображения, хранящиеся в базе данных. Вместо этого изображения должны быть организованы и индексируются так, чтобы только часть из них рассматривалась при обработке любого запроса. Существует ряд стандартных методов для индексации числовых и текстовых атрибутов, которые применяются в большинстве реляционных систем управления базами данных (СУБД). Известны также методы для индексации пространственных данных. Они используются, например, в геоинформационных системах. В рамках современных исследований также разрабатываются методы индексации изображений, рассчитанные на использование в системах поиска изображений на основе содержания.

### 8.5.1 Стандартные индексы

В большинстве реляционных СУБД пользователь может указать атрибут, по которому необходимо построить индекс. Обычно этот атрибут является ключевым значением, связанным с каждой записью таблицы базы данных. Например, если в базе данных хранятся записи о служащих некоторой компании, то в качестве атрибута для индексации этих данных мог бы быть выбран номер социального страхования. Так как у каждого служащего имеется уникальный номер страхования, то этот атрибут может быть выбран в качестве *первичного ключа*. Если данные часто извлекаются по каким-либо другим атрибутам, например, по фамилии служащего, то по этим атрибутам также можно построить отдельные индексы.

В реляционных базах данных *индекс* представляет собой структуру данных, которая используется системой для поиска по заданному значению атрибута и позволяет быстро найти все множество записей базы данных, имеющих атрибут с заданным значением. В реляционных СУБД часто используются индексы двух типов: хэш-индексы (hash index) и индексы на основе В-деревьев (B-tree index). Хэш-индексы обеспечивают быстрое обнаружение множества записей, имеющих атрибуты с тем же значением, которое задано в запросе. Индексы на основе В- или В<sup>+</sup>-деревьев позволяют быстро найти записи, у которых атрибуты имеют значения, лежащие в некотором диапазоне, указанном в запросе.



Рис. 8.11. Пример хэш-индекса.

**Хэш-индексы.** В реализации хэш-индексов для доступа к большому множеству записей базы данных применяются *хэш-таблицы*. При этом предполагается, что имеется большое множество возможных значений ключей, но в базе данных одновременно всегда будет присутствовать только часть из них. Допустим, что база данных содержит файл с  $N$  записями. Каждая запись состоит из нескольких различных полей, среди которых одно поле предназначено для хранения ключевых значений. Механизм доступа к хэш-индексу предполагает использование *хэш-функции*, которая преобразует любое значение ключа в адрес в файле, который содержит запись (или хранит ссылку на запись) базы данных, содержащую заданное ключевое значение. Если ключевые значения являются числами, то одной из простейших хэш-функций может быть  $f(x) = x \bmod N$ . Эта функция уменьшает значение ключа  $x$  посредством взятия остатка от целочисленного деления на  $N$  и использует это значение как номер записи для доступа к ней. На рис. 8.11 приведена иллюстрация использования хэш-индекса для базы данных с числовыми ключами. Хэш-таблица содержит десять элементов, пронумерованных от 0 до 9 (на практике хэш-таблицы могут быть намного больше). В качестве хэш-функции используется функция  $f(x) = x \bmod 10$ . Показан запрос, выбирающий все записи, у которых ключевой атрибут равен 45. Адреса записей, имеющих ключевой атрибут с таким значением, перечислены в 5-м элементе хэш-таблицы.

Если каждое ключевое значение хэшируется в собственном элементе таблицы, то время доступа по любому ключевому значению постоянно. В общем случае это не так. Несколько различных ключевых значений могут быть хэшированы в одном и том же элементе индекса. Это явление называется *коллизией*. Способы решения этой проблемы описаны в большинстве учебников по структурам данных. Решение, использованное на рис. 8.11, основано на использовании связного списка для хранения адресов всех записей, ключи которых хэшированы в одном и том же элементе хэш-таблицы. Конечный результат — в любом случае — состоит в том, что при поиске в базе данных просматривается небольшое количество данных, существенно меньшее, чем при простом непосредственном доступе. Для «хороших» хэш-функций и «не

слишком заполненных» хэш-таблиц время доступа остается примерно постоянным при любом значении ключа. Применение хэширования наиболее хорошо подходит для обработки запросов с ограничениями вида  $KEY = VALUE$  и менее пригодно для обработки запросов с указанием диапазона значений.

**Индексы на основе  $B^+$ -деревьев**  $B$ -деревья и  $B^+$ -деревья являются сбалансированными деревьями поиска, которые можно использовать для индексации баз данных. Они удобны для обработки запросов, в которых указан диапазон допустимых значений. В  $B$ -деревьях значения ключей и данных хранятся и во внутренних, и в листовых узлах. В  $B^+$ -деревьях данные хранятся только в листовых узлах. Далее будут рассматриваться  $B^+$ -деревья, так как данные в базе данных должны храниться отдельно от индекса.

Дерево поиска порядка  $p$  представляет собой дерево, в котором каждый узел содержит не более  $p - 1$  ключевых значений и  $p$  указателей.  $B^+$ -дерево является деревом поиска, у которого форматы представления внутренних и листовых узлов различаются. Каждый внутренний узел  $B^+$ -дерева должен удовлетворять следующим условиям:

1. Узел имеет формат  $\langle P_1, K_1, P_2, K_2, \dots, P_{q-1}, K_{q-1}, P_q \rangle$ , где каждое значение  $P_i$  является указателем на другой узел, а каждое  $K_i$  является ключевым значением. Можно считать, что  $P_{i-1}$  указывает на поддерево, у которого узлы содержат ключи, значения которых меньше или равны  $K_i$ , а  $P_i$  указывает на поддерево, у которого ключи имеют значения, превышающие  $K_i$ .
2. Если узел не корневой, то он содержит минимум  $\lceil (p/2) \rceil$  указателей на поддерева.
3. Корневой узел содержит минимум 2 указателя на поддерева.

Каждый листовой узел  $B^+$ -дерева удовлетворяет следующим условиям:

1. Узел имеет формат  $\langle K_1, Pr_1, K_2, Pr_2, \dots, K_{q-1}, Pr_{q-1}, P_{next} \rangle$ , где  $K_i$  является ключом,  $Pr_i$  — указателем на данные, а  $P_{next}$  — указателем на следующий листовой узел.
2. Указатель  $Pr_i$  указывает или на запись или блок записей, имеющих ключ поиска  $K_i$ .
3. В каждом листовом узле хранится  $\lfloor (p/2) \rfloor$  значений.
4. Все листовые узлы расположены на одном уровне дерева.

Порядок внутренних узлов  $p$  может отличаться от порядка листовых узлов. Часто узлы имеют такой объем, чтобы узел каждого типа помещался в физический блок диска. Использование объемов данных, кратных размеру физического блока, обычно обеспечивает наиболее эффективную передачу данных между диском и оперативной памятью.

При поиске с помощью  $B^+$ -дерева ключа с заданным значением или ключа, принадлежащего заданному диапазону значений, система поиска начинает

обработку с корневого узла дерева. Она считывает этот узел в память и выполняет двоичный поиск среди ключей, хранящихся в данном узле. Если в узле обнаруживается два смежных ключевых значения, между которыми лежит искомое значение, то указатель между этими ключами указывает на поддереву, которое будет содержать искомое ключевое значение или наименьшее значение из указанного диапазона ключей. Если искомое ключевое значение меньше первого ключевого значения узла, то указатель слева от этого ключа ссылается на поддерево для дальнейшего поиска. Аналогично, если искомое ключевое значение превышает последнее ключевое значение в узле, то указатель справа от этого ключа ссылается на поддерево для дальнейшего поиска. После обнаружения подходящего поддерева поиск продолжается с корневого узла этого поддерева. Поиск выполняется рекурсивно до достижения листового узла.

В листовом узле снова выполняется двоичный поиск для обнаружения искомого ключевого значения или начального значения ключа  $K_i$ . Связанный с ним указатель  $P_i$  указывает на запись данных, содержащую это ключевое значение. Если выполняется поиск только одного ключевого значения, то на этом этапе можно вернуть найденную запись данных. Если же выполняется поиск по диапазону, то указатели  $P_{next}$  на следующие листовые узлы могут использоваться для нахождения оставшихся записей данных, пока не будет обнаружен конец заданного в запросе диапазона ключей.

На рис. 8.12 приведен пример  $B^+$ -дерева, используемого для индексации записей базы данных с числовыми ключевыми значениями и данными изображений. Внутренние узлы обозначены сплошными прямоугольниками, а листовые узлы — штриховыми прямоугольниками. Корневой узел указывает на три различных поддерева: одно поддерево хранит значения ключей, меньшие или равные 100, другое хранит значения ключей от 100 до 200, а третье — значения ключей, превышающие 200. На рис. 8.12 показано поддерево с ключевыми значениями от 100 до 200. Корневой узел этого поддерева содержит

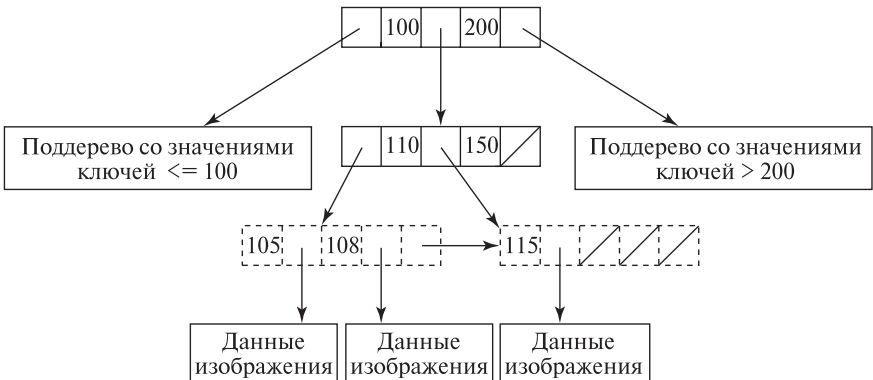


Рис. 8.12. Пример индекса на основе  $B^+$ -дерева.

указатели на два листовых узла: один со значениями ключей, меньшими 110, а другой — со значениями ключей между 110 и 150. Листовые узлы содержат некоторые существующие ключевые значения и ссылки на соответствующие этим значениям файлы с данными изображений.

$V^+$ -деревья являются гибкими и эффективными структурами данных. Они широко используются в реляционных системах управления базами данных. Они могут применяться и в базах данных изображений для индексации отдельных числовых или текстовых полей, сопоставленных с изображением. Индексы данного типа не предназначены для использования с многомерными данными.

### 8.5.2 Пространственная индексация

В информационных системах, обрабатывающих пространственную информацию, хранятся многомерные данные. Для пространственной индексации было предложено несколько различных структур данных.

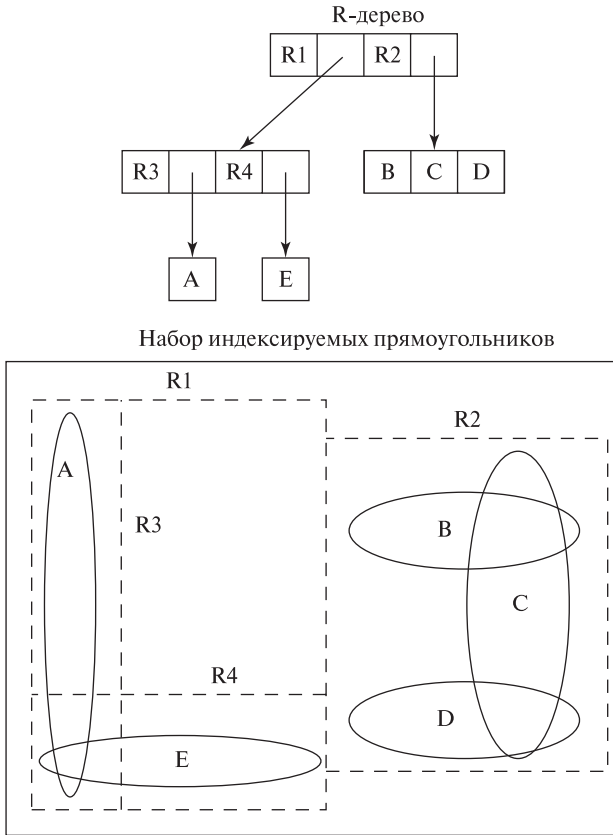
Квадрантные деревья — это иерархические структуры четвертого порядка, позволяющие разбивать пространство поиска двумерных данных на четыре квадранта на каждом уровне дерева. Квадрантные деревья могут использоваться для описания областей бинарных изображений.

$k$ -мерные деревья являются расширением бинарных деревьев поиска, предназначенными для организации поиска среди  $k$ -мерных данных.

$R$ -деревья — это расширение  $B$ -деревьев на данные большей размерности. Они удобны для применения в различных системах, работающих с пространственно упорядоченной информацией. В  $R$ -дереве объект данных индексируется с использованием  $n$ -мерного минимального описывающего прямоугольника (minimum bounding rectangle, MBR), который ограничивает область пространства, занимаемую объектом. Каждый существующий объект данных обозначается с помощью уникального идентификатора (ID). В листовых узлах  $R$ -дерева хранятся идентификаторы объектов данных. Внутренние узлы содержат записи вида (MBR, CHILD), где CHILD — это указатель на нижележащий узел  $R$ -дерева, а MBR охватывает все прямоугольники из записей нижележащих углов. На рис. 8.13 показан пример индекса на основе  $R$ -дерева для набора двумерных объектов. Распределение прямоугольников зависит от порядка формирования  $R$ -дерева и применяемого для этого алгоритма. Существуют несколько разновидностей  $R$ -деревьев, в том числе  $R^+$ -деревья и  $R^*$ -деревья.

### 8.5.3 Индексация для систем поиска изображений на основе содержания, в которых применяется несколько различных мер расстояния

Вышеописанные методы можно применять для индексации изображений, если требуется обеспечить поиск с использованием простых мер расстояния, основанных на одиночных атрибутах или на небольшом количестве атрибутов. Эти методы не подходят для больших универсальных систем, в которых



**Рис. 8.13.** Пример индекса на основе R-дерева для индексации двумерных данных. Эллипсы представляют объекты данных. Индексирующие прямоугольники показаны штриховыми отрезками. Прямоугольник R1 был разбит на прямоугольники R3 и R4, каждый из которых содержит один объект данных. Прямоугольник R2 дальнейшему разбиению подвергнут не был. Он содержит три объекта данных: B, C и D.

пользователь может выбирать различные меры расстояния и комбинировать эти базовые меры для построения новых мер расстояния. Для таких систем необходимы более гибкие методы организации и индексации данных. Если базовые меры расстояния являются метрическими, то для построения пользовательского метода индексации может использоваться свойство неравенства треугольника. Согласно неравенству треугольника, если  $Q$  является изображением из запроса,  $I$  — изображением из базы данных, а  $K$  — специально выбранным ключевым изображением, то

$$d(I, Q) \geq |d(I, K) - d(Q, K)|$$



для любой меры расстояния между изображениями  $\mathbf{d}$ . Таким образом, сравнивая изображения из базы данных и из запроса с третьим ключевым изображением, можно определить нижнюю границу расстояния между изображением из запроса и изображением из базы данных.

Рассмотрим сначала случай одной метрики расстояния  $\mathbf{d}$ . Предположим, что из базы данных был выбран набор ключевых изображений. Очевидно, они должны представлять различные классы сцен из базы данных. Изображение из запроса  $\mathbf{Q}$  сравнивается с каждым из этих ключевых изображений  $\mathbf{K}_1, \mathbf{K}_2, \dots, \mathbf{K}_M$ . В результате будет получено множество расстояний  $\mathbf{d}(\mathbf{Q}, \mathbf{K}_1), \mathbf{d}(\mathbf{Q}, \mathbf{K}_2), \dots, \mathbf{d}(\mathbf{Q}, \mathbf{K}_M)$ . Предположим, что в запросе пользователь указал поиск всех изображений, у которых расстояние от запроса  $\mathbf{Q}$  меньше  $T$ . Тогда для каждого ключа  $\mathbf{K}_i$ , можно сразу исключить из рассмотрения все изображения  $\mathbf{I}$  такие, для которых

$$|d(\mathbf{I}, \mathbf{K}_i) - d(\mathbf{Q}, \mathbf{K}_i)| > T$$

так как уже известно, что  $\mathbf{d}(\mathbf{I}, \mathbf{Q})$  будет слишком большим. Для реализации данного подхода была разработана специальная структура данных, называемая *треугольным деревом* (triangle-tree). Она позволяет при обработке поискового запроса исключить из рассмотрения большинство изображений из базы данных без непосредственного сравнения с запросом. Этот метод также был расширен для работы с динамически определяемыми мерами расстояния, которые могут быть выражены в виде линейной или булевой комбинации базовых мер расстояния.

### Упражнение 8.9. Индексация

Предположим, что набор изображений требуется проиндексировать по значениям энергетических текстурных характеристик Лавса. Предложите способ использования R-деревьев для подобной индексации.

## 8.6 ЛИТЕРАТУРА

Поиск изображений на основе содержания является относительно новой прикладной областью, в которой применяются методы компьютерного зрения. Первая коммерческая система общего назначения, QBIC, была разработана в фирме IBM. Она рассмотрена в статье [Niblack, 1993]. В этой системе для поиска изображений использовались признаки цвета, текстуры и формы. Еще одной из первых систем поиска изображений была система ART MUSEUM, [Kato и др., 1992]. Эта система обеспечивала поиск изображений картин на основе эскизов, нарисованных пользователем. Минка и Пикард (Minka, Picard, 1996) показали, как можно повысить производительность системы поиска с помощью анализа команд, выбираемых пользователем (релевантная обратная связь). В статье [Del Vimbo, Pala, Santini, 1994] представлена система для

поиска изображений, на которых присутствуют объекты, по форме похожие на указанную пользователем область. Для сопоставления описаний формы в этой системе применялся метод эластичного сопоставления.

В работе [Fleck, Forsyth, Bregler, 1996] рассмотрен алгоритм для поиска образов обнаженных людей на изображениях с использованием цветовых свойств и пространственных отношений между областями изображений. Позднее методы, предложенные в этой работе, были расширены для поиска других объектов, например, лошадей. В статье [Rowley, Baluja, Kanade, 1996] предложен метод поиска образов человеческих лиц с помощью нейронной сети, обученной на нескольких тысячах изображений. В статьях [Carson, Belongie, Greenspan, Malik, 1997] и [Ma, Manjunath, 1997] рассмотрен метод обработки поисковых запросов на основе анализа областей с использованием алгоритмов сегментации по цвету и текстуре для формирования областей интереса. В работах [Baeza-Yates и др., 1994] и [Berman, 1994] предложены структуры данных для использования при сопоставлении по близости объектов. Эти методы были расширены для использования в системе поиска изображений, представленной в работе [Berman, Shapiro, 1999]. В книге [Samet, 1990] описаны пространственные структуры данных общего назначения.

1. Baeza-Yates, R., W. Cunto, U. Manber, and S. Wu. 1994. Proximity matching using fixed queries trees. *Combinatorial Pattern Matching*. Springer-Verlag, New York, 198–212.
2. Berman, A. P. 1994. A new data structure for fast approximate matching, technical report 94-03-02. Department of Computer Science and Engineering, University of Washington.
3. Berman, A. P., and L. G. Shapiro. 1999. A flexible image database system for content-based retrieval. *Comput. Vision and Image Understanding*, v. 75(1–2):175–195.
4. Carson, C., S. Belongie, H. Greenspan, and J. Malik. 1997. Region-based image querying. *Proc. IEEE Workshop on Content-Based Access of Image and Video Libraries*, San Juan, Puerto Rico.
5. Del Bimbo, A., E. Vicario, and D. Zingoni. 1993. Sequence retrieval by contents through spatio temporal indexing. *IEEE Symp. Visual Lang.*, 88–92.
6. Del Bimbo, A., P. Pala, and S. Santini. 1994. Visual image retrieval by elastic deformation of object sketches. *IEEE Symp. Visual Lang.*, 216–223.
7. Fleck, M. M., D. A. Forsyth, and C. Bregler. 1996. Finding naked people. *Proc. Euro. Conf. Comput. Vision*. Springer-Verlag, New York, 593–602.
8. Forsyth, D. A., J. Malik, M. M. Fleck, H. Greenspan, T. Leung, S. Belongie, C. Carson, and C. Bregler. 1996. Finding pictures of objects in large collections of images. *Proc. 2nd Inter. Workshop on Object Representation in Comput. Vision* (April 1996).
9. Kato, T., T. Kurita, N. Otsu, and K. Hirata. 1992. A sketch retrieval method for full color image database. *11th Inter. Conf. Pattern Recog.*, 530–533.

10. Ma, W. Y., and B. S. Manjunath. 1999. Netra: a toolbox for navigating large image databases. *Multimedia Systems*, v. 7(3):184–198.
11. Minka, T. P., and R. W. Picard. 1996. Interactive learning with a society of models. *Proc. CVPR-96*, 447–452.
12. Niblack, W., and others. 1993. The QBIC project: Querying images by content using color, texture, and shape. *SPIE Proc. Storage and Retrieval for Image and Video Databases*, 173–187.
13. Rowley, H., S. Baluja, and T. Kanade. 1996. *Human Face Detection in Visual Scenes*. Carnegie-Mellon University, Pittsburgh, PA.
14. Samet, H. 1990. *The Design and Analysis of Spatial Data Structure*. Addison-Wesley, Reading, MA.

# Движение на двумерных изображениях

Последовательности изображений позволяют наблюдать изменяющиеся сцены. Например, при обучении игре в гольф можно пользоваться видеосъемками игр опытных игроков; или при изучении роста растений можно рассматривать последовательности изображений, полученных с интервалом в несколько часов. Динамические явления, наблюдаемые через некоторые временные промежутки, объясняются движением объектов, наблюдателя, или их обоих. Присутствующие в последовательности изображений изменения могут использоваться в качестве характерных признаков для обнаружения движущихся объектов или для определения их траекторий. В случае движения наблюдателя в пределах относительно неподвижной сцены изменения на изображениях позволяют определить траекторию наблюдателя в трехмерном пространстве.

Изменения яркости или цвета пикселей изображения оказываются важными характерными признаками, полезными для обнаружения и распознавания объектов. В результате анализа движения объекта можно определить его форму и ряд других свойств, например, скорость или выполняемые объектом операции. В некоторых задачах (например, в задаче слежения за транспортным потоком) анализ движения объекта на протяжении заданного промежутка времени является конечной целью.

В настоящее время накоплено огромное количество видеоданных о событиях и объектах реального мира. Возникает необходимость в автоматических средствах для сегментации этих последовательностей изображений с целью выделения важных событий или сцен. Это упростит поиск нужных сцен, обработку и редактирование соответствующих видеоданных.

В данной главе основное внимание уделяется обнаружению движения на двумерных изображениях и видеопоследовательностях, а также методам обработки для выделения характерных признаков. Обсуждаются подходы к решению перечисленных выше прикладных задач. Вопросы, касающиеся трехмерных структур и получения по двумерным изображениям информации о движении в трехмерном пространстве, будут рассматриваться в главе 13.

## 9.1 ЯВЛЕНИЯ И ПРИКЛАДНЫЕ ЗАДАЧИ, СВЯЗАННЫЕ С ДВИЖЕНИЕМ НА ИЗОБРАЖЕНИЯХ

Сначала рассмотрим разновидности движения, которые могут наблюдаться на последовательности изображений, а также связанные с этими случаями важные прикладные задачи. Спектр этих задач довольно широк — от простого обнаружения движущегося объекта до определения относительного движения и формы нескольких движущихся объектов.

Можно выделить 4 общих случая движения (далее термины *камера* и *наблюдатель* будем считать синонимами):

- Неподвижная камера, один движущийся объект, постоянный фон.
- Неподвижная камера, несколько движущихся объектов, постоянный фон.
- Движущаяся камера, относительно постоянный фон.
- Движущаяся камера, несколько движущихся объектов.

В простейшем случае неподвижная камера наблюдает относительно постоянный фон. Движущиеся на этом фоне объекты приводят к изменению яркости/цвета пикселей в соответствующих областях изображений. По этим пикселям можно определить форму, скорость и траекторию движущегося объекта. Датчики подобного типа широко применяются в охранных и пропускных системах. В домашних условиях такие датчики можно использовать для автоматического включения освещения при обнаружении «значимого» движения, которое может возникнуть вследствие прихода домой владельца или постороннего человека. Простейшие датчики движения могут также применяться на производстве для определения наличия какой-либо детали в рабочем пространстве или в системах контроля транспортных потоков для обнаружения движущихся автомобилей.

Неподвижная камера позволяет получать данные о движении одного или нескольких объектов. Движущиеся объекты можно проследивать в течение времени для определения траектории, которая, в свою очередь, может дать некоторую информацию о поведении объекта. Например, информацию о поведении людей, входящих в вестибюль некоторого учреждения или работающих рядом в одном помещении. С помощью нескольких камер можно получить несколько видов одного и того же объекта и вычислить по ним траекторию объекта в трехмерном пространстве. В качестве задач такого типа можно назвать анализ движений спортсменов или больных в процессе реабилитации. Известно, что сейчас разрабатывается система для слежения за игроками и мячом во время теннисной игры.

Движущаяся камера создает изменения на изображениях вследствие собственного движения, даже если трехмерная сцена неподвижна. Движущаяся камера применяется в нескольких целях. Во-первых, она позволяет получить больше сведений об окружающей среде, чем доступно с единственной точки наблюдения. Например, панорамная камера используется для получения

широкоугольного (панорамного) вида сцены. Во-вторых, становятся доступны данные для вычисления относительной глубины объектов сцены, так как изображения близких к камере объектов изменяются быстрее, чем далеких. В-третьих, обеспечиваются данные для восприятия/измерения трехмерной формы близких объектов — несколько точек наблюдения позволяют выполнить триангуляционные измерения, как в бинокулярной стереосистеме. При обработке видео- или фотосъемки часто бывает нужно определить моменты времени, когда камера перемещалась или менялось ее фокусное расстояние (масштаб изображения). В данном случае содержание сцены несущественно, важно определить именно способ наблюдения сцены.

В наиболее сложных задачах анализа движения на изображениях используются несколько датчиков и сцены с несколькими движущимися объектами, причем затруднительно выделить какой-либо постоянный фон. Например, это задача управления мобильным роботом в транспортном потоке. Пример еще одной интересной задачи — совместное использование нескольких камер для прослеживания в рабочем пространстве нескольких движущихся объектов, при этом приходится выполнять сопоставление изображений с разных камер.

В следующих разделах описываются несколько методов обработки последовательностей из двух и более изображений с целью обнаружения изменений, вызванных движением, или для анализа самих движущихся объектов или их траекторий.

---

### Упражнение 9.1

Расположите поблизости от себя детектор движения, предназначенный для автоматического включения освещения. Подобные устройства часто используются при входе в гараж или в дом. Убедитесь, что свет включается при быстром входе в область контроля. (а) Выясните, что будет, если входить в область контроля очень медленно. Можете ли вы обмануть детектор движения, чтобы он вас не обнаружил? (б) Что по результатам этого эксперимента можно сказать о принципе работы детектора движения? (в) Как это соотносится с эпизодом про тиранозавра из фильма «Парк юрского периода»?

## 9.2 ВЫЧИТАНИЕ ИЗОБРАЖЕНИЙ

Вычитание изображений было упомянуто в гл. 1 в качестве способа обнаружения объектов, движущихся на постоянном фоне. Предположим, что видеокамера с кадровой частотой 30 кадров в секунду наблюдает ленту производственного конвейера. Эта лента выглядит как однородный темный фон. Когда в поле зрения камеры перемещаются светлые объекты, то передние и задние края этих объектов продвигаются в течение кадра всего на несколько пикселей. Если вычесть изображение  $I_t$  из предыдущего изображения  $I_{t-1}$ , то на разностном изображении только эти пиксели будут отличны от 0 и поэтому хорошо заметны.



**Рис. 9.1.** (Слева) в ранее свободной области рабочего пространства появляется фигура человека. (В центре) вычитание изображений выявило три области изменений: фигура человека, закрывшая изображение фона, область двери и след кадровой развертки на мониторе. (Справа) изменения, вызванные появлением человека, считаются значимыми и фиксируются. Изменения в двух других областях считаются допустимыми и поэтому игнорируются. (С любезного разрешения S.-W. Chen.)

На рис. 9.1 показаны результаты вычитания кадров на временном промежутке длительностью несколько секунд в задаче контроля рабочего пространства (т. е. обнаружения доступа в контролируемое пространство). В данном случае фон является неоднородным, но постоянным. Его можно запомнить путем анализа некоторой последовательности кадров (например, просто осреднением кадров). Когда в рабочее пространство входит человек, то в соответствующей области изображения изменяются значения пикселей. Эти изменения можно найти вычитанием изображений (рис. 9.1). На рис. 9.1 ограничивающий прямоугольник обозначает прямоугольную область, в которой обнаружены «значительные» изменения. С помощью дальнейшей обработки изображения в пределах этого прямоугольника можно выяснить форму или даже тип возникшего объекта. На рис. 9.1, в центре, в действительности есть три области изменений. Они соответствуют (1) человеку, (2) двери, открытой человеком при входе, и (3) компьютерному монитору. Система наблюдения может быть настроена так, что расположение двери и мониторов будет ей известно, и изменения в этих областях могут не отслеживаться или игнорироваться (считаться «незначительными»). Или, наоборот, область двери можно отслеживать особенно тщательно, а компьютерные мониторы игнорировать. Аналогичные задачи возникают при наблюдении и учете парковочных мест, при наблюдении потока автомобилей на улицах или людей в комнатах.

В алгоритме 9.1 показаны основные этапы обнаружения изменений на постоянном фоне с помощью вычитания изображений. На отдельных этапах алгоритма выполняются операции, описанные в гл. 3.

## Упражнение 9.2

Для выполнения этого упражнения необходим компьютер с подключенной к нему видеокамерой и программным обеспечением для ввода отдельных кадров. Напишите программу для слежения за областью рабочего стола около

### Обнаружение различий между двумя изображениями

**Входные данные:** Два полутоновых изображения  $I_t[r, c]$  и  $I_{t-\Delta}[r, c]$ , введенных с промежутком  $\Delta$  секунд, и значение яркостного порога  $\tau$ .

**Выходные данные:** Бинарное изображение  $I_{out}[r, c]$  и массив  $\mathbf{B}$  с координатами ограничивающих прямоугольников.

1. Для всех пикселей  $[r, c]$  входных изображений  
 $I_{out}[r, c] = 1$ , если  $(|I_t[r, c] - I_{t-\Delta}[r, c]| > \tau)$ , или  
 $I_{out}[r, c] = 0$  в противном случае.
2. На выходном изображении  $I_{out}$  найти все связанные компоненты.
3. Для устранения шума удалить все мелкие области.
4. Для слияния близких областей выполнить замыкание изображения  $I_{out}$  с использованием структурирующего элемента в виде небольшого диска.
5. Вычислить координаты описывающих прямоугольников для всех оставшихся областей, содержащих изменившиеся пиксели.
6. Вернуть изображение  $I_{out}[r, c]$  и массив  $\mathbf{B}$  с координатами ограничивающих прямоугольников для областей с обнаруженными изменениями яркости.

**Алгоритм 9.1.** Обнаружение изменений на основе вычитания фона.

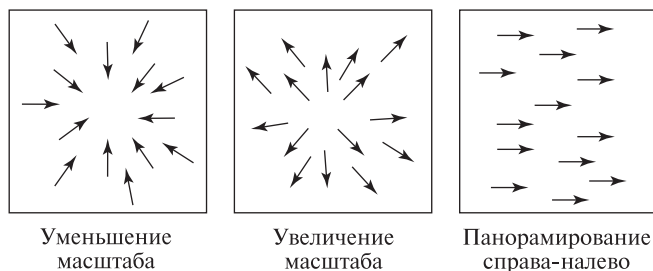
компьютера. Программа должна получать кадры, вычислять гистограмму для каждого из них и сигнализировать в случае значительного изменения гистограммы. Проверьте свою программу на нескольких неподвижных сценах, а затем переместите в рабочем пространстве какой-нибудь объект.

## 9.3 ВЫЧИСЛЕНИЕ ВЕКТОРОВ ПЕРЕМЕЩЕНИЯ

Перемещение трехмерных точек сцены приводит к перемещению их образов на изображении. На рис. 9.2 показаны три типичных случая. Уменьшение масштаба наблюдается при уменьшении фокусного расстояния неподвижной камеры или при удалении камеры с фиксированным фокусным расстоянием от объектов сцены. Образ точки на оптической оси на изображении не перемещается. Эта точка изображения называется *фокусом схода*. *Увеличение масштаба* (zoom in) происходит при увеличении фокусного расстояния неподвижной камеры или при перемещении камеры по направлению к точке схода. Панорамирование камеры (напоминает поворот головы наблюдателя) приводит к смещению образов трехмерных точек (рис. 9.2, справа).

**70 Определение.** **Полеом движения** называется двумерный массив двумерных векторов, представляющих движение точек трехмерной сцены





**Рис. 9.2.** Влияние масштабирования и панорамирования на местоположение характерных признаков изображения. Масштабирование напоминает приближение/удаление наблюдателя от объектов сцены, а панорамирование — поворот наблюдателя.

(см. рис. 9.2). Векторы перемещения на изображении описывают смещения образов движущихся трехмерных точек. Каждый вектор перемещения можно построить так: начало вектора находится в образе трехмерной точки, полученном в момент времени  $t$ , а конец — в образе той же точки в момент времени  $t + \Delta$ . Или, по-другому, каждый вектор соответствует мгновенной скорости точки в момент времени  $t$ .

**71 Определение. Фокус расширения** (focus of expansion, FOE) — это точка изображения, из которой расходятся все вектора поля движения. Обычно ей соответствует та точка трехмерной сцены, по направлению к которой движется камера. Точка, к которой направлены все вектора поля движения, называется **фокусом схода** (focus of contraction, FOC). Обычно это образ трехмерной точки, от которой удаляется камера.

Поле движения используется как для распознавания объектов, так и для анализа их движения. На значения интенсивности пикселей изображения при вычислении векторов перемещения обычно накладывается ограничение в одной из двух форм. Во-первых, можно предположить, что интенсивность образа трехмерной точки  $P$  и некоторой ее окрестности в течение временного интервала  $(t_1, t_2)$ , на котором выполняется оценка перемещения точки  $P$ , остается примерно постоянной. По-другому, можно предположить, что на временном интервале  $(t_1, t_2)$  постоянна разность интенсивностей на образах краев объектов.

**72 Определение. Поток изображения** (image flow) — это поле движения, вычисленное в предположении о постоянстве интенсивности изображений в окрестностях соответствующих точек.

Далее в этой главе будут описаны два метода для вычисления потока изображения, но сначала рассмотрим игровое приложение, в котором используются поля движения.

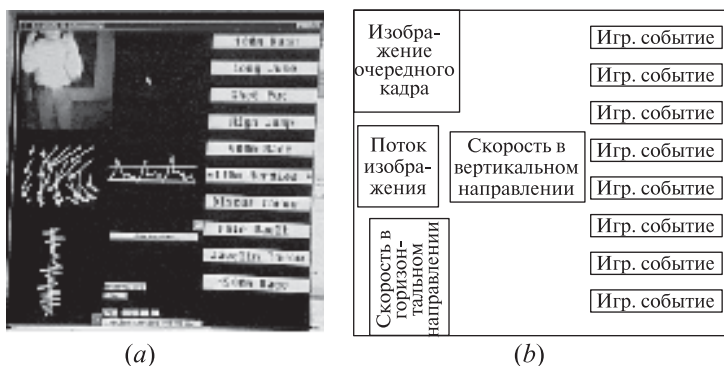
### 9.3.1 Игра Decathlete

В качестве примера применения методов анализа движения рассмотрим систему управления игрой *Decathlete* для игровой приставки Sega Saturn. Эта система была разработана в лаборатории MERL (Mitsubishi Electronic Research Laboratory, Кембридж, шт. Массачусетс). В ней для управления движением игрового персонажа вместо нажатий клавиш используется расчет потока изображения с помощью камеры низкого разрешения, наблюдающей реальные движения игрока-человека. Камера разработана на основе КМОП-датчика фирмы Mitsubishi Electric и в ней предусмотрены возможности аппаратной обработки изображений. В рассматриваемом примере игровой персонаж — это бегун, участвующий в беге с барьерами против бегуна, управляемого компьютером. На рис. 9.3 игрок-человек находится слева, руками он имитирует движения бегуна. Чем быстрее движения, тем быстрее бежит игровой персонаж. В соответствующие моменты времени он должен прыгать через барьеры — для этого игрок «прыгает», поднимая вверх оба кулака. Экран игры *Decathlete* на рис. 9.3 показан на мониторе в правой части. В нижнем правом углу рис. 9.3 можно заметить видеокамеру, с помощью которой анализируется поведение игрока. Два человека в центре рис. 9.3 — посторонние наблюдатели.

Процедура анализа движения для управления игрой поясняется на рис. 9.4. На рис. 9.4 (a) показан снимок служебного экрана системы анализа движения в некоторый момент времени, а на рис. 9.4 (b) поясняется структура этого экрана. Слева сверху на рис. 9.4 (a) расположен полученный с помощью видеокамеры кадр с «бегущим» игроком. Слева в средней части показаны векторы перемещения, построенные по нескольким кадрам. Слева внизу показаны



**Рис. 9.3.** Для управления персонажем компьютерной игры игрок, находящийся слева, руками имитирует движения бегуна с препятствиями. Справа расположен игровой экран. В правом нижнем углу видна камера для наблюдения за движениями игрока, которая используется для управления скоростью и прыжками игрового персонажа. (Воспроизведено по *IEEE Computer Graphics*, Vol.18, No.3, May–June 1998 с разрешения IEEE.)



(a)

(b)

**Рис. 9.4.** Иллюстрация к процедуре анализа движения для управления компьютерной игрой «бег с барьерами». На изображении экрана сверху слева показан кадр с «бегущим» игроком, а слева в центре — векторы перемещения, построенные по нескольким кадрам. Моменты прыжков через барьеры определяются по графику скорости движения в вертикальном направлении (график показан в центре). (Воспроизведено по *IEEE Computer Graphics*, Vol.18, No.3, May–June 1998.)

накапливаемые данные о движении относительно кадра в горизонтальном направлении. В центре на рис. 9.4 (a) выводятся аналогичные данные о движении в вертикальном направлении.

Камера должна быть установлена так, чтобы в кадре обе руки игрока были видны, и когда он «бежит», и когда «прыгает». Движение «бег» характеризуется изменениями данных о движении относительно кадра в горизонтальном направлении. Частота этих изменений характеризует скорость движений игрока и влияет на скорость игрового персонажа. При движении «прыжок» изменяется средняя скорость в вертикальном направлении относительно кадра, и если она превышает заданное пороговое значение, то команда «прыжок» посылается компьютерной игре. Пространственное разрешение используемых изображений низкое, а временное — высокое.

Грубый анализ движения, как в игре *Decathlete*, может помочь в разработке интерфейса пользователя на основе распознавания жестов. Например, в будущем у компьютеров может быть предусмотрен ввод данных с помощью языка жестов American Sign Language (ASL) или некоторого более ограниченного языка жестов.

### 9.3.2 Использование соответствующих точек

Для вычисления разреженного поля движения можно пользоваться парами соответствующих точек, найденных на двух изображениях для различных моментов времени  $t_1$  и  $t_1 + \Delta$ . Эти точки должны обладать некоторым характерным признаком, чтобы их можно было обнаружить на обоих изображениях. Например, и на полутоновых, и на цветных изображениях в качестве

соответствующих точек можно выбрать угловые точки. На сегментированных полутоновых изображениях с движущимися областями постоянной формы в качестве соответствующих точек можно искать центры тяжести этих областей. Поиск угловых точек выполняется с помощью масок (например, детектор краев Кирша или «волнистые» маски из базиса операторов Фрея-Чена, см. гл. 5) или *признакового оператора*. Для угловых точек этот оператор вычисляет изменение яркости по вертикальным, горизонтальным и диагональным соседям пиксела  $P$ . Только если минимальная из этих 4-х разностей превышает некоторый порог, то пиксел  $P$  считается точкой интереса (признаковой точкой). Эти операции представлены в алгоритме 9.2. Можно разработать и другие признаковые операторы, например, текстурный оператор (см. Упражнение 9.3).

### Упражнение 9.3. Текстурный признаковый оператор

Поэкспериментируйте с признаковым оператором, основанным на оценке текстуры в области размерами  $n \times n$ . Во-первых, с маской  $3 \times 3$  или  $2 \times 2$  вычислите градиент на всем входном изображении. Во-вторых, выполните пороговую бинаризацию градиентного изображения. Пиксел исходного изображения  $[r, c]$  считается «точкой интереса», только если в каждом из 4 направлений в области  $n \times n$  бинарного изображения  $B[r, c]$  есть существенное изменение. Изменение в направлении  $[\Delta r, \Delta c] = [0, 1], [1, 0], [1, 1], [1, -1]$  вычисляется как сумма  $B[r, c] \otimes B[r + \Delta r, c + \Delta c]$  для всех пикселов в окрестности  $n \times n$  с центром в пикселе  $B[r, c]$ . Оператор  $\otimes$  — это оператор «ИСКЛЮЧАЮЩЕЕ ИЛИ», который возвращает 1 только при различных значениях операндов. Признаковое изображение IN формируется путем присвоения пикселу  $IN[r, c]$  минимального значения среди 4 вычисленных разностей на  $B[r, c]$ . Примените свой оператор к нескольким полутоновым изображениям, в том числе к изображению шахматной доски.

После обнаружения точек интереса  $\{P_j\}$  на изображении  $I_1$  для момента времени  $t$  надо найти соответствующие точки на изображении  $I_2$  для момента времени  $t + \Delta$ . Вместо выполнения аналогичных вычислений и последующего подбора соответствующих пар точек на изображении  $I_2$  проще сразу искать новые положения точек интереса, найденных на  $I_1$ . Для этого можно пользоваться корреляционным методом, описанным в гл. 5. Для очередной точки интереса  $P_j$  с изображения  $I_1$  берем небольшую область вокруг этой точки из  $I_1$  и пытаемся найти для нее положение с максимальной корреляцией на  $I_2$ , предполагая, что возможное перемещение точки ограничено. Поиск в кадре  $I_2$  нового положения точки интереса  $P_j$ , найденной в кадре  $I_1$ , поясняется на рис. 9.5. В качестве нового положения точки интереса (т. е. конца вектора перемещения) запоминается центр  $P_k = [P_{kr}, P_{kc}]$  области  $I_2$ , обнаруженной корреляционным поиском области, взятой с изображения  $I_1$ . Началом вектора перемещения является точка  $P_j = [P_{jr}, P_{jc}]$ . Поиск точки  $P_k$  ограничен прямоугольной областью с размерами  $C \times R$  и координатами строк  $P_{jr} - R \dots P_{jr} + R$

**Обнаружение точек интереса на заданном входном изображении.**

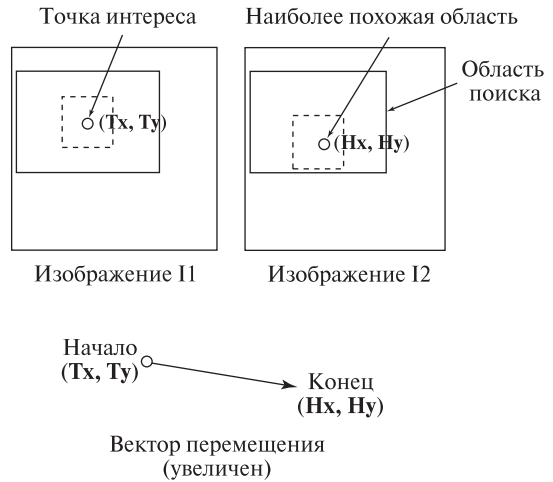
```

procedure detect_corner_points(I, V);
{
  “I[r, c] — входное изображение из MaxRow строк и
  MaxCol столбцов ”
  “V — выходной массив точек интереса изображения I.”
  “ $\tau_1$  — яркостный порог для оператора интереса”
  “ $w$  — половина размера окна признакового оператора”
  for r := 0 to MaxRow - 1
    for c := 0 to MaxCol - 1
      {
        if I[r,c] является граничным пикселом then break;
        else if (interest_operator (I, r, c, w )  $\geq \tau_1$ ) then Добавить
        две точки [(r,c),(r,c)] в массив V;
        “Вторая копия точки (r, c) нужна для резервирования в массиве
        места для координат конца вектора, который будет вычислен позже.”
      }
}
real procedure interest_operator (I, r, c, w)
{
  “ $w$  — половина размера окна оператора”
  v1 := дисперсия яркостей горизонтальных пикселей
            $\mathbf{I}_1[\mathbf{r}, \mathbf{c} - \mathbf{w}] \dots \mathbf{I}_1[\mathbf{r}, \mathbf{c} + \mathbf{w}]$ ;
  v2 := дисперсия яркостей вертикальных пикселей
            $\mathbf{I}_1[\mathbf{r} - \mathbf{w}, \mathbf{c}] \dots \mathbf{I}_1[\mathbf{r} + \mathbf{w}, \mathbf{c}]$ ;
  v3 := дисперсия яркостей диагональных пикселей
            $\mathbf{I}_1[\mathbf{r} - \mathbf{w}, \mathbf{c} - \mathbf{w}] \dots \mathbf{I}_1[\mathbf{r} + \mathbf{w}, \mathbf{c} + \mathbf{w}]$ ;
  v4 := дисперсия яркостей диагональных пикселей
            $\mathbf{I}_1[\mathbf{r} - \mathbf{w}, \mathbf{c} + \mathbf{w}] \dots \mathbf{I}_1[\mathbf{r} + \mathbf{w}, \mathbf{c} - \mathbf{w}]$ ;
  return minimum {v1, v2, v3, v4};
}

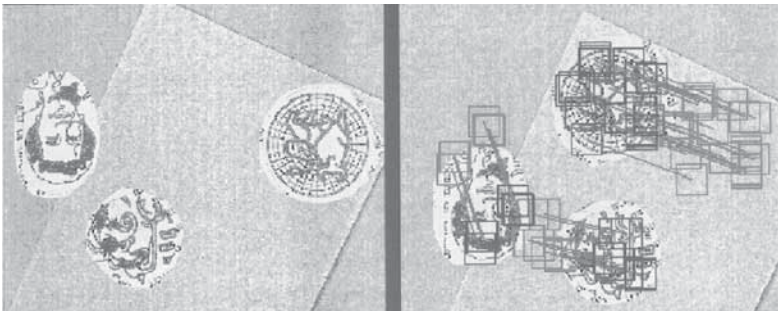
```

**Алгоритм 9.2.** Поиск точек интереса на данном изображении.

и столбцов  $P_{jc} - C \dots P_{jc} + C$ . С уменьшением размеров области ускоряется поиск и уменьшается вероятность ошибки из-за обнаружения неверной точки. Но уменьшать область имеет смысл только с учетом сведений о предельной скорости объектов. Окончательный алгоритм представлен в виде алгоритма 9.3. Результаты его работы показаны на рис. 9.6. Тестовые изображения были



**Рис. 9.5.** Для каждой точки интереса  $(T_x, T_y)$  изображения  $I_1$  в малой прямоугольной области в окрестности  $(T_x, T_y)$  изображения  $I_2$  производится поиск наилучшего совпадения с точкой интереса. Если поиск успешен, то найденная точка  $(H_x, H_y)$  запоминается в качестве конца вектора перемещения.



**Рис. 9.6.** Результаты работы алгоритма 9.3. Слева показано изображение для момента времени  $t_1$ . Справа приведено изображение для момента времени  $t_2$  и условные обозначения результатов анализа движения. Красные квадраты отмечают положение исходных областей, обнаруженных признаковым оператором на левом изображении. Синие квадраты отмечают наилучшие положения этих областей на правом изображении. Видны три связанных множества векторов перемещения (зеленые отрезки), соответствующие трем движущимся объектам. Самый левый объект переместился вниз и немного вправо. Нижний объект сместился вправо и немного вниз. Правый объект переместился влево и немного вверх. (С любезного разрешения Adam T. Clark.) См. цветную вклейку.

получены путем перемещения трех текстурированных объектов на фоне со слабо выраженной текстурой.

Алгоритм 9.3 можно применять для циклической обработки пар кадров для непрерывного прослеживания точек интереса на видеопоследовательности с большим количеством кадров. При этом угловые точки, обнаруженные на кадре  $t + \Delta$ , можно после очередного применения алгоритма запомнить в качестве точек, обнаруженных на очередном кадре, но для вычисления

Алгоритм должен обнаружить на двух входных изображениях пары соответствующих признаков точек и построить по ним векторы перемещения.

**Входные данные:** Два полутоновых изображения  $I_1[r, c]$  и  $I_2[r, c]$  с размерами **MaxRow** строк и **MaxCol** столбцов.

**Выходные данные:**  $V$  — множество векторов перемещения  $\{(T_x, T_y), (H_x, H_y)\}_i$ , где  $(T_x, T_y)$  — начало, а  $(H_x, H_y)$  — конец вектора перемещения.

**procedure** extract\_motion\_field( $I_1, I_2, V$ )

{

“Процедура ищет соответствующие угловые точки и возвращает векторы перемещения  $V$ ”

“ $\tau_2$  — порог значения кросс-корреляции по окрестности точки интереса”

detect\_corner\_points( $I_1, V$ );

**for** каждый вектор  $[(T_x, T_y), (U_x, U_y)]$  из  $V$

  match := best\_match( $I_1, I_2, T_x, T_y, H_x, H_y$ );

**if** (match <  $\tau_2$ ) **then** удалить  $[(T_x, T_y), (U_x, U_y)]$  из  $V$ ;

**else** заменить в  $V$  вектор  $[(T_x, T_y), (U_x, U_y)]$  на  $[(T_x, T_y), (H_x, H_y)]$ ;

}

**real procedure** best\_match( $I_1, I_2, T_x, T_y, H_x, H_y$ );

“ $(H_x, H_y)$  возвращается как центр области в  $I_2$ , которая лучше всего совпадает с областью из” “ $I_1$  с центром  $(T_x, T_y)$ .”

“sh и sw задают прямоугольную область поиска: h и w определяют размеры области.”

{

  “Сбросить флаг, обозначающий успешный поиск”

$H_x := -1; H_y := -1; \text{ best} := 0.0;$

**for** r :=  $T_y - sh$  to  $T_y + sh$

**for** c :=  $T_x - sw$  to  $T_x + sw$

      {

        “Вычислить кросс-корреляцию областей двух изображений (см. Главу 5)”

        match := cross\_correlate( $I_1, I_2, T_x, T_y, r, c, h, w$ );

**if** (match > best) **then**

          {

$H_y := r; \quad H_x := c; \quad \text{best} := \text{match};$

          }

        }

}

**Алгоритм 9.3.** Вычисление разреженного множества векторов перемещения на паре изображений.

кросс-корреляции лучше будет использовать новые, возможно, изменившиеся окрестности. Подобным образом признаковые точки можно проследивать на динамической сцене в условиях постепенного изменения окрестностей этих точек. В общем случае надо предусмотреть действия алгоритма в случае появления новых или исчезновения старых угловых точек из-за перекрытия другими объектами. Эти вопросы будут обсуждаться в разд. 9.4.

---

### Упражнение 9.4

Рассмотрим изображение обычной шахматной доски. (а) Разработайте детектор угловых точек, который будет выделять угловые точки, принадлежащие четырем клеткам, и пропускать точки, принадлежащие только двум клеткам или обнаруженные внутри клеток. (б) Получите несколько изображений шахматной доски, медленно движущейся перед неподвижной камерой. (в) Проверьте свой детектор угловых точек на полученных изображениях и для каждого изображения запишите количество правильно и неправильно найденных точек. (г) Реализуйте и проверьте алгоритм 9.3 на нескольких парах изображений (смещение доски не должно быть слишком большим).

### 9.3.3 Алгоритм MPEG для сжатия потока видеоданных

Алгоритм MPEG предназначен для сжатия последовательностей изображений (видеопотока) с коэффициентом сжатия до 200:1. Этот алгоритм сложен, и мы не будем описывать его в деталях, но рассмотрим сходство способа учета движения на изображениях в алгоритме MPEG и в алгоритме 9.3. Назначение MPEG состоит не в вычислении поля движения, а в уменьшении объема последовательности изображений за счет предсказания содержания части кадров по содержанию других кадров. Для решения задачи сжатия не существенно, насколько корректным описанием движущихся объектов могут служить векторы перемещений, но важно то, что они являются хорошим средством для приближенного описания содержания одной области изображения по содержанию другой. MPEG-упаковщик обрабатывает изображения, разбивая их на прямоугольные области размерами  $16 \times 16$  пикселей (блоки). MPEG-упаковщик пытается представить каждый блок очередного кадра в виде вектора перемещения, позволяющего отыскать наиболее похожий блок на одном из предыдущих кадров. Применение прогноза движения в MPEG на примере обработки нескольких блоков иллюстрируется на рис. 9.7. Комментарии приведены на подписи к рисунку. Признаковые точки не используются. Вместо них в MPEG применяется равномерная сетка блоков. При обработке очередного блока текущего кадра для него отыскивается похожий блок на предыдущем кадре видеопоследовательности. На рис. 9.7 показаны всего несколько из большого количества обрабатываемых блоков. В идеальном случае упаковщик мог бы заменить каждый блок  $B_k$  одним вектором  $[V_x, V_y]_k$ , указывающим на наиболее похожий блок на предыдущем кадре. Если распре-

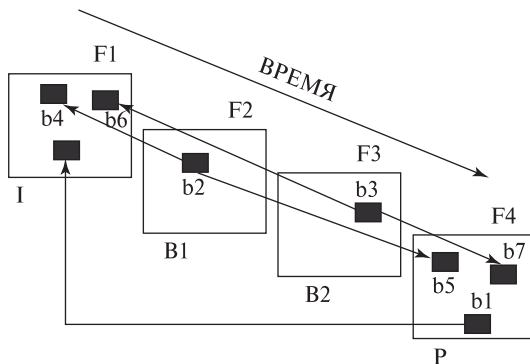


деления интенсивности в этих двух блоках незначительно отличаются, то это различие можно представить небольшим количеством битов и также поместить в сжатый видеопоток.

Изначально векторы перемещения в MPEG предназначались для сжатия видеопотока, а не для анализа движения на изображениях. Современные MPEG-упаковщики способны выполнять сжатие в реальном времени. Сжатый видеопоток уже содержит данные о векторах перемещения. В будущем в упаковщиках MPEG будут предусмотрены параметры, позволяющие извлечь данные о полях движения для использования в приложениях анализа движения.

### Упражнение 9.5

Дана видеопоследовательность из полутоновых кадров размерами  $320 \times 240$  пикселей, 8 бит на пиксел. (а) В каком виде будут представлены промежуточные кадры при сжатии MPEG-подобным упаковщиком? (б) Сколько байт требуется на каждый промежуточный кадр? (в) Чему равен коэффициент сжатия промежуточного кадра относительно исходного изображения?



**Рис. 9.7.** Упрощенная схема использования векторов перемещения для сжатия видеопоследовательности из четырех кадров F1, F2, F3, F4 с помощью алгоритма MPEG. Кадр F1 сжимается как *независимый* (I), или *опорный*, кадр с помощью алгоритма JPEG, рассчитанного на отдельные изображения. F4 является кадром, *предсказанным* (P) по кадру F1 с помощью векторов перемещения. В F4 также хранятся различия содержимого блоков (блоки разностей). Блоки пикселей  $16 \times 16$  (b1) кадра F4 с помощью векторов перемещения можно найти в кадре F1 и для точного восстановления добавить к ним блоки разностей. Промежуточные кадры B1 и B2 целиком вычисляются путем интерполяции с использованием векторов перемещения. Блоки  $16 \times 16$  (b2) вычисляются как среднее блоков (b4) в кадре F1 и (b2) в кадре F4. Промежуточные кадры F2 и F3 можно распаковать только после распаковки предсказанного кадра F4. Наибольший эффект сжатия дают промежуточные кадры, так как каждый блок  $16 \times 16$  представляется всего двумя векторами перемещения. Опорные кадры сжимаются меньше остальных.

### 9.3.4 Вычисление потока изображения\*

Известны методы для вычисления потока изображения во всех точках, а не только в точках интереса. Рассмотрим классический метод, в котором совместно учитываются пространственный и временной градиенты, вычисленные, как минимум, по двум последовательным кадрам. На рис. 9.8 приведен идеализированный пример изображений объекта, движущегося в поле зрения камеры. На рис. 9.8 (а) для момента времени  $t_1$  показан треугольный объект в левом нижнем углу изображения, а на рис. 9.8 (b) показан этот же треугольник, сместившийся вверх, в момент времени  $t_2$ . Этот простой пример поясняет некоторые допущения, которые будут сделаны перед построением математической модели потока изображения.

- Будем полагать, что отражающие свойства и освещенность объекта в течение временного интервала  $[t_1, t_2]$  не меняются.
- Предположим, что на этом временном интервале расстояние от объекта до камеры или источников света изменяется незначительно.
- Допустим, что каждая малая окрестность изображения  $N_{x,y}$  для момента времени  $t_1$  может быть найдена в некотором смещенном положении  $N_{x+\Delta x, y+\Delta y}$  в момент времени  $t_2$ .

В реальности эти допущения точно не выполняются, но в некоторых случаях они позволяют вычислить векторы потока изображения, полезные в практических целях. После обоснования математической модели на простом дискретном случае перейдем к записи уравнения потока изображения для непрерывной функции интенсивности  $f(x, y)$ , зависящей от непрерывных пространственных координат.

3333333333	3333333333
3333333333	3333333333
3333333333	3373333333
3373333333	3397533333
3397533333	3399753333
3399753333	3399975333
3399975333	3333333333
3333333333	3333333333
(a) $t_1$	(b) $t_2$

**Рис. 9.8.** Пример потока изображения. Между моментами времени  $t_1$  и  $t_2$  яркий треугольник смещается на один пиксел вверх. Интенсивность фона равна 3, интенсивность объекта — 9.

### Упражнение 9.6

Рассмотрим рис. 9.8. Обозначим функцию интенсивности  $f(x, y, t)$ . В момент времени  $t_1$  рассмотрим пиксел с координатами  $x = y = 4$  и яркостью 7 из окрестности пикселей 9 7 5. В области  $3 \times 3$  оцените частные производные функции интенсивности  $\partial f/\partial x$  и  $\partial f/\partial y$  по координатам при  $x = y = 4; t = t_1$ . Оцените частную производную функции интенсивности  $\partial f/\partial t$  по времени при  $x = y = 4; t = t_1$ . Как выполнить эти вычисления?

### 9.3.5 Уравнение потока изображения\*

С учетом предположений из разд. 9.3.4 выполним вывод *уравнения потока изображения* и покажем, как с его помощью можно вычислять векторы потока движения. Будем пользоваться непрерывной функцией интенсивности  $f(x, y, t)$ . Разложим ее в ряд Тейлора в малой окрестности произвольной точки  $(x, y, t)$ .

$$f(x + \Delta x, y + \Delta y, t + \Delta t) = f(x, y, t) + \frac{\partial f}{\partial x} \Delta x + \frac{\partial f}{\partial y} \Delta y + \frac{\partial f}{\partial t} \Delta t + h.o.t. \quad (9.1)$$

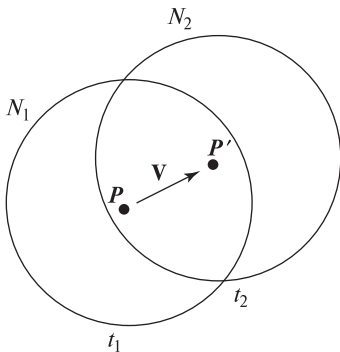
Отметим, что (9.1) является обобщением интуитивно понятного выражения для одной переменной  $f(x + \Delta x) \approx f(x) + f'(x)\Delta x$  на случай нескольких переменных. В малой окрестности точки  $(x, y, t)$  в (9.1) можно пренебречь слагаемыми высокого порядка  $h.o.t.$  и учитывать только линейные члены. Следующий важный шаг вывода поясняется на рис. 9.9. Вектор потока изображения  $\mathbf{V} = [\Delta x, \Delta y]$  — это вектор, для которого надо найти смещение окрестности  $(x, y)$  изображения  $N_1$  в момент времени  $t_1$  в равную по интенсивности окрестность  $(x + \Delta x, y + \Delta y)$  изображения  $N_2$  в момент времени  $t_2$ . Это значит, что

$$f(x + \Delta x, y + \Delta y, t + \Delta t) = f(x, y, t) \quad (9.2)$$

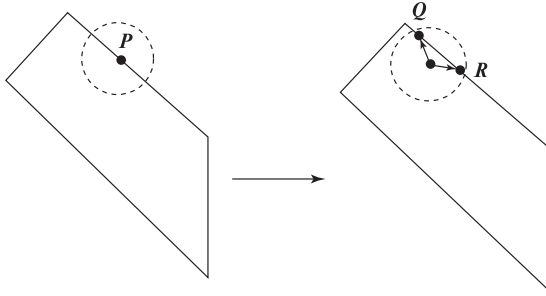
Уравнение потока изображения получим, объединяя (9.1) и (9.2) и игнорируя слагаемые высокого порядка.

$$-\frac{\partial f}{\partial t} \Delta t = \frac{\partial f}{\partial x} \Delta x + \frac{\partial f}{\partial y} \Delta y = \left[ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right] \circ [\Delta x, \Delta y] = \nabla f \circ [\Delta x, \Delta y] \quad (9.3)$$

Решение уравнения потока изображения точно не определяет вектор потока  $\mathbf{V}$ , но налагает на него линейное ограничение. На изображении может быть много окрестностей  $N_2$ , совпадающих по яркости с  $N_1$ . На рис. 9.10 показано, что в случае малой окрестности (*апертуры*) с центром в точке  $(x, y)$  может существовать несколько возможных векторов потока. При рассмотрении малой апертуры с центром в точке  $\mathbf{P}$  может показаться, что эта точка переме-

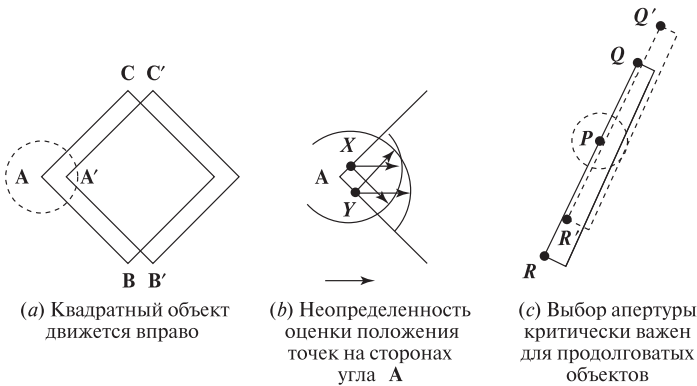
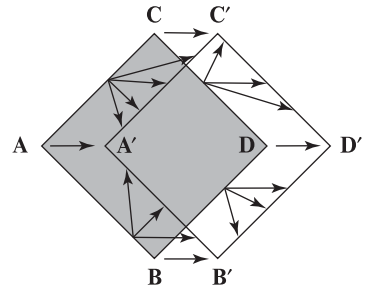


**Рис. 9.9.** При движении в направлении  $\mathbf{v}$  интенсивность окрестности  $N_2$  в момент времени  $t_2$  равна интенсивности окрестности  $N_1$  в момент времени  $t_1$ .



**Рис. 9.10.** Прямолинейный край (перепад интенсивности) между моментами времени  $t_1$  и  $t_2$  сдвигается вправо. Из-за ограниченных размеров рассматриваемой окрестности, или *апертуры*, в качестве нового местоположения точки  $P$  может быть найдена точка  $R$ ,  $Q$  или какая-либо другая точка вдоль края.

**Рис. 9.11.** Квадратный объект движется вправо. На векторы перемещения и их начала на краях в момент времени  $t_1$  накладывается линейное ограничение, согласно которому концы векторов также должны располагаться на краях в момент времени  $t_2$ . Одинаковые ограничения для вершин  $A$ ,  $B$ ,  $C$  и  $D$  позволяют говорить о *сдвиге вправо*. Так как объект считается твердым, то эту интерпретацию можно распространить на все точки вдоль границы объекта.



**Рис. 9.12.** (a) Квадратный объект движется вправо. (b) Рассмотрение точек на разных сторонах квадрата в окрестности вершины  $A$  показывает, как ограничения из двух уравнений потока изображения для соседних точек позволяют уменьшить неоднозначность направления до  $\pi/2$ . (c) Проблема малой апертуры. Продолговатый объект движется в направлении своей длины. Апертура с центром в точке  $P$  позволяет оценить направление движения с точностью до  $\pi$ .

щается в точку  $R$ ,  $Q$  или некоторую другую точку вдоль отрезка  $QR$ . На рис. 9.11 показаны четыре различных края, наблюдаемых в случае квадратного объекта. В общем случае четко выраженных прямолинейных краев объекта может и не быть, но рис. 9.9 в равной степени применим и к криволинейным контурам постоянной яркости (тогда надо края, показанные на этом рисунке, рассматривать как касательные к контуру, пригодные для его локальной аппроксимации).

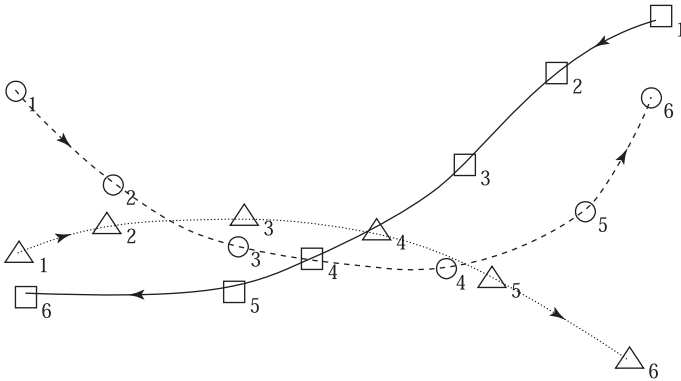
Рис. 9.10 можно интерпретировать следующим образом. Изменения интенсивности, наблюдаемые в точке  $P$ , могут быть записаны как  $-\frac{\partial f}{\partial t} \Delta t$ . Это изменение равно скалярному произведению пространственного градиента  $\nabla f$  и вектора потока  $\mathbf{V}$ . Длина вектора  $|\mathbf{V}|$  может быть равна расстоянию до нового положения перепада яркости вдоль перпендикуляру к нему, или она может оказаться намного больше, если направление вектора потока сильно отличается от направления пространственного градиента. Примером этого может служить веревка, которую очень быстро тянут в вертикальном направлении, и при этом она слегка вибрирует в горизонтальном направлении. Тогда края веревки на изображении будут смещаться незначительно.

### 9.3.6 Распространение ограничений при вычислении потока изображения\*

Уравнение потока изображения дает ограничение, которое можно применять для каждого пиксела изображения. Если предположить связность изображения, то у соседних пикселов должны быть похожие векторы потока. На рис. 9.12 показано, как наложение ограничений на соседние пикселы изображения позволяет уменьшить неоднозначность оценки направления движения. На рис. 9.12 (b) показаны точки в окрестности угла  $A$  движущегося квадрата. Точки  $X$  и  $Y$  взяты на разных сторонах квадрата. В точке  $X$  уравнение потока изображения ограничивает направление перемещения  $\theta_x$  интервалом  $(5\pi/4, \pi/4)$ . В точке  $Y$  направление перемещения  $\theta_y$  ограничено интервалом  $(-\pi/4, 3\pi/4)$ . Так как предполагается, что точки  $X$  и  $Y$  принадлежат одному твердому телу, то на векторы потока в точках  $X$  и  $Y$  эти два ограничения накладываются совместно. Поэтому направление перемещения точек  $X$  и  $Y$  будет ограничено интервалом  $(-\pi/4, \pi/4)$ .

Рис. 9.11 и 9.12 подчеркивают два важных момента. Во-первых, малыми апертурами можно пользоваться для вычисления потока изображения только в признаковых точках (например, угловых). Во-вторых, ограничения на векторы потока в угловых точках можно распространять (экстраполировать) вдоль краев. Однако, как показано на рис. 9.12 (c), для получения интерпретации (т. е. решения, куда смещается окрестность) для неугловых краевых точек (например,  $P$ ) может потребоваться несколько итераций.

В некоторых экспериментах по вычислению потока изображения можно пользоваться изображениями, созданными с помощью генератора случайных чисел. Наш опыт показал, что такие изображения проще обрабатывать, чем



**Рис. 9.13.** Траектории трех объектов  $\bigcirc$ ,  $\triangle$ ,  $\square$ . Для 6 моментов времени отмечено местоположение каждого объекта. Объекты  $\bigcirc$  и  $\triangle$  двигаются слева направо, а  $\square$  — справа налево.

изображения с хорошо выраженной структурой — на случайно сгенерированных изображениях меньше одинаковых окрестностей.

Двумерные релаксационные методы будут рассмотрены в гл. 11. Вычисление потока изображения с использованием дифференциальных уравнений описано в статье [Horn, Schunck, 1981].

## 9.4 ВЫЧИСЛЕНИЕ ТРАЕКТОРИЙ ДВИЖУЩИХСЯ ТОЧЕК

Выше были рассмотрены способы поиска признаков точек объекта на кадре для момента времени  $t_1$  и для обнаружения этих же точек на следующем кадре для момента времени  $t_2$ . Если интенсивность в окрестности каждой точки описывается уникальной текстурой, то для прослеживания каждой точки в течение длительного промежутка времени можно пользоваться методом вычисления нормированной кросс-корреляции. Прослеживание объекта на последовательности изображений может быть основано на специфических особенностях задачи, например, слежение за оранжевым теннисным мячом на съемке игры в теннис или за лицом оператора, смотрящего на экран компьютера.

В данном разделе будет рассматриваться более общий случай, когда движущиеся точки не имеют уникальной текстурной или цветовой информации. Для накопления точек и построения траектории требуется использовать свойства самого движения. На рис. 9.13 показаны гладкие траектории трех объектов, построенные по 6-ти точкам, соответствующим различным моментам времени.

Перед рассмотрением абстрактного общего случая в качестве примера приведем три частные задачи. Первая задача: из коробки на землю высыплют много теннисных мячей и по последовательности изображений надо вычислить траекторию каждого мяча. Вторая задача относится к изучению течения

жидкости через некоторую полость. В жидкость внесены флуоресцирующие частицы. Они наблюдаются в течение некоторого времени. Предполагается, что все частицы на изображении выглядят одинаковыми. Третья задача: определить траектории людей, проходящих по аллее. Из-за особенностей одежды изображения некоторых людей будут уникальными, но также очень вероятно, что некоторые люди будут выглядеть примерно одинаково.

Будем пользоваться следующими общими предположениями, верными для движения физических объектов в трехмерном пространстве.

1. Местоположение движущегося физического объекта с течением времени изменяется непрерывно.
2. Скорость движущегося физического объекта с течением времени изменяется непрерывно и по величине, и по направлению.
3. В каждый момент времени объект может находиться только в одной области пространства.
4. Два различных объекта в один момент времени не могут находиться в одной области пространства.

Первые три допущения выполняются и для двумерных проекций трехмерного пространства; при непрерывном движении объектов в трехмерном пространстве на изображениях видны непрерывные траектории их образов. Четвертое допущение при проектировании может быть нарушено, так как объекты могут закрывать друг друга перед камерой. Это создает проблемы при наблюдении с помощью одной камеры с фиксированным местоположением. Эксперименты с участием людей показали, что люди способны распознавать объекты и анализировать их движение по набору кадров, содержащих только отдельные точки движущихся объектов. В известном эксперименте, рассмотренном в [Johansson, 1976] в различных точках человеческого тела были закреплены источники света. Наблюдатели, видевшие в темноте источники света на неподвижном человеке (которого не было видно), не могли понять, что это человек. Однако когда он начинал двигаться, то наблюдатели легко распознавали, что это именно человек.

Теперь перейдем к рассмотрению алгоритмов, представленных в работе [Sethi, Jain, 1987], где были использованы перечисленные выше предположения для вычисления множества гладких траекторий, проходящих через точки, обнаруженные на последовательности кадров. Сначала приведем математическое определение гладкого куска траектории. Затем введем определение гладкого множества из  $m$  кусков траекторий как множества таких кусков гладких однозначных кривых, для которых максимальна сумма  $m$  значений «гладкости». В заключение рассмотрим *жадный обменный алгоритм*, который циклически наращивает  $m$  кусков с момента времени  $t_1$  до  $t_n$ , подбирая для каждого момента времени наиболее подходящие точки для каждого куска.

**73 Определение.** Если объект  $i$  наблюдается в моменты времени  $t = 1, 2, \dots, n$  в точках изображений  $p_{i,1}, p_{i,2}, \dots, p_{i,t}, \dots, p_{i,n}$ , то после-

довательность этих точек  $T_i = (p_{i,1}, p_{i,2}, \dots, p_{i,t}, \dots, p_{i,n})$  называется **траекторией** объекта  $i$ .

Для любых двух точек траектории можно определить вектор разности

$$V_{i,t} = p_{i,t+1} - p_{i,t} \quad (9.4)$$

На основе разности векторов, входящих и выходящих из точки траектории  $p_{i,t}$ , введем новую характеристику траектории — «гладкость в точке». Гладкость направления определяется скалярным произведением этих векторов, а гладкость скорости определяется на основе сравнения геометрического и арифметического среднего длин этих векторов.

$$S_{i,t} = w \left( \frac{V_{i,t-1} \circ V_{i,t}}{|V_{i,t-1}| |V_{i,t}|} \right) + (1-w) \left( \frac{2\sqrt{|V_{i,t-1}| |V_{i,t}|}}{|V_{i,t-1}| + |V_{i,t}|} \right) \quad (9.5)$$

Весовой коэффициент  $w$  выбран в диапазоне  $0 \leq w \leq 1$ , так что  $0 \leq S_{i,t} \leq 1$  (см. упражнения). Для прямолинейной траектории с равномерно распределенными точками все разностные векторы равны и  $S_{i,t} = 1.0$  — это оптимальное значение гладкости в точке. Изменение направления или скорости уменьшает значение  $S_{i,t}$ . Будем считать, что на каждом из  $n$  кадров найдены  $m$  уникальных точек, хотя далее мы сможем ослабить это требование. Точки на первом кадре обозначим  $i = 1, 2, \dots, m$ . Задача заключается в том, что надо построить  $m$  траекторий  $T_i$  с максимальным суммарным значением гладкости. Суммарная гладкость  $T_s$  — это сумма значений гладкости во всех внутренних точках всех  $m$  кусков траекторий (9.6).

$$\text{Суммарная гладкость } T_s = \sum_{i=1}^m \sum_{t=2}^{n-1} S_{i,t} \quad (9.6)$$

### Упражнение 9.7

Пусть  $w = 0.5$ , так что направление и скорость в выражении (9.5) учитываются с одинаковым весом. (а) Покажите, что точечная гладкость в каждой вершине правильного шестиугольника с единичной стороной равна 0.75. (б) Чему равна точечная гладкость в вершинах квадрата?

### Упражнение 9.8

(а) С помощью неравенства Коши—Шварца (гл. 5) покажите, что в выражении  $S_{i,t}$  значение множителя  $\frac{V_{i,t-1} \circ V_{i,t}}{|V_{i,t-1}| |V_{i,t}|}$  лежит между 0 и 1. (б) Покажите, что для двух положительных чисел  $x$  и  $y$  их геометрическое среднее  $\sqrt{xy}$  никогда не превышает их арифметического среднего  $(x+y)/2$ . Используйте этот факт, чтобы доказать, что множитель  $\frac{2\sqrt{|V_{i,t-1}| |V_{i,t}|}}{|V_{i,t-1}| + |V_{i,t}|}$  лежит между 0 и 1. (в) Покажите теперь, что в (9.5)  $0 \leq S_{i,t} \leq 1$  за счет того, что  $0 \leq w \leq 1$ .



Алгоритм должен вычислить гладкие траектории на основе входных массивов двумерных точек, найденных в последовательные моменты времени.

$\mathbf{P}[i, t]$  содержит  $i = 1, 2, \dots, m$  двумерных координат точек, найденных на кадрах для моментов времени  $t = 1, 2, \dots, n$ ;

$\mathbf{T}[i, t]$  является выходным массивом траекторий ( $m$  строк,  $n$  столбцов);

$\mathbf{T}[i, t] = k$  означает, что на кадре  $t$  объект  $i$  находился в  $k$ -й точке.

1. **Инициализация:** создание  $m$  полных кусков траекторий путем соединения ближайших соседей.

(а) Первый кадр: для всех  $i$ , задать метки объектов  $\mathbf{T}[i, 1] = i$ ;

(б) Остальные кадры: для  $t = 2, 3, \dots, n$ , присвоить  $\mathbf{T}[i, t] = k$ , где точка  $\mathbf{P}[k, t]$  является еще не выбранной точкой, ближайшей к точке  $\mathbf{P}[\mathbf{T}[i, t-1], t-1]$ .

2. **Обменный цикл:** for  $t = 2$  to  $n - 1$

(а) Для всех пар  $(j, k)$  при  $j \neq k$ , вычислить приращение гладкости при обмене выбора  $\mathbf{T}[j, t]$  на  $\mathbf{T}[k, t]$ ;

(б) Выполнить обмен точек для достижения максимального значения гладкости. Если суммарная гладкость при обмене не увеличивается, выполнять его не надо.

(с) Установить флаг обмена, если он был выполнен.

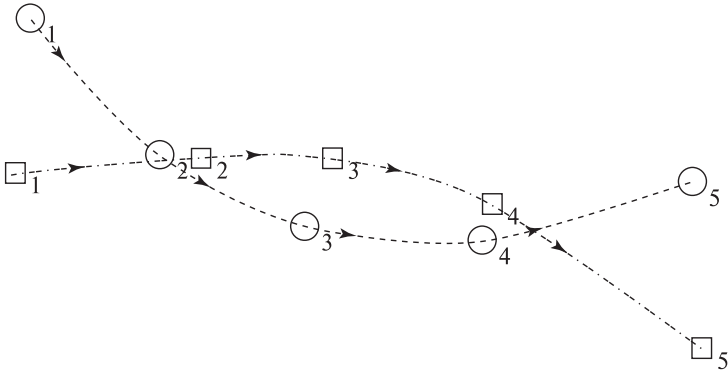
3. **Проверка условия завершения:** если в цикле на предыдущем шаге был выполнен обмен, то сбросить флаг обмена и повторить обменный цикл.

**Алгоритм 9.4.** Жадный обменный алгоритм.

## Упражнение 9.9

Вычислите суммарную гладкость 4-точечной траектории вдоль четырех сторон восьмиугольника со сторонами  $s$  и 4-точечной траектории вдоль сторон квадрата со сторонами  $s$ .

Алгоритм 9.4 строит множество из  $m$  траекторий на  $n$  кадрах. Достижение минимально возможного значения  $T_s$  не гарантируется, но в экспериментах этот алгоритм продемонстрировал неплохие результаты. Сначала рассмотрим некоторые действия алгоритма на качественном уровне на примере данных с рис. 9.14. В табл. 9.1 приведены данные о гладкости некоторых рассматриваемых кусков траекторий. На первом кадре можно произвольно присвоить метки точкам; например, объект  $\square_1 \equiv 1 = T[1, 1]$  и объект  $\bigcirc_1 \equiv 2 = T[2, 1]$ . Затем

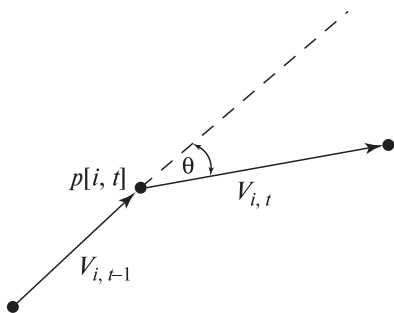


**Рис. 9.14.** Траектории двух объектов,  $\bigcirc$  и  $\square$ , и векторы потока изображения для первых четырех точек. Алгоритм слежения мог бы ошибочно принять  $\square_2$  за вероятный последующий элемент после  $\bigcirc_1$ , а  $\bigcirc_5$  — за вероятную конечную точку последовательности  $\square_1, \square_2, \square_3, \square_4$ .

можно продолжить траектории к ближайшей точке на последующих кадрах: ближайшая точка  $T[1, 2] = \bigcirc_2$  и остается  $T[2, 2] = \square_2$ . Здесь мы сделали ошибку, перепутав настоящие траектории. Вычислим суммарную гладкость для этих двух кусков после того, как выберем ближайшего соседа для момента времени  $t = 3$ . Из первых двух строк табл. 9.1 видно, что суммарная гладкость этих двух кусков равна  $0.97 + 0.98 = 1.95$ . Если поменять местами присвоения  $T[1, 2] = \bigcirc_2$  и  $T[2, 2] = \square_2$ , то будет достигнуто лучшее значение гладкости  $0.99 + 0.99 = 1.98$ . После этого обмена две траектории в момент времени  $t = 2$  будут иметь вид  $(\square_1, \square_2)$  и  $(\bigcirc_1, \bigcirc_2)$ . Выбор очередной точки как ближай-

**Таблица 9.1.** Гладкость кусков траекторий по рис. 9.14

$t = 1$	$t = 2$	$t = 3$	$t = 4$	$t = 5$	гладкость
$\bigcirc_1(112\ 262)$	$\square_2(206\ 185)$	$\bigcirc_3(250\ 137)$			0.97
$\square_1(106\ 175)$	$\bigcirc_2(180\ 188)$	$\square_3(280\ 185)$			0.98
$\bigcirc_1(112\ 262)$	$\bigcirc_2(180\ 188)$	$\bigcirc_3(250\ 137)$			0.99
$\square_1(106\ 175)$	$\square_2(206\ 185)$	$\square_3(280\ 185)$			0.99
$\bigcirc_1(112\ 262)$	$\bigcirc_2(180\ 188)$	$\bigcirc_3(250\ 137)$	$\bigcirc_4(360\ 137)$		1.89
$\square_1(106\ 175)$	$\square_2(206\ 185)$	$\square_3(280\ 185)$	$\square_4(365\ 156)$		1.96
$\bigcirc_1(112\ 262)$	$\bigcirc_2(180\ 188)$	$\bigcirc_3(250\ 137)$	$\bigcirc_4(360\ 137)$	$\square_5(482\ 80)$	2.84
$\square_1(106\ 175)$	$\square_2(206\ 185)$	$\square_3(280\ 185)$	$\square_4(365\ 156)$	$\bigcirc_5(478\ 170)$	2.91
$\bigcirc_1(112\ 262)$	$\bigcirc_2(180\ 188)$	$\bigcirc_3(250\ 137)$	$\bigcirc_4(360\ 137)$	$\bigcirc_5(478\ 170)$	2.89
$\square_1(106\ 175)$	$\square_2(206\ 185)$	$\square_3(280\ 185)$	$\square_4(365\ 156)$	$\square_5(482\ 80)$	2.94



**Рис. 9.15.** Входящий и выходящий векторы для точки траектории  $p[i, t]$ .

шего соседа даст наилучшее значение гладкости при  $t=3$  и  $t=4$ , поэтому выполнять обмен значений не придется. Однако при  $t=5$  выбор ближайшего соседа даст траектории  $(\square_1, \square_2, \square_3, \square_4, \bigcirc_5)$  и  $(\bigcirc_1, \bigcirc_2, \bigcirc_3, \bigcirc_4, \square_5)$ . Вычисление суммарной гладкости после обмена двух последних значений улучшит значение с  $2.84 + 2.91 = 5.75$  до  $2.89 + 2.94 = 5.83$ , так что окончательный результат будет корректным и соответствовать рис. 9.14.

Перед применением критерия максимальной гладкости алгоритм 9.4 инициализирует  $m$  полных кусков траекторий из  $n$  точек каждая. Затем алгоритм пытается увеличить гладкость траекторий, выполняя обмен точек между парами кусков посредством некоторого количества обменных циклов. Если для некоторого момента времени  $t$  улучшение достигнуто, то обменный цикл повторяется еще раз. В общем, на каждом кадре  $t$  должны быть рассмотрены  $\binom{m}{2}$  возможных изменений. Алгоритм выполняет как минимум  $(n-2)\binom{m}{2}$  операций; любой дополнительный обменный цикл увеличивает их количество. Суммарная гладкость не может превысить  $1.0m(t-2)$ , так как количество возможных увеличений ограничено и алгоритм обязательно завершает свою работу.

На кадре  $t=1$  точки выбираются произвольно, поэтому всего возможны  $m^{n-1}$  кусков траекторий — это слишком большое количество для вычислений. Для каждого момента времени алгоритм 9.4 рассматривает изменения максимум в одной паре точек, поэтому достижение глобального максимума суммарной гладкости траекторий не гарантируется. Алгоритм можно модифицировать, чтобы он выполнял инициализацию только по одному или трем кадрам, а затем продолжал обработку и поиск признаков точек по мере ввода очередных кадров. Если известны все точки на всех кадрах, то алгоритм можно улучшить за счет введения в обменном цикле прямой и обратной обработки. Также алгоритм можно усовершенствовать для обработки ситуаций появления и исчезновения точек из-за взаимного перекрытия движущихся объектов. Для представления точек на кадрах, содержащих менее чем  $m$  точек, можно ввести *фиктивные* (фантомные) точки.

### Упражнение 9.10

Ниже в таблице приведены координаты наборов из трех точек, найденных на 6 кадрах видеопоследовательности для данных, показанных на рис. 9.13. С помощью жадного обменного алгоритма найдите гладкое множество из трех траекторий.

$t = 1$	$t = 2$	$t = 3$	$t = 4$	$t = 5$	$t = 6$
(483 270)	(155 152)	(237 137)	(292 128)	(383 117)	(475 220)
(107 225)	(420 237)	(242 156)	(358 125)	(437 156)	(108 108)
(110 133)	(160 175)	(370 180)	(310 145)	(234 112)	(462 75)

### Упражнение 9.11

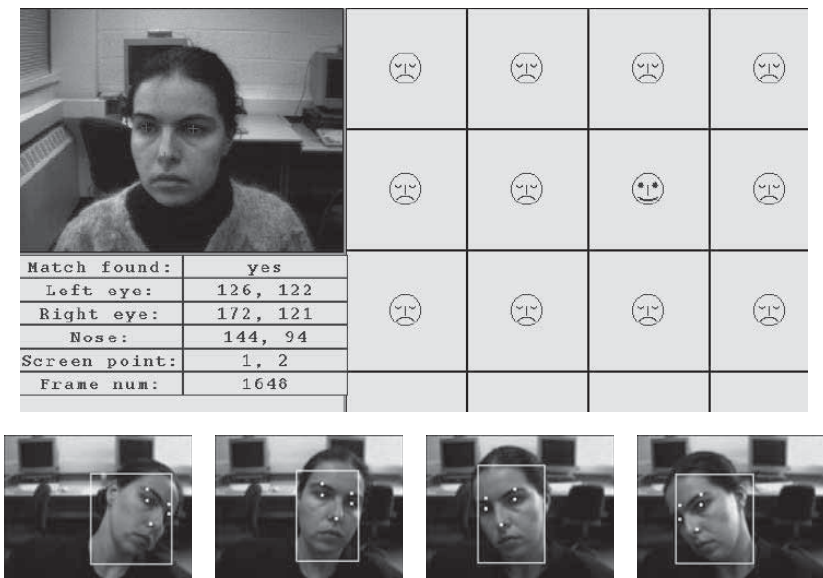
Сможет ли жадный обменный алгоритм успешно сформировать траектории по точкам, найденным на видеопоследовательностях в перечисленных далее случаях? (а) Съемка вращающейся карусели с деревянными лошадьми, которые двигаются вверх и вниз. (б) Сделанная с тротуара съемка улицы: две машины движутся в противоположных направлениях со скоростью 60 км/ч прямо перед камерой. (в) Высокоскоростная съемка столкновения двух бильярдных шаров. Движущийся белый шар попадает в неподвижный красный шар. После столкновения белый шар останавливается, а его импульс полностью передается красному шару.

## 9.4.1 Слежение с учетом априорных знаний из предметной области

Мощные возможности использования общих ограничений на гладкость траектории для организации слежения были продемонстрированы на примере алгоритма 9.4. Во многих прикладных задачах надежность и скорость слежения могут быть увеличены за счет дополнительной информации. Если для каждой из  $m$  точек определены некоторые характерные признаки, то к вычислению гладкости можно добавить сравнение и согласование признаков. Более того, на основе частичной траектории до текущего момента времени  $t$  можно предсказать местоположение следующей точки траектории на кадре для момента времени  $t + 1$ . Этот прием позволяет существенно уменьшить время корреляционного поиска очередной точки траектории. Алгоритмы на основе подобных методов описаны в ряде современных исследовательских работ. В статье [Maes и др., 1996] представлена система для слежения за движениями человека посредством вычисления траекторий его рук, ног и головы. Эти части тела обнаруживались на силуэте движущегося человека как выступы с высокой кривизной. В работе [Bakic, Stockman, 1999] слежение за лицом оператора рабочей станции, его глазами и носом выполнялось для бесконтактного управления курсором мыши. На рис. 9.16 приведен пример экрана этой системы. На нем отмечены характерные признаки, найденные на

текущем кадре, и вычисленное положение курсора в массиве  $8 \times 8$  команд меню. Команда, выбранная пользователем, обозначена в третьем столбце второй строки в виде пиктограммы с улыбающимся лицом. В данной системе за счет интегрального использования знаний предметной области обеспечивается обработка 15 и более кадров в секунду. Для поиска лица оператора на изображении используются сведения о цвете лица, а для обнаружения глаз и носа — сведения о структуре образа лица. Траектории глаз и носа используются для прогнозирования положения соответствующих признаков на следующем кадре: если их удастся обнаружить в предсказанных областях, то глобальное обнаружение лица не выполняется.

Фильм «Титаник» является ярким примером совместного использования технологий компьютерной графики и компьютерного зрения для комбинирования реальных и искусственных образов. В процессе съемки этого фильма была выполнена съемка модели корабля, а затем в нее были добавлены модели людей, движущихся по палубе корабля. Для имитации движения женщины в платье, типичном для начала XX века, использовались съемки реальной актрисы в подобном платье. К платью было прикреплено большое количество маленьких источников света, так что на последовательности изображений можно было легко найти признаковые точки. Затем траектории этих точек были использованы для согласования движения моделей людей и моделей



**Рис. 9.16.** Слежение за глазами и носом оператора рабочей станции для бесконтактного перемещения курсора мыши. (Сверху) положение лица определяет выбор команды в меню. (Внизу) последовательность изображений лица, на которой отмечено прослеживание глаз и носа. (С любезного разрешения Vera Bakic.)

одежды. Эти модели были добавлены в различные места изображений модели корабля. На каждый кадр этого блокбастера было затрачено много минут компьютерного и операторского времени. Не все съемочные операции требовалось полностью автоматизировать.

## 9.5 ОБНАРУЖЕНИЕ ЗНАЧИТЕЛЬНЫХ ИЗМЕНЕНИЙ УСЛОВИЙ ВИДЕОСЪЕМКИ

Видеозаписи могут содержать большое количество различных сюжетов и сцен. Например, записи телевизионных программ содержат выпуски телевизионных новостей, документальные и художественные фильмы и др. Для создания цифровых библиотек видеоданных очень важно уметь сегментировать и хранить фрагменты видеопоследовательностей так, чтобы можно было обеспечить к ним произвольный доступ. В данном разделе рассматриваются некоторые важные понятия и методы, относящиеся к разбору и анализу видеопоследовательностей. Далее определены несколько типов изменения характера видеосъемки или просто некоторой последовательности изображений.

- **Смена сцены** — это изменение среды, в которой производится видеосъемка. Например, переход от съемки сцены в ресторане к съемке сцены на улице. При этом обычно происходит существенное изменение фона. Часто изменения сцены занимают от 10 до 50 кадров и оформляются с использованием одного из эффектов камеры (см. ниже).
- **Смена монтажного кадра** — это значительное изменение параметров поля зрения при съемке одной и той же сцены. Часто это обеспечивается за счет переключения камер. Например, при съемке сцены в ресторане камера 1 снимает говорящего актера А, а затем берутся кадры с камеры 2, которая снимает ответ собеседника — актера Б, сидящего за тем же столом напротив.
- **Панорамирование камеры** применяется для охвата горизонтального вида сцены. Если камера поворачивается справа налево, то кажется, что объекты появляются с левой стороны кадра и движутся через изображение направо, и затем уходят за правую границу кадра. Если на последовательных кадрах панорамной съемки статической сцены вычислить поле движения, то векторы перемещения тоже будут направлены слева направо.
- **Масштабирование** выполняется путем изменения фокусного расстояния в течение некоторого времени. Это нужно для увеличения или уменьшения (и захвата близлежащих участков фона) изображения некоторой части сцены.
- **Эффекты камеры** (плавное исчезновение/появление, наплыв, вытеснение одного кадра другим) используются для организации перехода от одного источника изображений к другому.

*Плавное исчезновение* (fade out) — это непрерывный переход от одного видеисточника к пустым (черным или белым) кадрам, а постепенное появление — это переход от пустых (черных или белых) кадров к некоторому видеисточнику. Переход от видеисточника А к видеисточнику В можно выполнить, последовательно выполняя исчезновение изображений от А и появление изображений от В. *Эффект наплыва* (dissolve) заключается в том, что в течение нескольких кадров пиксели изображений видеисточника А заменяются на пиксели изображений В. В одном из вариантов наплыва пиксели изображений от видеисточника А на кадрах  $t = 0, 1, \dots, T$  умножаются на коэффициент  $(1 - t/T)$ , а пиксели изображений от В — на коэффициент  $t/T$ . *Эффект вытеснения* (wipe) представляет собой такой переход от изображений источника А к изображениям В, что изменяется размер областей кадра, в которых видны изображения от А и В. Эти области могут иметь различную форму. Например, возможен эффект вытеснения наподобие дворников на ветровом стекле, когда с одной стороны щетки выводится изображения от А, а по другую — от источника В. Вытеснение может выполняться в любом направлении с использованием вертикальной, горизонтальной, диагональной и любой другой границы между двумя областями. Или, например, изображения от источника В могут появляться в маленькой круговой области, которая затем постепенно увеличивается до размеров кадра.

### Упражнение 9.12

На псевдокоде запишите алгоритм для перехода от видеисточника А к В методом вытеснения. Источник А дает последовательность изображений  $A_t[r, c]$ , а источник В — последовательность  $B_t[r, c]$ . (а) Предположите, что вытеснение происходит во временном интервале  $(t_1, t_2)$ . В качестве границы используется диагональная прямая с угловым коэффициентом 1, которая в момент времени  $t_1$  проходит через пиксел  $[0, 0]$ , а в момент времени  $t_2$  — через пиксел  $[M - 1, N - 1]$ . (б) Предположите, что вытеснение производится с помощью увеличивающейся круговой области с центром в центре изображения. В момент времени  $t_1$  радиус окружности равен 0, а в момент времени  $t_2$  радиус становится достаточно большим, так, чтобы весь кадр целиком попал внутрь окружности.

### 9.5.1 Сегментация видеопоследовательностей

Сегментация видеопоследовательностей — это разновидность задач обработки видеоданных, цель которых состоит в разбиении длинной последовательности кадров видеосъемки на фрагменты, представляющие отдельные кинематографические планы или сцены. Допустим, есть 30-минутная запись телевизионных новостей. В ней будет несколько фрагментов длительностью 10–15 секунд, когда камера снимает план с ведущим новостей, сидящим за столом в студии. Фон в этих фрагментах постоянен, но возможно изменение масштаба сцены.

После подобного фрагмента обычно происходит переход к источнику видеоданных, иллюстрирующему другое событие, например, наводнение, спортивный матч, демонстрацию или правительственную пресс-конференцию. Часто в течение сообщения о событии демонстрируется несколько различных планов с переходами между ними. Наличие этих переходов может стать основой для процедуры сегментации видеоданных. Переходы можно обнаружить автоматически как значительные изменения характерных признаков на изображениях в течение некоторого промежутка времени.

Очевидным способом количественной оценки различий между двумя кадрами видеопоследовательности  $I_t$  и  $I_{t+\Delta}$  является вычисление средней разности между соответствующими пикселями пары изображений (9.7). Временной интервал  $\Delta$  может занимать один или несколько кадров, в зависимости от обнаруживаемого эффекта камеры.

$$d_{pixel}(I_t, I_{t+\Delta}) = \frac{\sum_{r=0}^{MaxRow-1} \sum_{c=0}^{MaxCol-1} |I_t[r, c] - I_{t+\Delta}[r, c]|}{MaxRow \times MaxCol} \quad (9.7)$$

Выражение (9.7) может дать качественно неверные результаты, так как разность получается большой даже при небольшом панорамном сдвиге камеры или при движении некоторого объекта в рамках в целом неподвижной сцены. Более устойчивый вариант выражения для оценки различия кадров был предложен в статье [Kasturi, Jain, 1999]. Они предложили разбить изображение на большие блоки и проверить, совпадают ли большинство из этих блоков на обоих изображениях. Для вычисления признака наличия в соответствующих блоках существенного изменения яркости использовалось отношение (9.8), в которое входят статистические характеристики блока — математическое ожидание и дисперсия интенсивности. Пусть на изображении  $I_1$  математическое ожидание интенсивности в блоке  $B_1$  равно  $u_1$ , а дисперсия —  $v_1$ ; а на изображении  $I_2$  математическое ожидание и дисперсия интенсивности в блоке  $B_2$  равны, соответственно,  $u_2$  и  $v_2$ . Различие блоков определяется через отношение этих статистических характеристик (9.8). Если некоторое «достаточное» количество блоков имеют нулевое различие, то считается, что два изображения принадлежат одному плану. Очевидно, выражение (9.8) будет более стабильно, чем (9.7), когда изображения имеют ярко выраженную текстуру. Но (9.8) также не стабильно по отношению к межкадровым изменениям, вызванным малыми смещениями камеры.

$$r = \frac{\left[ \frac{v_1 + v_2}{2} + \left( \frac{u_1 - u_2}{2} \right)^2 \right]^2}{v_1 v_2} \quad (9.8)$$

$$d_{block}(B_1, B_2) = \begin{cases} 1 & \text{если } r > \tau_r \\ 0 & \text{если } r \leq \tau_r \end{cases}$$

$$d(I_1, I_2) = \sum_{B_{1i} \in I_1; B_{2i} \in I_2} d_{block}(B_{1i}, B_{2i})$$



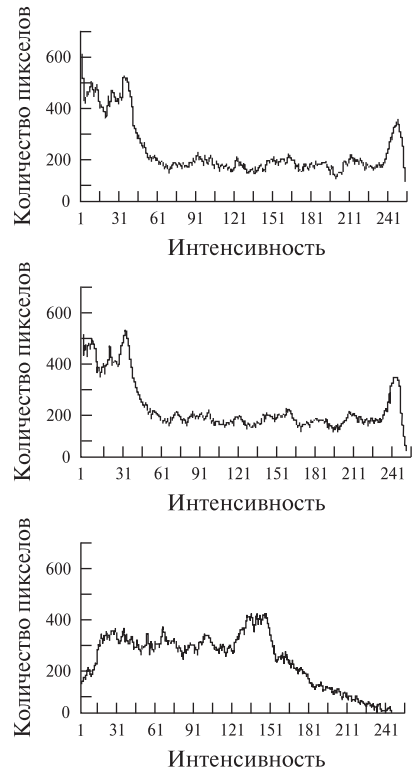
Различие между двумя изображениями можно оценить путем вычисления разности гистограмм (см. вычисление  $d_{hist}(I, Q)$  в гл. 8). Применительно к нашей задаче, в  $d_{hist}(I, Q)$  обозначению  $I$  соответствует кадр  $I_1$ , а обозначению  $Q$  — кадр  $I_2$ . Для вычислений достаточно использовать гистограммы с 64 разрядами. При обработке цветных изображений значение в диапазоне  $[0, 63]$  можно получить, объединяя в одно 6-разрядное число по два старших бита от красной, зеленой и синей цветовых составляющих пиксела. Сравнение гистограмм может быть выполнено быстрее, чем вычисления предыдущими методами. При сравнении гистограмм обычно лучше учитываются глобальные характерные признаки сцены. Но, поскольку в этом методе не предполагается вычисление и сравнение каких-либо пространственных характеристик изображений, то могут быть получены совершенно неверные результаты для пары изображений с похожими гистограммами, но разными пространственными распределениями значений яркостей (т. е. для изображений разных сцен).



**Рис. 9.17.** Четыре кадра из документальной видеосъемки. Верхние и нижние пары кадров разделены паузой камеры. (С любезного разрешения Springer-Verlag.)

На рис. 9.17 показаны четыре кадра, взятые из одной и той же документальной видеосъемки. Верхняя и нижняя пары кадров разделены паузой камеры. На рис. 9.18 приведены гистограммы, построенные по трем первым кадрам с рис. 9.17. Верхние две гистограммы похожи: это значит, что два кадра, по которым они построены, вероятнее всего относятся к одной сцене. Нижняя

гистограмма значительно отличается от двух первых, значит, третий кадр с рис. 9.17 относится к другой сцене.



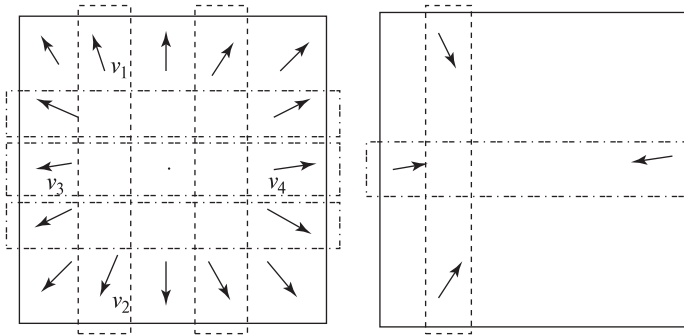
**Рис. 9.18.** Гистограммы первых трех кадров, показанных на рис. 9.17. Верхние две гистограммы похожи, так же как похожи и кадры, по которым они построены. Нижняя гистограмма существенно отличается от двух верхних. Это означает, что кадр, по которому она построена, отличается от двух первых кадров. (С любезного разрешения Springer-Verlag.)

### 9.5.2 Игнорирование некоторых эффектов камеры

При сегментации видеопоследовательностей часто не надо разбивать их в тех местах, в которых значительное различие смежных кадров вызвано исключительно эффектами камеры, например, панорамированием или масштабированием. Переходы, обнаруженные одним из рассмотренных выше способов, можно дополнительно проанализировать с помощью несложного анализа движения и определить, не являются ли эти переходы эффектами камеры, которые следует игнорировать.

Для обнаружения панорамирования можно вычислить векторы движения и проверить, нельзя ли сгруппировать эти векторы по некоторому преобладающему направлению и величине. Эту процедуру можно построить в форме обработки выходного массива  $V$ , возвращаемого алгоритмом 9.3.

Для обнаружения масштабирования можно проверить направление векторов движения в периферийных областях поля движения. Признаком увели-



**Рис. 9.19.** Эвристические правила для обнаружения масштабирования камеры путем сравнения векторов движения в периферийных областях поля движения. Разность вертикальных компонент векторов с противоположных краев поля движения оказывается больше вертикальной компоненты каждого вектора:  $|v_{1r} - v_{2r}| > \max\{|v_{1r}|, |v_{2r}|\}$ . Аналогично, для горизонтальных компонент противоположно направленных векторов  $|v_{3c} - v_{4c}| > \max\{|v_{3c}|, |v_{4c}|\}$ . Эти соотношения верны и для увеличения (слева), и для уменьшения (справа) масштаба.

чения или уменьшения масштаба является то, что сумма векторов движения в периферийных областях примерно равна 0. Использование векторов из периферийных областей позволяет выполнить обработку в случаях, когда фокус схода (или расхождения) смещен относительно центра поля движения.

Предположим, что векторы движения были вычислены с помощью MPEG-подобного алгоритма блочного сопоставления. Векторы движения для блоков вдоль верхней и нижней области поля движения определяются содержимым кадров  $I_1$  и  $I_2$ . Разность вертикальных компонент векторов движения в соответствующих положениях на противоположных сторонах изображения будет больше вертикальных компонент каждого из этих векторов (см. рис. 9.19). Аналогичное соотношение верно и для горизонтальных компонент векторов движения, расположенных на противоположных областях поля движения вдоль горизонтального направления. Эти эвристические правила позволяют достаточно надежно обнаруживать увеличение или уменьшение масштаба. Следует учитывать, что качество поля движения, вычисленного алгоритмом блочного сопоставления, ухудшается при увеличении скорости масштабирования.

### Упражнение 9.13

Получите два последовательных кадра видеосъемки одной и той же сцены. (а) Вычислите среднюю пиксельную разность по формуле (9.7). (б) Разделите изображение на  $2 \times 2 = 4$  блока и вычислите сумму разностей соответствующих блоков согласно формуле (9.8).

### 9.5.3 Хранение фрагментов видеопоследовательностей

Содержательные фрагменты, выделенные в длинной видеопоследовательности, можно сохранить в базе данных с целью обработки в дальнейшем запросов на поиск и получение видеоданных. Для доступа к видеоданным можно применять методы, рассматривавшиеся в гл. 8. Как описывалось в гл. 8, во фрагментах можно выбрать и использовать для поиска некоторые избранные *ключевые кадры*. Вероятно, в будущем будут разработаны более совершенные методы обработки видеоданных. Например, автоматический анализ движения позволит присваивать видеофрагментам смысловые метки наподобие «бег», «борьба», «обсуждение». Возможно, распознавание лиц позволит приписывать видеофрагментам фамилии известных людей, или, с помощью алгоритмов распознавания объектов общего назначения можно будет формировать метки наподобие «дом» и «лошадь». Обработка видеопоследовательностей, по сравнению с отдельными изображениями, требует существенно больших вычислительных затрат, но информация, получаемая с помощью анализа движения, поможет увеличить возможности сегментации для отделения объектов от фона и для классификации объектов.

---

#### Упражнение 9.14

Обсудите приложение (упоминавшееся ранее в данной главе) для обработки видеосъемок игры в теннис. (а) О каких действиях или событиях должна сообщать эта программа? (б) Какие количественные данные должна выдавать программа?

## 9.6 ЛИТЕРАТУРА

Слежение за игроками и мячом в теннисной игре описано в недавней работе [Pingali, Jean, Carlbom, 1998] из Bell Laboratories. В работе [Freeman и др., 1998] приведены результаты нескольких экспериментов, в которых методы компьютерного зрения использовались для организации интерфейса на основе распознавания жестов для управления несколькими существующими приложениями. В этой работе приведено описание быстрого алгоритма оценки движения. Детальное описание подобного алгоритма и игрового интерфейса содержится в работе [Kage и др., 1999]. Наше рассмотрение сегментации и индексирования видеопоследовательностей следует работам [Zhang и др., 1993] и [Smolier, Zhang, 1996]. Описание определения гладких траекторий по нескольким кадрам с точками без дополнительных характеристик основано на работе [Sethi, Jain, 1987]. Эта работа выполнена в тот период компьютерного зрения, когда была распространена практика изучения возможностей вычисления каких-либо свойств на основе небольшого набора предположений общего характера. В более поздних работах, например, [Maes и др., 1996], [Darrell и др., 1998] и [Bakic, Stockman, 1999], для ускорения процесса формирования

траектории и повышения надежности обработки учитываются специфические знания из предметной области конкретной задачи. В работе [Ayers, Shah, 1998] показано, как можно интерпретировать движение и изменения изображений в терминах, семантически значимых для приложений видеонаблюдения.

1. Ayers, D., and M. Shah. 1998. Recognizing human actions in a static room. *Proc. 4th IEEE Workshop on Applications of Computer Vision*, Princeton, NJ (19–21 Oct. 1998), 42–47.
2. Bakic, V., and G. Stockman. 1999. Menu selection by facial aspect. *Proc. Vision Interface '99*, Quebec, Canada (18–21 May 1999), 18–21.
3. Darrell, T. 1998. A radial cumulative similarity transform for robust image correspondence. *Proc. IEEE CVPR*, Santa Barbara, CA (June 1998), 656–662.
4. Darrell, T., G. Gordon, M. Harville, and J. Woodfill. 1998. Integrated person tracking using stereo, color, and pattern detection. *Proc. IEEE CVPR*, Santa Barbara, CA (June 1998), 601–608.
5. Freeman, W., D. Anderson, P. Beardsley, C. Dodge, M. Roth, C. Weissman, W. Yerazunis, H. Kage, K. Kyuma, Y. Miyake, and K. Tanaka. 1998. Computer vision for interactive computer graphics. *IEEE Comput. Graphics and Applications*, v. 18(3) (May–June 1998), 42–53.
6. Horn, B., and B. Schunck. 1981. Determining optical flow. *Artificial Intelligence*, v. 17:185–203.
7. Johansson, G. 1964. Perception of motion and changing form. *Scandinavian J. Psychology*, v. 5:181–208.
8. Kage, H., W. T. Freeman, Y. Miyake, E. Funatsu, K. Tanaka, and K. Kyuma. 1999. Artificial retina chips as on-chip image processors and gesture-oriented interfaces. *Optical Engineering*, 38(12):1979–1988.
9. Kasturi, R., and R. Jain. 1991. Dynamic vision. In *Computer Vision Principles*, R. Kasturi and R. Jain, eds. IEEE Computer Society Press, Washington, D.C., 469–480.
10. Maes, P., T. Darrell, B. Blumberg, and A. Pentland, 1996, *The ALIVE System: Wireless, Full-Body, Interaction with Autonomous Agents*. ACM Multimedia Systems: Special Issue on Multimedia and Multisensory Virtual Worlds, Sprint.
11. Pingali, G., Y. Jean, and I. Carlbom. 1998. Real time tracking for enhanced tennis broadcasts. *Proc. IEEE CVPR*, Santa Barbara, CA (June 1998), 260–265.
12. Salari, V., and I. Sethi. 1990. Correspondence of feature points in presence of occlusion. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, v. 12(1):87–91.

13. Sethi, I., and R. Jain. 1987. Finding trajectories of feature points in a monocular image sequence. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, v. 9(1):56–73.
14. Smolier, S., and H-J Zhang. 1996. Video indexing and retrieval. In *Multimedia Systems and Techniques*, B. Furht, ed. Kluwer Academic Publishers, Boston, 293–322.
15. Zhang, H-J., A. Kankanhalli, and S. Smoliar. 1993. Automatic partitioning of full-motion video. *Multimedia Systems*, v. 1(1):10–28.

# Сегментация изображений

Термин *сегментация изображения* означает разбиение изображения на множество покрывающих его областей. Цель многих задач анализа изображений заключается в сегментации на области, с которыми связана существенная для данной задачи информация. Например, при обработке спутниковой фотографии областями интереса могут быть области городских построек, лесных массивов и сельскохозяйственных полей. В других задачах в качестве областей интереса рассматриваются граничные пиксели, сгруппированные в такие структуры, как отрезки или дуги окружностей. Например, сегментация такого типа может применяться при обработке изображений трехмерных промышленных объектов. В качестве областей интереса также могут выбираться группы пикселей с границей определенной формы (окружность, эллипс, многоугольник и т. п.). Даже если области не покрывают изображение целиком, все равно можно говорить о сегментации изображения на области переднего плана (области интереса) и игнорируемые фоновые области.

Сегментация изображений имеет две основные цели. Первая заключается в декомпозиции изображения на части, более удобные для дальнейшего анализа. В простых случаях удается специально организовать сцены так, чтобы в процессе сегментации требовалось надежно выделять небольшое количество областей, необходимых для дальнейшей обработки. Например, в гл. 6 рассматривался алгоритм сегментации для обнаружения на цветном изображении образа человеческого лица. В этой задаче для успешной сегментации необходимо, чтобы одежда человека и другие попадающие в кадр объекты (например, стены комнаты) не были окрашены в цвета, похожие на цвет человеческой кожи. В других приложениях, например, при выделении дорожной сети на полутоновом изображении аэрофотосъемки, сегментация может оказаться весьма сложной задачей, для решения которой потребуется учитывать большое количество сведений предметной области. Вторая цель сегментации заключается в изменении формы описания изображения. В результате сегментации пиксели изображения преобразуются в высокоуровневые структуры, содержащие боль-



**Рис. 10.1.** (Слева) изображение футбольной игры. (Справа) результаты сегментации изображения. Каждая сегментированная область представляет собой множество связанных пикселей с похожими цветами. См. цветную вклейку.

ше информации или обеспечивающие эффективную организацию дальнейших операций анализа изображения (или и то, и другое).

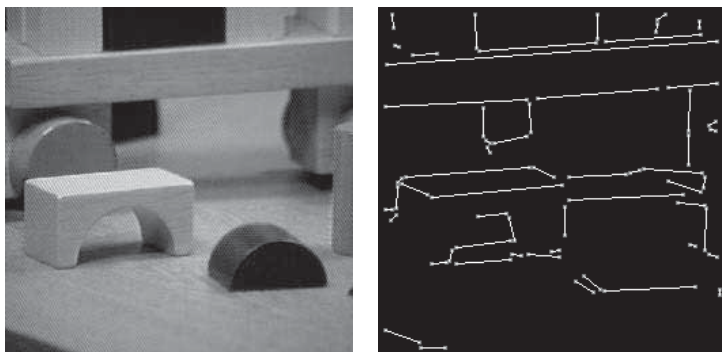
Ключевой вопрос состоит в том, можно ли выполнять сегментацию в различных задачах с помощью общих методов обработки «снизу-вверх» без учета специфической информации из конкретной предметной области. В данной главе рассматриваются методы сегментации, пригодные для использования во многих различных предметных областях. В следующих разделах описываются методы на основе областей и методы на основе границ. Перспективы разработки одного метода сегментации, одинаково хорошо подходящего для всех задач, выглядят весьма туманными. Опыт показывает, что разработчику приложений машинного зрения обычно приходится выбирать один из нескольких известных методов и, чаще всего, модифицировать этот метод с учетом специфических особенностей конкретной задачи.

В данной главе обсуждаются алгоритмы сегментации нескольких типов, в т. ч. классические алгоритмы наращивания областей, алгоритмы кластеризации и определители прямых и дуг окружностей. На рис. 10.1 представлены результаты сегментации цветного изображения футбольной игры на области примерно одинакового цвета. На рис. 10.2 показаны отрезки прямых, выделенные на изображении игрушечных деталей. Надо отметить, что результаты сегментации в обоих примерах, на взгляд человека, далеки от идеальных. Однако эти результаты могут использоваться в качестве входных данных для высокоуровневых процедур автоматической обработки, например, для идентификации игроков по номерам на майках или для распознавания деталей в задачах сборки.

## 10.1 ОБНАРУЖЕНИЕ ОБЛАСТЕЙ

- Области сегментированного изображения должны быть целостными и однородными по некоторому признаку (например, по интенсивности, цвету или текстуре).





**Рис. 10.2.** (Слева) изображение игрушечных деталей. (Справа) результаты выделения прямолинейных отрезков. Для выделения отрезков использовался пакет Object Recognition Toolkit. (С любезного разрешения John Illingworth, Ata Etamadi.)

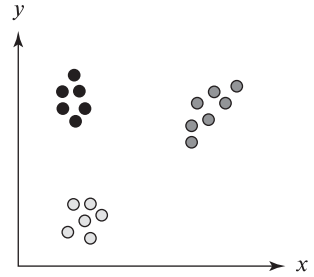
- Внутренние части сегментированных областей должны иметь простую форму и не содержать большого количества мелких отверстий.
- Смежные сегментированные области должны существенно отличаться по значению того характерного признака, по которому области являются однородными.
- Границы каждой сегментированной области должны быть гладкими, без зазубрин и разрывов.

Сложно обеспечить одновременное выполнение всех перечисленных требований, определяющих желательные свойства сегментированных областей. Например, строго однородные области обычно содержат много мелких отверстий и имеют зазубренные границы. Требование существенного отличия смежных областей по значению характерного признака может привести к слиянию нескольких областей и потере их границ. Кроме того, области, которые человек воспринимает как однородные, могут не быть таковыми на низком уровне характерных признаков, доступных процедуре сегментации. Тогда для сегментации может потребоваться учитывать высокоуровневую информацию. Цель данной главы состоит в рассмотрении алгоритмов, которые могут применяться к изображениям различных типов и обеспечивать получение информации для разнообразных высокоуровневых процедур анализа изображений.

### 10.1.1 Методы кластеризации

В распознавании образов *кластеризацией* (clustering) называется процесс разбиения множества векторов признаков на подмножества, называемые *кластерами*. Допустим, векторы признаков являются двумерными и представляются в виде пары действительных чисел (рис. 10.3). Кластеризация таких векторов может выполняться посредством поиска подмножеств близкорасположенных точек на евклидовой плоскости.

**Рис. 10.3.** На евклидовой плоскости задано множество точек, которые представляют результаты измерений. Это множество может быть разбито на три кластера, состоящих из близкорасположенных точек. В данном примере кластеры можно найти как подмножества точек, плотно расположенных внутри описывающих окружностей.



С кластеризацией связано большое количество разнообразных методов. В данной главе рассматриваются несколько разнотипных алгоритмов кластеризации, которые применяются при сегментации изображений. К ним относятся классические алгоритмы кластеризации, простые гистограммные методы, рекурсивный гистограммный метод Оландера (Ohlander) и метод графового разбиения Ши (Shi).

**Классические алгоритмы кластеризации.** Цель кластеризации в общем случае состоит в разбиении множества векторов на подмножества «похожих» векторов. В задачах анализа изображений векторы обычно представляют пиксели или, иногда, малые окрестности пикселей. Компонентами векторов признаков могут быть:

1. значения интенсивности;
2. цветовые коды RGB или вычисленные на их основе цветовые характеристики;
3. вычисленные характерные признаки;
4. текстурные характерные признаки.

Любой характерный признак, сопоставимый пикселу, может использоваться для их группировки. После группировки пикселей в кластеры на основе этих значений из *пространства измерений* с помощью алгоритма маркировки связанных компонент (гл. 3) легко можно найти связанные области.

Как обычно в задачах кластеризации, будем рассматривать  $K$  кластеров  $C_1, C_2, \dots, C_K$  с математическими ожиданиями  $m_1, m_2, \dots, m_K$ . *Квадратичная ошибка* определяется выражением

$$D = \sum_{k=1}^K \sum_{x_i \in C_k} \|x_i - m_k\|^2.$$

Эта величина характеризует близость исходных данных к назначенным для них кластерам. При кластеризации методом наименьших квадратов можно было бы перебирать все возможные разбиения на  $K$  кластеров и найти тот вариант, при котором  $D$  принимает минимальное значение. На практике такой подход неприменим из-за большого объема требуемых вычислений, поэтому

часто применяются приближенные методы. Важно решить, известно ли множество кластеров  $K$  заранее или нет. Многие алгоритмы требуют задавать  $K$  в качестве входного параметра. Другие алгоритмы пытаются построить наилучшее множество  $K$  на основе некоторого критерия, например, чтобы дисперсия каждого кластера не превышала заданного порогового значения.

**Итерационная кластеризация по математическому ожиданию.** Алгоритм *кластеризации по математическому ожиданию* (iterative K-means clustering) представляет собой простой итерационный алгоритм поиска экстремума. Основные действия этого алгоритма перечислены далее.

### Формирование $K$ кластеров на множестве $n$ -мерных векторов.

1. Присвоить счетчику итераций  $ic$  значение 1.
2. Случайным образом выбрать  $K$  значений математических ожиданий  $m_1(1), m_2(1), \dots, m_K(1)$ .
3. Для каждого вектора  $x_i$  вычислить расстояние  $D(x_i, m_k(ic))$  для каждого  $k = 1, \dots, K$  и поместить  $x_i$  в кластер  $C_j$  с ближайшим вектором математического ожидания.
4. Увеличить  $ic$  на 1 и скорректировать значения математических ожиданий для получения нового множества  $m_1(ic), m_2(ic), \dots, m_K(ic)$ .
5. Повторять шаги 3 и 4 до тех пор, пока при всех  $k$  не будет выполняться равенство  $C_k(ic) = C_k(ic + 1)$ .

**Алгоритм 10.1.** Алгоритм кластеризации по математическому ожиданию.

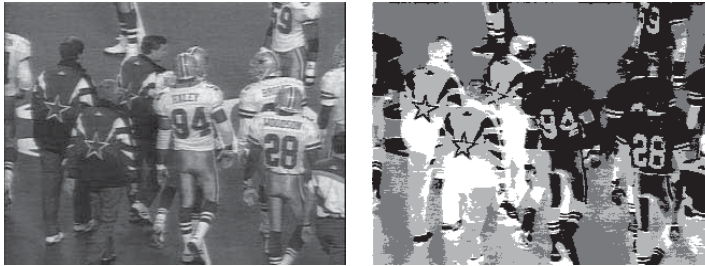
Этот алгоритм гарантированно завершается, но полученное решение не обязательно будет глобально оптимальным (в смысле критерия наименьшей квадратичной ошибки). Шаг 2 можно изменить для разбиения множества векторов на  $K$  случайных кластеров и последующего вычисления их математических ожиданий. На шаге 5 можно предусмотреть завершение работы, если на последней итерации смена кластеров была выполнена у небольшого количества векторов. На рис. 10.4 показан результат применения алгоритма кластеризации по математическому ожиданию в цветовом пространстве RGB к изображению футбольной игры.

**Алгоритм кластеризации ISODATA.** Алгоритм *ISODATA* — это еще один итерационный алгоритм, в основе которого лежит метод разделения-объединения. Снова будем рассматривать множество  $K$  кластеров  $C_1, C_2, \dots, C_K$  с математическими ожиданиями  $m_1, m_2, \dots, m_K$ . Обозначим через  $\Sigma_k$  ковариационную матрицу кластера  $k$  (определяется далее). Если  $x_i$  являются векторами вида

$$x_i = [v_1, v_2, \dots, v_n]$$



**Рис. 10.4.** (Слева) изображение футбольной игры. (Справа) на полутоновом изображении представлены  $K = 6$  кластеров, сформированные с помощью алгоритма кластеризации по математическому ожиданию. Шесть кластеров соответствуют шести преобладающим цветам исходного изображения: темно-зеленый, зеленый, темно-синий, белый, серебристый и черный. См. цветную вклейку.



**Рис. 10.5.** (Слева) изображение футбольной игры. (Справа) на полутоновом изображении представлены  $K = 5$  кластеров, сформированные с помощью алгоритма кластеризации ISODATA. Пять кластеров соответствуют пяти группам цветов: зеленый, темно-синий, белый, серебристый и черный. См. цветную вклейку.

то каждый вектор математического ожидания  $m_k$  имеет вид

$$m_k = [m_{1k}, m_{2k}, \dots, m_{nk}]$$

а матрица  $\Sigma_k$  выражается как

$$\Sigma_k = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \dots & \sigma_{1n} \\ \sigma_{12} & \sigma_{22} & \dots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1n} & \sigma_{2n} & \dots & \sigma_{nn} \end{bmatrix} \quad (10.1)$$

где  $\sigma_{ii} = \sigma_i^2$  — дисперсия  $i$ -й векторной компоненты  $v_i$ , а  $\sigma_{ij} = \rho_{ij}\sigma_i\sigma_j$  представляет собой ковариацию  $i$ -й и  $j$ -й векторных компонент ( $\rho_{ij}$  — коэффициент корреляции между  $i$ -й и  $j$ -й компонентами,  $\sigma_i$  — среднеквадратическое отклонение  $i$ -й компоненты,  $\sigma_j$  — среднеквадратическое отклонение  $j$ -й компоненты).

На рис. 10.5 приведены результаты обработки изображения футбольной игры (рис. 10.1) с помощью алгоритма кластеризации ISODATA (см. алгоритм

10.2) в цветовом пространстве RGB. Кластерное изображение использовалось в качестве входных данных для алгоритма маркировки связанных компонент. С его помощью были получены результаты сегментации, показанные на рис. 10.5. Пороговое значение  $\tau_v$  для алгоритма ISODATA было выбрано равным 10 процентам длины стороны цветового куба в пространстве RGB.

**Упражнение 10.1.** Сравнение алгоритмов кластеризации по математическому ожиданию и ISODATA.

При обработке изображений футбольной игры алгоритм ISODATA дал лучшие результаты по сравнению с кластеризацией по математическому ожиданию. Преимущество заключается в корректной группировке темно-зеленых областей в верхней части изображения с областями похожего цвета в нижней части. Как это объяснить?

**Простые гистограммные методы.** Итерационная модификация разбиения векторов приводит к многократной обработке данных изображения. В гистограммных методах выполняется только один проход по исходным данным, поэтому эти алгоритмы обычно работают быстрее итерационных алгоритмов кластеризации.

**Формирование кластеров «похожих» данных (isodata) на множестве  $n$ -мерных векторов.**

1. Поместить  $x_i$  в кластер  $l$ , при котором достигается минимум

$$D_{\Sigma} = [x_i - m_l]' \Sigma_l^{-1} [x_i - m_l].$$

2. Объединить кластеры  $i$  и  $j$ , если

$$|m_i - m_j| < \tau_v$$

где  $\tau_v$  — пороговое значение дисперсии.

3. Разделить кластер  $k$ , если максимальное собственное значение  $\Sigma_k$  превышает порог  $\tau_v$ .
4. Завершить работу, если для каждого кластера  $i$  выполняется условие

$$|m_i(t) - m_i(t + 1)| < \epsilon$$

или если было выполнено максимально допустимое количество итераций.

**Алгоритм 10.2.** Кластеризация методом ISODATA.

Метод *поиска моды гистограммы* (histogram mode seeking) предназначен для кластеризации в пространстве измерений. Предполагается, что однородные объекты на изображении образуют кластеры в пространстве измерений, в качестве которого рассматривается гистограмма. Сегментация изображения выполняется посредством обратного отображения кластеров из пространства измерений на изображение. При этом максимальные связные компоненты кластерных меток (т. е. связные компоненты на гистограмме) используются для выделения сегментов изображения. При обработке полутоновых изображений кластеризация в пространстве измерений может быть выполнена посредством поиска точек минимума на гистограмме. В качестве кластеров выбираются интервалы значений между точками минимума. Пиксел изображения со значением интенсивности, попадающим в  $i$ -й кластер гистограммы, маркируется индексом  $i$ . Сегмент изображения, которому принадлежит этот пиксел, является одной из связных компонент пикселей с метками  $i$ .

В гл. 3 рассматривался метод автоматического выбора порога бинаризации. Он представляет собой разновидность метода поиска моды гистограммы, рассчитанную на применение к бимодальной гистограмме.

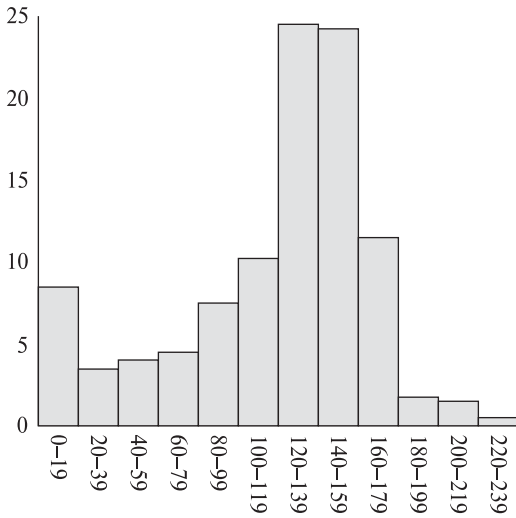
---

### **Упражнение 10.2.** Поиск моды гистограммы.

Напишите программу для поиска мод многомодальной гистограммы. Сначала разделите гистограмму на две части (как в методе Оцу, см. гл. 3), а затем попытайтесь рекурсивно делить каждую часть еще на две. Проверьте программу на нескольких полутоновых и цветных изображениях.

В общем случае полутоновое изображение имеет многомодальную гистограмму. Поэтому любой алгоритм автоматической бинаризации должен искать на гистограмме существенные точки максимума и разделяющие их точки минимума. Данную задачу решить намного сложнее, чем сформулировать. На рис. 10.6 показана гистограмма полутонового изображения игрушечных деталей. Простой алгоритм поиска точек минимума на гистограмме мог бы посчитать ее бимодальной и выбрать пороговое значение в интервале между 39 и 79. Однако алгоритм выбора порога методом проб и ошибок сгенерировал три пороговых значения. Для них были построены четыре бинарных изображения, приведенных на рис. 10.7. Все они содержат некоторые значимые области изображения. Данный пример поясняет необходимость использования методов бинаризации на основе априорной информации, в которых пороги бинаризации выбираются как с учетом содержания гистограммы, так и учетом характеристик качества/полезности выделенных областей применительно к решению конкретной задачи.

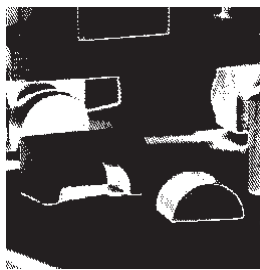
**Рекурсивный гистограммный метод Оландера.** Авторы работы [Ohlander, и др., 1978] усовершенствовали гистограммный метод кластеризации, предложив рекурсивную процедуру поиска мод. Основная идея заключалась в том, чтобы сначала искать моду на гистограмме полного изображения, а



**Рис. 10.6.** Гистограмма изображения деталей (рис. 10.2).



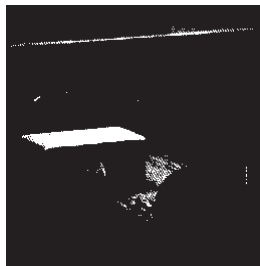
Диапазон бинаризации от 0 до 30



Диапазон бинаризации от 31 до 100



Диапазон бинаризации от 101 до 179



Диапазон бинаризации от 180 до 239

**Рис. 10.7.** Четыре бинарных изображения игрушечных деталей, полученные методом пороговой бинаризации по диапазону. Диапазон вручную задавался по гистограмме.

затем на гистограммах областей каждого сформированного кластера. Обработка повторяется до тех пор, пока удастся производить дальнейшее разделение сегментированных областей.

В начале обработки формировалась маска, выбирающая все пиксели изображения. Была реализована процедура построения гистограммы для области изображения, заданной с помощью маски. Для этой гистограммы выполнялась кластеризация в пространстве измерений. В результате формировалось множество кластеров. Затем устанавливалась принадлежность пикселей изображения найденным кластерам. Если кластеров было больше одного, то для каждого из них выполнялась маркировка связных компонент и строилось множество связных областей, соответствующих каждому кластеру. Затем для каждой связной компоненты генерировалась новая маска, которая помещалась в стек масок.

Маски в стеке представляли области, являющиеся кандидатами для дальнейшей сегментации. В течение последовательных итераций, очередная маска из стека определяла пиксели для построения гистограммы. Кластеризация повторялась до исчерпания стека масок.

Описанный процесс, который будем называть рекурсивной пространственной кластеризацией на основе гистограммы, поясняется на рис. 10.8.

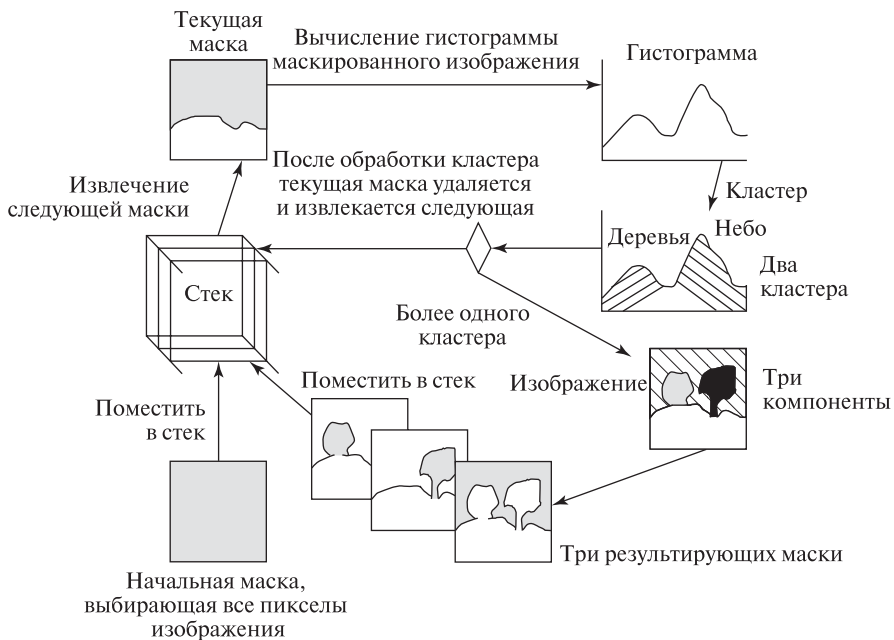
Для обычных цветных изображений авторы статьи [Ohta, и др., 1980] предложили вычислять гистограммы не по каждой из трех цветовых компонент RGB, а по набору переменных, похожих на основные компоненты преобразования Карунена-Ловэа:  $(R + G + B)/3$ ,  $(R - B)/2$  и  $(2G - R - B)/4$ .

**Графовое разбиение Ши\*** Алгоритмы Оландера (Ohlander) и Ота (Ohta) хорошо работают с относительно простыми цветными изображениями сцен с искусственными объектами и одноцветными областями. Результаты обработки сложных изображений естественных сцен оказываются хуже, так как в текстурных областях образуется большое количество мелких сегментированных областей. В работе [Shi, Malik, 1997] представлен метод, позволяющий выполнять сегментацию на основе цвета, текстуры, или любой комбинации этих и других характерных признаков. Авторы этой работы сформулировали задачу сегментации в форме задачи разбиения графа и разработали новый метод разбиения графа, который сводится к решению задачи собственных векторов и собственных значений.

Рассмотрим взвешенный граф  $G = (V, E)$ , узлы которого представляют точки в пространстве измерений. Ребрам соответствуют веса  $w(i, j)$ , характеризующие сходство узлов  $i$  и  $j$ . Цель сегментации заключается в нахождении такого разбиения вершин на непересекающиеся множества  $V_1, V_2, \dots, V_m$ , которое характеризуется высоким сходством элементов внутри множеств и низким сходством элементов из разных множеств.

Граф  $G = (V, E)$  можно разбить на два непересекающихся графа с множествами узлов  $A$  и  $B$ , если удалить все ребра, соединяющие узлы множества





**Рис. 10.8.** Схема рекурсивной пространственной кластеризации на основе гистограммы. На исходном изображении заметны четыре области: трава, небо и два дерева. Текущая маска (показана вверху слева) соответствует области, содержащей небо и деревья. Кластеризация гистограммы этой области позволяет сформировать два кластера в цветовом пространстве, один для неба и один для деревьев. Кластер неба содержит одну связную компоненту, а кластер деревьев — две. По каждой из трех связных компонент строится маска и помещается в стек масок для использования на следующих итерациях сегментации.

$A$  с узлами множества  $B$ . Численную характеристику различия множеств  $A$  и  $B$  можно определить как сумму весов удаленных ребер. Этот суммарный вес называется *разрезом* (cut):

$$\text{cut}(A, B) = \sum_{u \in A, v \in B} w(u, v) \quad (10.2)$$

В одной из формулировок задачи сегментации предполагается нахождение *минимального разреза* (minimum cut) графа и повторение этого поиска до тех пор, пока сегментированные области не станут достаточно однородными. Однако критерий минимального разреза приводит к формированию небольших множеств изолированных узлов. Это нежелательно при поиске больших однородных областей по цвету или текстуре. Ши (Shi) предложил использовать характеристику *нормированного разреза* (normalized cut, Ncut), определяемого с помощью разреза  $\text{cut}(A, B)$  и *ассоциации* множества  $A$  и полного множества

вершин  $V$ :

$$asso(A, V) = \sum_{u \in A, t \in V} w(u, t) \quad (10.3)$$

Нормированный разрез определяется следующим образом:

$$Ncut(A, B) = \frac{cut(A, B)}{asso(A, V)} + \frac{cut(A, B)}{asso(B, V)} \quad (10.4)$$

Согласно этому определению, разрезы, отделяющие небольшие множества изолированных точек, не могут иметь малые значения  $Ncut$ . Поэтому в задаче сегментации изображения предпочтительными являются разрезы с малыми значениями  $Ncut$ .

Еще одной используемой величиной является общая нормированная ассоциация:

$$Nasso(A, B) = \frac{asso(A, A)}{asso(A, V)} + \frac{asso(B, B)}{asso(B, V)} \quad (10.5)$$

Эта величина характеризует, насколько сильно связаны друг с другом узлы данного множества. Она связана с  $Ncut$  выражением:

$$Ncut(A, B) = 2 - Nasso(A, B) \quad (10.6)$$

так что в процедуре разбиения графа можно использовать как  $Ncut$ , так и  $Nasso$ .

С учетом определений  $Ncut$  и  $Nasso$  требуется построить процедуру для сегментации изображения путем разбиения множества пикселей. Процедура сегментации Ши представлена в виде алгоритма 10.3.

Ши применил этот алгоритм для сегментации изображений по яркости, цвету и текстурным признакам. Вес ребра  $w(i, j)$  определялся выражением:

$$w(i, j) = e^{-\frac{\|F(i) - F(j)\|_2}{\sigma_I}} * \begin{cases} e^{-\frac{\|X(i) - X(j)\|_2}{\sigma_X}} & \|X(i) - X(j)\|_2 < r \\ 0 & \text{иначе} \end{cases} \quad (10.11)$$

где

- $X(i)$  — пространственное местоположение узла  $i$ .
- $F(i)$  — вектор признаков, построенный на основе яркостной, цветовой или текстурной информации и определяемый следующим образом:
  - При сегментации полутоновых изображений  $F(i) = I(i)$  (значение интенсивности (яркости)).
  - При сегментации цветных изображений  $F(i) = [v, v \cdot s \cdot \sin(h), v \cdot s \cdot \cos(h)](i)$ , где  $h$ ,  $s$ , и  $v$  — цветовые коды HSV.

**Выполнение графовой кластеризации на графе, узлы которого соответствуют пикселям, а веса ребер характеризуют сходство пар пикселей.**

1. Сформируем взвешенный граф  $G = (V, E)$ . Множество вершин  $V$  соответствует множеству пикселей изображения, а множество ребер  $E$  является множеством взвешенных ребер. Веса ребер  $w(i, j)$  между узлами  $i$  и  $j$  вычисляются на основе меры сходства векторов  $i$  и  $j$  в пространстве измерений. Размер множества вершин  $V$  обозначим  $N$ . Определим вектор  $d$  с компонентами  $d(i)$  вида

$$d(i) = \sum_j w(i, j) \quad (10.7)$$

так что  $d(i)$  описывает соединения узла  $i$  со всеми другими узлами. Обозначим символом  $D$  диагональную матрицу  $N \times N$  с компонентами вектора  $d$  на диагонали. Обозначим символом  $W$  симметричную матрицу  $N \times N$  с элементами  $W(i, j) = w(i, j)$ .

2. Введем вектор  $x$ , компоненты которого определяются следующим образом:

$$x_i = \begin{cases} 1 & \text{если } i \text{ принадлежит } A \\ -1 & \text{в противном случае} \end{cases} \quad (10.8)$$

Определим для  $x$  непрерывное приближенное представление  $y$

$$y = (1 + x) - \frac{\sum_{x_i > 0} d_i}{\sum_{x_i < 0} d_i} (1 - x). \quad (10.9)$$

Решим систему уравнений

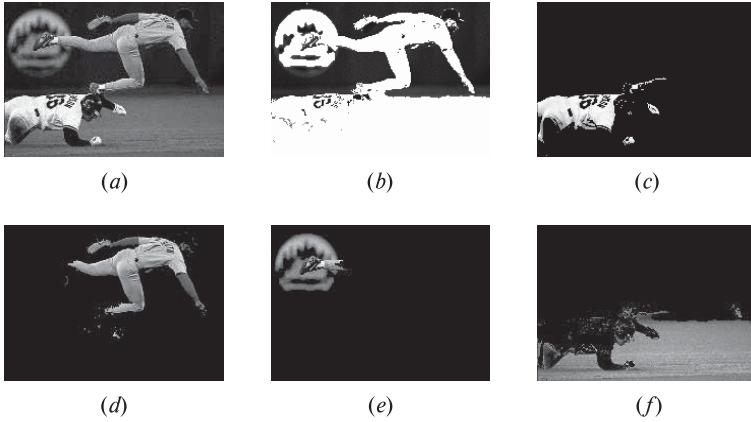
$$(D - W)y = \lambda Dy \quad (10.10)$$

для нахождения собственных векторов  $y$  и собственных значений  $\lambda$ .

3. Используем собственный вектор со вторым наименьшим собственным значением для бинарного разбиения графа и нахождения такой точки разбиения, которой соответствует минимальное значение  $Ncut$ .<sup>a</sup>
4. Оценим необходимость дальнейшего разбиения. Для этого разрез должен быть стабильным, а значение  $Ncut$  не должно превышать заданного порога.
5. При необходимости выполним рекурсивное разбиение сегментированных частей графа.

**Алгоритм 10.3.** Кластеризация графовым методом Ши.

<sup>a</sup> Ши показал, что второй наименьший собственный вектор обобщенной собственной системы является действительным решением задачи о нормированном разрезе.



**Рис. 10.9.** Исходное полутоновое изображение (a) и области, полученные с помощью алгоритма сегментации Ши (b)–(f). На результирующем изображении (b) выделена темная фоновая область. Она показана черным цветом. На остальных изображениях выделенные области отображены с использованием исходных оттенков серого цвета, а оставшаяся часть изображений заполнена черным цветом. (С любезного разрешения Jianbo Shi.)

- При текстурной сегментации  $F(i) = [|I * f_1|, \dots, |I * f_n|](i)$ , где  $f_i$  — разностные гауссовские фильтры различных масштабов и ориентации.

Отметим, что вес ребра  $w(i, j) = 0$  для любой пары узлов  $i$  и  $j$ , расстояние между которыми превышает заданный порог  $r$  (в пикселах).

Алгоритм 10.3 показывает весьма хорошие результаты цветовой и текстурной сегментации. На рис. 10.9 демонстрируется применение этого алгоритма к изображению естественной сцены. Но, несмотря на хорошие результаты, сложность этого алгоритма делает его малоподходящим для использования в системах реального времени.

### 10.1.2 Нарращивание областей

В противоположность методам разбиения изображения, при *наращивании областей* обработка начинается с некоторой точки изображения (обычно в левом верхнем углу) и затем производятся попытки наращивания каждой области. Нарращивание не выполняется, если сравниваемые пикселы сильно отличаются от пикселов областей. Для сравнения обычно используются статистические критерии. В 1985 г. Харалик и Шапиро (Haralick, Shapiro) предложили метод наращивания областей, называемый методом наращивания областей Харалика. В нем предполагается, что область представляет собой множество связанных пикселов с общим выборочным средним и дисперсией.

Обозначим символом  $R$  область из  $N$  пикселей, соседних с пикселем с интенсивностью  $y$ . Определим выборочное среднее  $\bar{X}$  и разброс  $S^2$  выражениями

$$\bar{X} = \frac{1}{N} \sum_{[r,c] \in R} I[r,c] \quad (10.12)$$

и

$$S^2 = \sum_{[r,c] \in R} (I[r,c] - \bar{X})^2 \quad (10.13)$$

В предположении, что все пиксели  $R$  и проверяемый пиксел  $y$  независимы и их интенсивности описываются одинаковым нормальным распределением, случайная величина

$$T = \left[ \frac{(N-1)N}{(N+1)} (y - \bar{X})^2 / S^2 \right]^{\frac{1}{2}} \quad (10.14)$$

будет описываться  $t$ -распределением (распределением Стьюдента) с  $(N-1)$  степенями свободы. Если значение  $T$  достаточно мало, то пиксел  $y$  добавляется к области  $R$  и значения выборочного среднего и разброса пересчитываются для учета значения  $y$ . Новое среднее значение и разброс вычисляются по формулам:

$$\bar{X}_{\text{new}} \leftarrow (N\bar{X}_{\text{old}} + y) / (N+1) \quad (10.15)$$

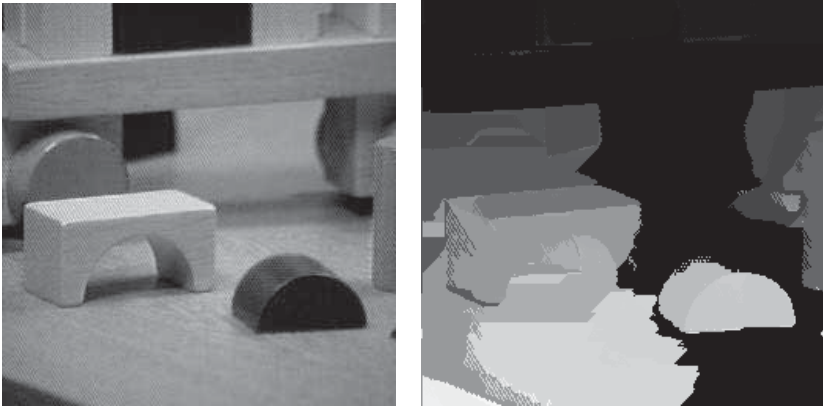
и

$$S_{\text{new}}^2 \leftarrow S_{\text{old}}^2 + (y - \bar{X}_{\text{new}})^2 + N(\bar{X}_{\text{new}} - \bar{X}_{\text{old}})^2 \quad (10.16)$$

Если значение  $T$  велико, то это означает, что вероятность принадлежности пиксела  $y$  области  $R$  мала. Если значение  $y$  отличается от всех соседних областей, то этот пиксел принимается в качестве новой области. Более строгий критерий наращивания может требовать не только близости значения  $y$  к выборочному среднему соседней области, но и чтобы соседний пиксел из данной области имел значение, достаточно близкое к  $y$ .

Для уточнения понятия «слишком большого различия», можно воспользоваться статистическим критерием значимости по  $\alpha$ -процентным точкам. Значение  $\alpha$  представляет вероятность того, что значение статистики  $T$  с  $N-1$  степенями свободы превысит значение  $t_{N-1}(\alpha)$ . Если наблюдаемое значение  $T$  превышает  $t_{N-1}(\alpha)$ , то различие считается существенным. Если данный пиксел и сегментированная область в действительности образуют одну область изображения с общим распределением значений признака, то вероятность неверного ответа при проверке по приведенному критерию равна  $\alpha$ .

Уровень значимости  $\alpha$  задается пользователем. Значения  $t_{N-1}(\alpha)$  больше при малом количестве степеней свободы и меньше при большом количестве степеней свободы. Значит, значение разброса в пределах области считается одинаковым, и чем больше область, тем ближе значение пиксела должно быть к среднему значению области, чтобы следовало присоединить пиксел к данной области. Это правило предотвращает присоединение к большим областям мно-



**Рис. 10.10.** (Слева) изображение игрушечных деталей. (Справа) результаты сегментации методом наращивания областей Харалика. (С любезного разрешения John Pllingworth, Ata Etamadi. Сегментация выполнена с помощью системы обработки изображений GIPSY.)

гих лишних пикселей и дрейф выборочного среднего при увеличении размера области. На рис. 10.10 показан результат сегментации изображения методом наращивания областей Харалика.

---

### Упражнение 10.3. Наращивание областей.

Напишите программу с реализацией сегментации методом наращивания областей Харалика и проверьте работу этой программы на нескольких полутоновых изображениях.

## 10.2 СПОСОБЫ ПРЕДСТАВЛЕНИЯ ОБЛАСТЕЙ

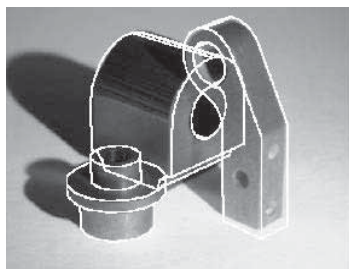
В любом алгоритме, обнаруживающем множество областей изображения, необходимо предусмотреть сохранение характеристик найденных областей для последующего использования. Известно несколько различных способов представления областей, в т. ч. оверлейное наложение на исходные изображения, маркированные изображения, кодирование границ, использование структур данных на основе квадрантных деревьев и таблиц свойств. Вероятно, чаще всего для представления областей применяются маркированные изображения. Далее будет описан каждый из перечисленных способов представления.

### 10.2.1 Оверлейные представления

Оверлейное представление — это метод отображения выделенных областей изображения, при котором поверх исходного изображения с использованием одного или нескольких цветов выводится информация об областях интереса.



(a) Оверлейное представление границ областей



(b) Оверлейное представление каркасной модели

**Рис. 10.11.** Примеры оверлейных представлений. (a) Оверлейное наложение границ выделенных областей на исходное изображение футбольной игры. (b) Оверлейное наложение каркасных моделей трехмерных объектов на изображение машиностроительных деталей. (Изображение (b) предоставлено с любезного разрешения Mauro Costa.)

Во многих пакетах обработки изображений оверлейное представление реализовано как одна из стандартных процедур вывода изображений. Исходное изображение обычно является полутоновым, а в качестве оверлейного цвета выбирается хорошо заметный на таком изображении цвет, например, красный или белый. Для демонстрации результатов сегментации можно было бы присвоить граничным пикселям сегментированных областей значения белого цвета и отобразить полученное преобразованное полутоновое изображение. В целях отображения границ областей иногда формируются границы толщиной больше одного пикселя.

На рис. 10.11 (a) показаны границы выделенных темных областей, в т. ч. темно-синие куртки судей и номера игроков. Эти границы наложены на исходное полутоновое изображение футбольной игры. Еще одно применение оверлейных представлений состоит в отображении некоторых характерных признаков изображения. На рис. 10.11 (b) воспроизведено изображение сцены с тремя деталями из гл. 1. На это полутоновое изображение наложены проекции каркасных моделей распознаваемых объектов.

### 10.2.2 Маркированные изображения

Маркированные изображения представляют собой полезное промежуточное представление областей, которое может применяться при дальнейшей обработке. Идея заключается в том, чтобы каждой обнаруженной области присвоить уникальный идентификатор (обычно целое число) и построить изображение, на котором всем пикселям области присвоено значение идентификатора данной области. Подобные выходные данные генерируют большинство алгоритмов маркировки связанных компонент (гл. 3). Маркированное изображение можно применять в качестве маски, определяющей пиксели области при выполнении

каких-либо операций по вычислению свойств области (например, площади или длин полуосей при аппроксимации области эллипсом). Маркированные изображения можно отображать в полутоновом или псевдоцветовом виде.

Если целочисленные идентификаторы областей имеют небольшие значения, то при отображении в полутоновом режиме маркированное изображение может оказаться слишком темным. В таком случае оно может быть отображено в увеличенном виде. Для облегчения восприятия полутоновых изображений также применяются методы выравнивания гистограммы изображения. Приведенные выше сегментированные изображения футбольной игры представляют собой маркированные изображения, отображенные в полутоновом виде.

### 10.2.3 Кодирование границ

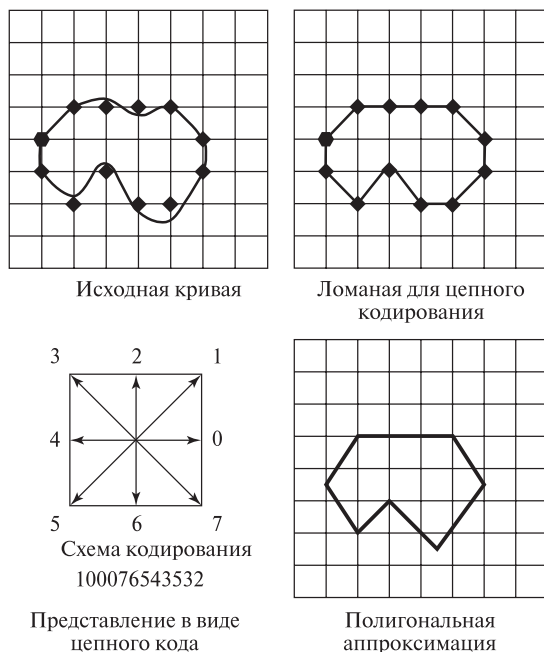
Для представления областей можно использовать их границы. Информация о границах обычно хранится в некоторой структуре данных, а не на изображении. Одной из простейших структур данных является линейный список граничных пикселей каждой области. (Далее будет рассмотрен алгоритм, выделяющий границы областей на маркированном изображении.)

*Цепной код Фримана* (Freeman chain code) представляет собой разновидность списка граничных точек, обеспечивающую более компактное представление границ. При преобразовании в цепной код информация из списка точек подвергается дополнительной дискретизации. Кратко рассмотрим основной принцип цепного кодирования. Кодированная граничная кривая накладывается на квадратную сетку. Размеры клеток определяют пространственное разрешение кодирования. У граничной кривой выбирается начальная точка. Начиная с этой точки, определяются узлы квадратной сетки, ближайшие к граничной кривой. Эти узлы рассматриваются в качестве вершин ломаной, аппроксимирующей граничную кривую. Каждая сторона ломаной является отрезком, соединяющим узел сетки с одним из его соседей. Направления этих отрезков кодируются небольшими целыми числами в полуинтервале от 0 до количества рассматриваемых при кодировании соседей. Совокупность кодов всех отрезков ломаной линии образует цепной код.

На рис. 10.12 поясняется способ кодирования восьмисвязным цепным кодом. Значение кода 0 соответствует направлению отрезка  $0^\circ$ , значение кода 1 — направлению  $45^\circ$  и т. д., до значения 7 для направления отрезка  $315^\circ$ .

Начальная точка замкнутой кривой на рис. 10.12 отмечена шестиугольником, а остальные точки пересечения сетки и кривой обозначены ромбами. Для восстановления граничной кривой с разрешением выбранной сетки необходимо знать координаты начальной точки и цепной код. Цепной код не только уменьшает объем памяти для хранения границ, но и может использоваться в ряде операций обработки кривых, например, при распознавании объектов по форме. Если область имеет не только внешнюю границу, но и содержит одно или несколько отверстий, то каждое из них можно представить отдельным цепным кодом.





**Рис. 10.12.** Два способа кодирования границ: цепной код и полигональная аппроксимация. В цепном коде применяется восьмизначный код для представления восьми возможных углов наклона прямолинейных отрезков, с помощью которых на квадратной сетке аппроксимируется граничная кривая. При полигональной аппроксимации также используются отрезки, приближенно представляющие исходную кривую. Но, в отличие от цепного кода, вершины отрезков имеют вещественные координаты и не обязательно совпадают с узлами сетки.

Если при хранении граничной кривой не требуется запоминать координаты всех ее пикселей, то кривую можно приближенно представить с помощью отрезков, формирующих *полигональную аппроксимацию* кривой (см. рис. 10.12, справа внизу). Данный способ представления позволяет сэкономить память и упростить алгоритмы обработки границ.

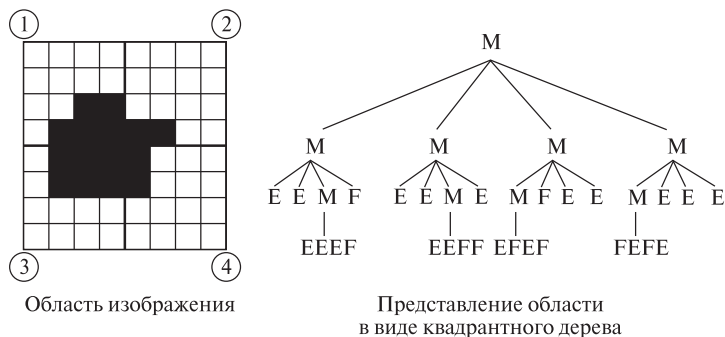
#### 10.2.4 Квадрантные деревья

*Квадрантные деревья* — это еще один способ компактного представления областей. Области описываются целиком, а не только их границы. В виде квадрантного дерева, в принципе, можно представить любую область интереса.

Каждый узел квадрантного дерева сопоставлен некоторой квадратной области изображения. Узел помечен одной из трех меток: *заполнено* (full, F), *пусто* (empty, E), *частично заполнено* (mixed, M). Если узел помечен как заполненный, то каждый пиксел квадратной области принадлежит области

интереса. Если узел помечен как пустой, то соответствующая квадратная область и область интереса не пересекаются. Если узел помечен как частично заполненный, то некоторые пиксели квадратной области являются пикселями области интереса, а некоторые нет. В квадрантном дереве дочерние узлы есть только у частично заполненных узлов. Заполненные и пустые узлы являются листовыми.

На рис. 10.13 приведен пример представления области изображения в виде квадрантного дерева. Граница области выглядит неровной, так как изображение имеет размеры всего  $8 \times 8$  пикселей. Поэтому квадрантное дерево имеет 4 уровня. Для получения достаточно гладкой границы могло бы потребоваться большее разрешение изображения и, соответственно, большая глубина дерева. Квадрантные деревья применяются для представления областей карт в геоинформационных системах.



**Рис. 10.13.** Представление области изображения в виде квадрантного дерева. Четыре дочерних узла соответствуют верхнему левому, верхнему правому, нижнему левому и нижнему правому квадрантам. Для первого уровня дерева порядок перечисления квадрантов обозначен цифрами внутри окружностей. Назначение символьных меток узлов: М=частичное заполнение, Е=пусто и F=заполнено.

### 10.2.5 Таблицы свойств

Иногда требуется представить область с помощью выделенных признаков, а не пикселей области. В таких случаях применяется представление, называемое *таблицей свойств* (property table). Таблица свойств напоминает таблицу реляционной базы данных. Каждой области изображения соответствует строка таблицы, а каждому свойству — столбец. В качестве свойств могут быть выбраны размер, форма, интенсивность, цвет, текстура области, а также любые другие характерные признаки, рассматривавшиеся в главах 3, 6 и 7. Например, в системе для поиска изображений на основе содержания области могли бы описываться с использованием площади, отношения полуосей аппроксимирующего эллипса, двух преобладающих цветов, и одним или несколькими текстурными

признаками. В таблице свойств может предусматриваться непосредственное хранение или ссылка на описание области в виде цепного кода или квадрантного дерева.

---

**Упражнение 10.4.** Вычисление площади и периметра

Область изображения представлена (а) на маркированном изображении и (б) с помощью цепного кода.

1. Предложите алгоритмы для вычисления площади и периметра области.
2. Оцените временные затраты разработанных алгоритмов.

---

**Упражнение 10.5.** Проверка принадлежности пикселей области

Область изображения представлена (а) на маркированном изображении и (б) с помощью полигональной аппроксимации границы области.

1. Для обоих вариантов предложите алгоритм проверки принадлежности произвольного пиксела  $[r, c]$  заданной области.
2. Оцените временные затраты разработанных алгоритмов в зависимости от существенных параметров задачи, например, от количества пикселей области или количества сегментов полигональной аппроксимации.

## 10.3 ОБНАРУЖЕНИЕ КОНТУРОВ

В некоторых задачах анализа изображений непосредственно обрабатывается информация об областях изображений, а в других задачах требуется использовать границы этих областей или некоторые другие структуры, например, прямолинейные отрезки или дуги окружностей. В данном разделе рассматриваются вопросы, связанные с извлечением подобных структур из изображений.

### 10.3.1 Прослеживание границ существующих областей

После получения множества областей изображения (например, с помощью процедуры сегментации или маркировки связных компонент) можно найти границы каждой области. *Выделение границ* (boundary extraction) особенно легко выполнить на изображениях небольшого размера. Для этого достаточно перебрать все точки изображения и сформировать список, содержащий первые граничные точки каждой связной компоненты. Затем для каждой области, начиная с первой граничной точки, надо прослеживать границу связной компоненты по часовой стрелке, до тех пор, пока не произойдет возврат в начальную точку. При обработке больших изображений, которые могут не помещаться в памяти целиком, этот простой алгоритм прослеживания границ

может привести к большому количеству обращений к устройству ввода-вывода, на котором хранится изображение.

Рассмотрим алгоритм *border*, предназначенный для выделения границ всех областей посредством просмотра изображения в направлении слева-направо, сверху-вниз. В качестве входных данных *border* принимает маркированное изображение. Выходными данными являются границы каждой области в виде списка координат точек, упорядоченных в направлении обхода по часовой стрелке. При необходимости алгоритм может быть модифицирован для выделения границ только некоторых заданных областей.

Значения пикселей маркированного изображения, передаваемого алгоритму *border*, соответствуют меткам областей. Предполагается, что имеется только одна фоновая область. Ее метка назначается тем областям, возможно, не связным, границы которых выделять не требуется. Вместо того чтобы по отдельности проследивать границу каждой области, алгоритм обрабатывает пиксели изображения слева-направо, сверху-вниз и формирует цепочки граничных пикселей, которые образуют связные фрагменты границ областей. В процессе работы алгоритма в любой момент времени существует множество *текущих областей* (*current regions*), границы которых были частично выделены, но еще не окончательно, множество *обработанных областей* (*past regions*), чьи границы были полностью выделены, и множество *ожидающих областей* (*future regions*), которые в процессе перебора пикселей изображения еще не встретились.

Цепочки граничных пикселей текущих областей хранятся в структурах данных. На изображении может присутствовать много различных меток областей, но одновременно активными могут быть только  $2 \times \text{number\_of\_columns}$ . Для быстрого доступа к цепочкам областей по значению метки области может использоваться хэш-таблица. (Значение  $(2 \times \text{number\_of\_columns})$  является безопасной верхней границей, действительное количество областей обычно меньше.) Если область полностью обработана и ее граница помещена в выходные данные, то информация об этой области из хэш-таблицы удаляется. Когда в процессе обработки изображения встречается новая область, то она добавляется в хэш-таблицу. Запись хэш-таблицы для данной области содержит указатель на связный список цепочек, которые были сформированы к настоящему моменту для данной области. Каждая цепочка представляет собой связный список координат граничных пикселей, упорядоченных в порядке обнаружения.

На каждом шаге алгоритма прослеживания границ обращается к трем последовательным строкам маркированного изображения. Это текущая обрабатываемая строка, а также строки выше и ниже текущей. Сверху и снизу изображения добавляется по одной фиктивной строке фоновых пикселей, чтобы все строки обрабатывались однотипно. Алгоритм обработки маркированного изображения **S** размерами **NLINES** на **NPIXELS** рассмотрен далее.

В описании алгоритма символом **S** обозначено маркированное изображение. Тогда **S[R, C]** представляет собой значение (**LABEL**) текущего обрабатываемого

## Выделение границ всех областей маркированного изображения S

**S[R, C]** — входное маркированное изображение.

**NLINES** — количество строк изображения.

**NPIXELS** — количество пикселей в строке изображения.

**NEWCHAIN** — флаг, равный true, когда пиксел начинает новую цепочку, и false, когда новый пиксел добавляется в существующую цепочку.

```
procedure border(S);
{
  for R:= 1 to NLINES
  {
    for C:= 1 to NPIXELS
    {
      LABEL:= S[R, C];
      if new_region(LABEL) then add(CURRENT,LABEL);
      NEIGHB:= neighbors(R,C,LABEL);
      T:= pixeltype(R,C,NEIGHB);
      if T == 'border' then
        for each pixel N in NEIGHB
        {
          CHAINSET:= chainlist(LABEL);
          NEWCHAIN:= true;
          for each chain X in CHAINSET
            while NEWCHAIN
              if N==rear(X) then
                {
                  add(X,[R, C]);
                  NEWCHAIN:= false
                }
              if NEWCHAIN then
                make_new_chain(CHAINSET,[R,C],LABEL);
            }
          }
        }
      for each region REG in CURRENT
        if complete(REG) then
          {
            connect_chains(REG);
            output(REG);
            free(REG)
          }
        }
      }
    }
  }
}
```

	1	2	3	4	5	6	7
1	0	0	0	0	0	0	0
2	0	0	0	0	2	2	0
3	0	1	1	1	2	2	0
4	0	1	1	1	2	2	0
5	0	1	1	1	2	2	0
6	0	0	0	0	2	2	0
7	0	0	0	0	0	0	0

(a) Маркированное изображение с двумя областями

**Рис. 10.14.** Обработка маркированного изображения с помощью алгоритма выделения границ областей.

Область	Длина	Список
1	8	(3, 2)(3, 3)(3, 4)(4, 4)(5, 4)(5, 3)(5, 2)(4, 2)
2	10	(2, 5)(2, 6)(3, 6)(4, 6)(5, 6)(6, 6)(6, 5)(5, 5)(4, 5)(3, 5)

(b) Выходные данные процедуры выделения границ областей на маркированном изображении

мого пиксела. Если это новая метка, то она добавляется в множество меток текущих областей **CURRENT\_NEIGHB** — это список соседей пиксела  $[R, C]$  с метками **LABEL**. Функция *pixeltype* просматривает значения пиксела  $[R, C]$  и его соседей, чтобы решить, является ли  $[R, C]$  граничным пикселом не фоновой области. Если является, то процедура ищет цепочку области с меткой **LABEL**, в конце которой расположен соседний пиксел для  $[R, C]$ . Если такая цепочка находится, то в ее конец добавляется пиксел  $[R, C]$ . Для этого вызывается процедура *add*, которой в качестве первого параметра передается цепочка, а в качестве второго — пиксел  $[R, C]$ . Если в конце цепочки области нет соседа пиксела  $[R, C]$ , то создается новая цепочка с единственным пикселом  $[R, C]$ . Для этого вызывается процедура *make\_new\_chain*. Ее первым параметром является множество цепочек, в которое надо добавить новую цепочку с единственным элементом  $[R, C]$ , передаваемым во втором параметре. Третий параметр представляет собой метку **LABEL**, соответствующую новой цепочке.

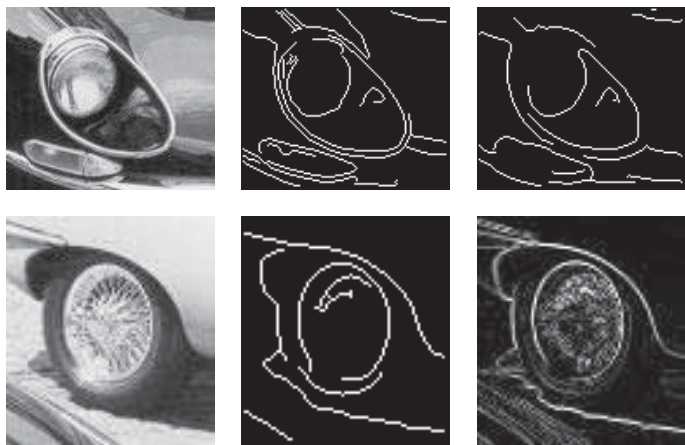
После обработки каждой строки **R**, цепочки тех текущих областей, границы которых были завершены, объединяются в одну границу и помещаются в выходные данные. Из хэш-таблицы и списка удаляются элементы, связанные с этими полностью обработанными областями. На рис. 10.14 показано маркированное изображение и выходные данные алгоритма выделения границ областей.

### Упражнение 10.6.

Ограничения алгоритма прослеживания границ  
Алгоритм прослеживания границ разработан с учетом нескольких предположений относительно свойств обрабатываемых областей. При каких условиях этот алгоритм не смог бы корректно выделить границу области?

### 10.3.2 Детектор и компоновщик краев Кэнни

Детектор и компоновщик краев Кэнни (Canny) предназначен для выделения граничных сегментов (контуров) на полутоновом изображении. Он кратко рассматривался в гл. 5 вместе с другими детекторами краев. Оператор Кэнни часто применяется на практике. Недавние работы, в которых проводится сравнение различных детекторов краев, подтверждают популярность оператора Кэнни. Примеры его использования рассматривались в гл. 5. На рис. 10.15 показаны два фрагмента изображения автомобиля, взятых из более крупного изображения из гл. 2. Оба этих изображения иллюстрируют хорошо известные проблемы, возникающие при использовании любых детекторов краев и алгоритмов прослеживания границ. Сегменты контуров частей объектов ошибочно сливаются с сегментами контуров засвеченных или теневых областей. Такие контуры сложно анализировать с помощью методов «снизу-вверх», часто применяемых в системах распознавания объектов. Однако, как будет показано в последующих главах, сопоставление таких контурных представлений с моделями известных объектов может успешно выполняться методами «сверху-вниз». Следовательно, оценка качества контурных представлений изображений зависит от способа их использования в системе машинного зрения.



**Рис. 10.15.** (Сверху слева) изображение фары черного автомобиля. (Сверху в центре) результаты применения оператора Кэнни при  $\sigma = 1$ . (Сверху справа) результаты применения оператора Кэнни при  $\sigma = 4$ ; (Снизу слева) изображение автомобильного колеса. (Снизу в центре) результаты применения оператора Кэнни при  $\sigma = 1$ . (Снизу справа) результаты применения оператора Робертса. Обратите внимание, что в верхней строке зеркальное отражение в верхней части изображения мешает детектору краев сформировать границу хромированного обода фары. В нижней строке область тени от автомобиля сливается с образом шины, который также сливается с крылом. Ни шина, ни спицы колеса удовлетворительно не выделяются.

Детектор краев Кэнни представлен далее в виде алгоритма 10.5. Этот алгоритм генерирует тонкие фрагменты контуров изображения. Алгоритм управляется тремя параметрами: параметром сглаживания  $\sigma$  и двумя пороговыми значениями интенсивности.

Сначала изображение сглаживается гауссовым фильтром с размахом  $\sigma$ . Затем в каждом пикселе сглаженного изображения вычисляется величина и направление градиента. Направление градиента используется для утончения контуров посредством удаления пикселей, в которых величина градиента не превышает величины градиента в двух соседних пикселях по любую сторону от данного пиксела вдоль направления градиента. Этот метод называется *немаксимальным подавлением* (nonmaximum suppression) и представляет собой полезную операцию для применения в любом детекторе краев, который должен генерировать утонченные контуры. Два восьмисвязных соседа центрального пиксела  $[x, y]$ , с которыми сравнивается величина градиента центрального пиксела, определяются с учетом округления значения направления градиента. При этом выбирается по одному соседу с каждой стороны от центрального пиксела. После обработки пикселей по величине градиента производится прослеживание контуров, состоящих из пикселей с большими значениями градиента. На окончательном этапе накопления контуров выполняется последовательное прослеживание непрерывных сегментов контуров. Прослеживание контуров начинается только на краевых пикселях, в которых значение градиента превышает верхнее пороговое значение. Однако, после начала прослеживания, контур может проходить через пиксели, в которых величина градиента меньше верхнего, но больше нижнего порогового значения. Обычно нижнее пороговое значение выбирается примерно наполовину меньше верхнего.

Иногда сегменты границ довольно точно соответствуют областям изображений. В таких случаях области изображения можно обнаружить по данным о выделенных границах. Примеры таких изображений приведены на рис. 10.16 и 10.17. Выделенные сегменты можно проанализировать далее, сегментируя множество граничных пикселей на прямолинейные отрезки, дуги окружностей и т. п. Например, граница прямоугольного здания может быть разбита на четыре прямолинейных отрезка. Отрезки прямых можно обнаружить с помощью преобразования Хафа или непосредственным подбором параметрической модели прямой.

### 10.3.3 Группировка согласующихся соседних контурных фрагментов в кривые

Алгоритм прослеживания границ из разд. 10.3.1 в качестве входных данных принимает маркированное изображение, которое служит представлением множества областей. В процессе построчной обработки изображения этот алгоритм прослеживает границу каждой области. В алгоритме принято допущение о том, что каждая граница ограничивает замкнутую область. Поэтому считается, что нет ни одной точки, в которой граница могла бы разделиться на две



или более частей. Допустим, что в качестве входных данных используется не маркированное изображение, а маркированное контурное изображение, на котором значения 1 соответствуют контурным пикселям, а 0 — всем остальным пикселям. В таком случае задача прослеживания сегментов краев (контуров) становится более сложной. Теперь контурные пиксели не обязательно ограничивают замкнутые области. Сегменты контуров состоят из связанных контурных пикселей. Сегменты могут начинаться в вершине, угловой точке или стыке и заканчиваться в другой вершине, угловой точке или стыке. Сегменты не содержат промежуточных стыков или углов.

На рис. 10.18 представлен пример маркированного контурного изображения. Пиксел [3, 3] является *пикселом стыка* (junction pixel), в котором соединяются три различных прямолинейных контурных сегмента. Пиксел [5, 3] — это угловой пиксел, который может рассматриваться в качестве концевой точки сегмента, если в данной задаче разрешается завершать сегменты в угловых точках. В алгоритме для прослеживания подобных сегментов требуется предусмотреть выполнение следующих операций:

1. прослеживание нового сегмента,
2. добавление внутреннего пикселя к сегменту,
3. завершение сегмента,
4. поиск стыка, и
5. поиск угла.

### Упражнение 10.7

Рассмотрим этап прослеживания контуров в алгоритме детектора краев Кэнни. Когда контур на изображении прослеживается вдоль пикселей с большим значением градиента, может быть, стоило бы выбирать следующий возможный пиксел только среди двух соседей, перпендикулярных к направлению градиента. Имеет ли смысл так делать? Поясните свой ответ с использованием примеров.



**Рис. 10.16.** Во многих случаях образы символов характеризуются высоким контрастом с фоном и их легко обнаружить. Представленные результаты получены с помощью оператора Кэнни. (Два изображения слева) символы написаны чернилами на бумаге. (Два изображения справа) выветрившаяся надпись на кирпичной стене. (С любезного разрешения John Weng.)

## Вычисление тонких связных сегментов контуров на входном изображении.

$I[x, y]$  : входное полутоновое изображение;

$\sigma$  : размах, используемый при гауссовском сглаживании;

$E[x, y]$  : выходное бинарное изображение;

$IS[x, y]$  : сглаженное полутоновое изображение;

$Mag[x, y]$  : величина градиента;

$Dir[x, y]$  : направление градиента;

$T_{low}$  — нижнее пороговое значение интенсивности;

$T_{high}$  — верхнее пороговое значение интенсивности;

**procedure** Canny(  $I[]$ ,  $\sigma$ );

```
{
     $IS[]$  = изображение  $I[]$ , сглаженное гауссовским фильтром  $G_{\sigma}(x, y)$ ;
    Оператор Робертса для вычисления  $Mag[x, y]$  и  $Dir[x, y]$  по  $IS[]$ ;
    Suppress_Nonmaxima( $Mag[]$ ,  $Dir[]$ ,  $T_{low}$ ,  $T_{high}$ );
    Edge_Detect( $Mag[]$ ,  $T_{low}$ ,  $T_{high}$ ,  $E[]$ );
}
```

**procedure** Suppress\_Nonmaxima( $Mag[]$ ,  $Dir[]$ );

```
{
    define +Del[4] = (1,0), (1,1), (0,1) (-1,1);
    define -Del[4] = (-1,0), (-1,-1), (0,-1) (1,-1);
    for x := 0 to MaxX-1;
        for y := 0 to MaxY-1;
            {
                direction := ( $Dir[x, y]$  +  $\pi/8$ ) modulo  $\pi/4$ ;
                if ( $Mag[x, y] \leq Mag[(x, y)+Del[direction]]$ )
                    then  $Mag[x, y] := 0$ ;
                if ( $Mag[x, y] \leq Mag[(x, y)-Del[direction]]$ )
                    then  $Mag[x, y] := 0$ ;
            }
}
```

**procedure** Edge\_Detect( $Mag[]$ ,  $T_{low}$ ,  $T_{high}$ ,  $E[]$ );

```
{
    for x := 0 to MaxX - 1;
        for y := 0 to MaxY - 1;
            {
                if ( $Mag[x, y] \geq T_{high}$ ) then Follow_Edge(x, y,  $Mag[]$ ,
                     $T_{low}$ ,  $T_{high}$ ,  $E[]$ );
            }
}
```

**procedure** Follow\_Edge(x, y,  $Mag[]$ ,  $T_{low}$ ,  $T_{high}$ ,  $E[]$ );

```
{
     $E[x, y] := 1$ ;
    while  $Mag[u, v] > T_{low}$  для некоторого 8-связного соседа  $[u, v]$ 
    пиксела  $[x, y]$ 
        {
             $E[u, v] := 1$ ;
             $[x, y] := [u, v]$ ;
        }
}
```



**Рис. 10.17.** Изображение здания и результаты применения оператора Кэнни при  $\sigma = 1$  и  $\sigma = 2$ . Несколько объектов выделены очень хорошо, но также присутствует ряд теневых областей.

	1	2	3	4	5
1	1	0	0	0	1
2	0	1	0	1	0
3	0	0	1	0	0
4	0	0	1	0	0
5	0	0	1	1	1

**Рис. 10.18.** Маркированное контурное изображение, содержащее стык трех отрезков в пикселе [3, 3] и возможную угловую точку в пикселе [5, 3].

### Упражнение 10.8. Измерения вдоль контуров, выделенных детектором Кэнни

Выполните следующий эксперимент. Найдите программу с реализацией детектора краев Кэнни или какую-нибудь графическую утилиту, содержащую соответствующую функцию. Возьмите несколько плоских объектов с точно параллельными краями, например, бритвенные лезвия, и несколько округлых объектов (например, карандаши). Получите несколько изображений этих объектов в различных ориентациях. Используйте максимально возможное разрешение. Примените детектор краев Кэнни и изучите качество полученных краев, в том числе оцените постоянство расстояния между параллельными краями. Есть ли разница между измерениями на изображении расстояний между резкими краями лезвий и более плавными краями карандашей?

Как и при прослеживании границ, для хранения информации на каждом шаге процедуры группировки контуров целесообразно пользоваться эффективными структурами данных. Применяемые структуры данных похожи на структуры данных из алгоритма выделения границ. Вместо обработанных, текущих и ожидающих областей теперь используются обработанные, текущие и ожидающие сегменты контуров. Сегменты — это списки контурных точек, соответствующие прямолинейным или криволинейным линиям на изображении. Текущие сегменты хранятся в оперативной памяти. Для доступа к ним

применяется хэш-таблица. Обработанные сегменты сохраняются в дисковом файле. Занимаемое ими пространство в хэш-таблице освобождается. Обнаружение точек стыка и входящих в них сегментов выполняется в направлении сверху-слева, а исходящих сегментов — в направлении справа-вниз.

Будем полагать, что был реализован расширенный оператор проверки окрестности *pixeltype*. Этот оператор определяет, является ли заданный пиксел изолированной точкой, начальной точкой нового сегмента, внутренней точкой старого сегмента, концевой точкой старого сегмента, точкой стыка или углом. Если пиксел является концевой или внутренней точкой существующего сегмента, то оператор *pixeltype* также возвращает идентификатор данного сегмента. Если пиксел является угловой точкой или точкой стыка, то возвращается список (INLIST) идентификаторов входящих сегментов и список (OUTLIST) пикселов, представляющих исходящие сегменты. Процедура для прослеживания контуров на маркированном контурном изображении представлена в виде алгоритма 10.6. На рис. 10.19 показаны результаты применения этого алгоритма к маркированному контурному изображению на рис. 10.18.

Идентификатор сегмента	Длина	Список
1	3	(1, 1)(2, 2)(3, 3)
2	3	(1, 5)(2, 4)(3, 3)
3	3	(3, 3)(4, 3)(5, 3)
4	3	(5, 3)(5, 4)(5, 5)

**Рис. 10.19.** Результаты применения процедуры *edge\_track* к изображению на рис. 10.18. Точка (5, 3) рассматривается в качестве угловой точки. Если бы для завершения сегментов угловые точки не использовались, то сегмент 3 имел бы длину 5 и представлялся в виде списка ((3, 3)(4, 3)(5, 3) (5, 4)(5, 5)).

Здесь не будем подробно рассматривать детали хранения идентификаторов сегментов, являющихся входящими и исходящими сегментами стыка. Эта часть процедуры может быть очень простой в предположении, что каждый пиксел, смежный пикселу стыка, принадлежит какому-либо другому сегменту. В таком случае, если толщина сегментов превышает один пиксел, алгоритм будет обнаруживать большое количество мелких сегментов, которые в действительности не являются новыми прямолинейными отрезками. Чтобы избежать этого, можно применить к контурному изображению оператор связной обработки с целью удаления мелких компонент. Другой альтернативой может быть усовершенствование оператора *pixeltype*. Он мог бы анализировать большую окрестность и использовать эвристические правила, чтобы определять, является ли пиксел частью утолщенного текущего сегмента, или началом нового сегмента. Во многих задачах содержание таких эвристических правил определяется спецификой предметной области.

## Поиск отрезков прямых на бинарном контурном изображении S.

**S[R, C]** — входное маркированное контурное изображение.

**NLINES** — количество строк изображения.

**NPIXELS** — количество пикселей в строке.

**IDNEW** — идентификатор нового сегмента.

**INLIST** — список идентификаторов входящих сегментов, возвращаемый функцией `pixeltype`.

**OUTLIST** — список идентификаторов исходящих сегментов, возвращаемый функцией `pixeltype`.

```
procedure edge_track(S);
{
  IDNEW := 0;
  for R := 1 to NLINES
    for C := 1 to NPIXELS
      if S[R,C] ≠ фоновый пиксел
        {
          NAME := address(R, C);
          NEIGHB := neighbors(R, C);
          T := pixeltype(R,C,NEIGHB,ID,INLIST,OUTLIST);
          case
            T = изолированная точка : next;
            T = начальная точка нового сегмента:
            {
              IDNEW := IDNEW + 1;
              make_new_segment(IDNEW,NAME);
            };
            T = внутренняя точка старого сегмента : add(ID,NAME);
            T = конечная точка старого сегмента :
            {
              add(ID,NAME);
              output(ID); free(ID)
            };
            T = точка стыка или угловая точка:
            for каждый идентификатор ID в списке INLIST
            {
              add(ID,NAME);
              output(ID);
              free(ID);
            };
            for каждый пиксел в OUTLIST
            {
              IDNEW := IDNEW + 1;
              make_new_segment(IDNEW,NAME);
            };
          };
        }
}
```

### 10.3.4 Преобразование Хафа для обнаружения прямых и дуг окружностей

*Преобразование Хафа* (Hough transform) — это метод обнаружения прямых и кривых линий на полутоновых или цветных изображениях. Метод позволяет указать параметры семейства кривых и обеспечивает поиск на изображении множества кривых заданного семейства. В данном разделе описывается преобразование Хафа и его применение для поиска на изображении прямолинейных отрезков и дуг окружностей.

---

#### Упражнение 10.9. Классификация пикселей для прослеживания контуров

Реализуйте оператор `pixelture` для классификации текущего пиксела по результатам обработки его окрестности размерами  $3 \times 3$ . Возможные варианты классификации: изолированный пиксел, начало или конец сегмента, внутренний пиксел сегмента, точка стыка или угловая точка.

**Преобразование Хафа.** В алгоритме преобразования Хафа используется аккумуляторный массив, размерность которого соответствует количеству неизвестных параметров в уравнении семейства искомым кривых. Например, при обнаружении прямых, описываемых уравнением  $y = mx + b$ , для каждой прямой необходимо вычислить значения двух параметров  $m$  и  $b$ . При обнаружении прямых два измерения аккумуляторного массива могли бы соответствовать дискретным значениям  $m$  и дискретным значениям  $b$ . При этом в аккумуляторном массиве в элементах  $A[M, B]$  накапливаются значения, указывающие на вероятность наличия на изображении прямой  $y = mx + b$ . Индексы  $M$  и  $B$  соответствуют дискретным значениям  $m$  и  $b$ .

Аккумуляторный массив  $A$  используется в алгоритме Хафа для проверки каждого пиксела изображения и его окрестности. Определяется, присутствует ли в данном пикселе достаточно выраженный край. Если присутствует, то вычисляются параметры искомой кривой, проходящей через данный пиксел. В примере с прямой, описываемой уравнением  $y = mx + b$ , потребовалось бы оценить параметры  $m$  и  $b$  для прямой, проходящей через данный пиксел, если характеристика мощности края (например, величина градиента) в данном пикселе имеет достаточно большое значение. После оценки параметров прямой в данном пикселе они дискретизируются для получения соответствующих значений  $M$  и  $B$  и значение элемента массива  $A[M, B]$  увеличивается. В некоторых реализациях увеличение выполняется на единицу, а в других — на величину мощности края в обработанном пикселе. После обработки всех пикселей выполняется поиск локальных максимумов в аккумуляторном массиве. Точки локальных максимумов соответствуют параметрам наиболее вероятных прямых на изображении.

Аккумуляторный массив позволяет определить параметры бесконечно протяженных прямых или кривых линий, но с его помощью нельзя определить,

где именно на изображении начинаются и заканчиваются отрезки этих линий. Для получения этой информации можно завести еще одну структуру данных **PTLIST**. Элемент **PTLIST[M, V]** содержит список координат всех пикселей, которые внесли вклад (проголосовали) в значение элемента аккумуляторного массива **A[M, V]**. По содержанию этих списков можно найти присутствующие на изображении отрезки или сегменты кривых.

Выше было приведено общее описание метода Хафа. В нем не рассмотрены многие детали, важные для программной реализации метода. Перейдем теперь к более подробному обсуждению алгоритмов для обнаружения прямых и окружностей.

**Обнаружение отрезков прямых** Уравнение прямой  $y = mx + b$  не подходит для представления вертикальных прямых. Удобнее выбрать уравнение прямой в виде  $d = x \cos \theta + y \sin \theta$ , где  $d$  — это длина перпендикуляра к прямой, опущенного из начала координат, а  $\theta$  — угол между этим перпендикуляром и горизонтальной осью. Будем пользоваться этим уравнением прямой. Преобразуем его в растровую систему координат, оси которой направлены вдоль строк ( $r$ ) и столбцов ( $c$ ) изображения. Так как координата  $c$  соответствует  $x$ , а координата  $r$  — координате  $(-y)$ , то уравнение прямой принимает вид

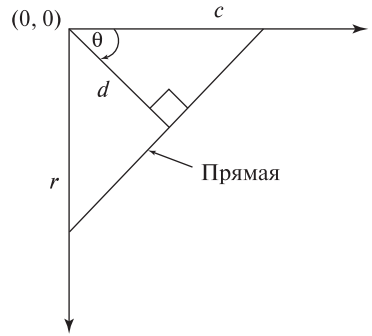
$$d = c \cos \theta - r \sin \theta \quad (10.17)$$

где  $d$  — длина перпендикуляра к прямой, опущенного из начала растровой системы координат (левый верхний угол изображения), а  $\theta$  — угол между этим перпендикуляром и осью  $c$ . Геометрический смысл параметров уравнения прямой поясняется на рис. 10.20. Допустим, что перпендикуляр пересекается с прямой в точке  $[50, 50]$ , а угол  $\theta = 315^\circ$ . Тогда

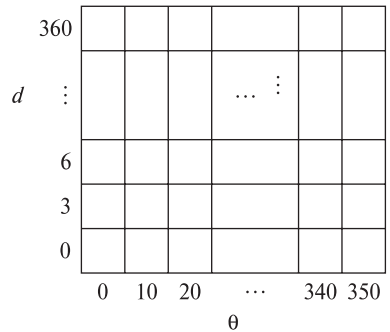
$$d = 50 \cos(315) - 50 \sin(315) = 50(.707) - 50(-.707) \approx 70$$

Индексы аккумуляторного массива **A** соответствуют дискретным значениям  $d$  и  $\theta$ . О'Горман (O'Gorman) и Кловс (Clowes) дискретизовали значения  $d$  с шагом 3 и значения  $\theta$  с шагом  $10^\circ$ . Эксперименты (1976) проводились с полутоновыми изображениями сцен, содержащих игрушки. Подобный дискретный аккумуляторный массив показан на рис. 10.21. Ниже в виде процедуры *accumulate\_lines* приведен алгоритм О'Гормана и Кловса для заполнения аккумуляторного массива **A** и массива списков **PTLIST**.

Алгоритм работает в двумерном координатном пространстве (строка, столбец). Функции *row\_gradient* и *column\_gradient* обрабатывают окрестности пикселей для оценки компонент градиента в направлении строк и столбцов. Функция *gradient* комбинирует две эти компоненты для получения величины градиента. Функция *atan2* — это стандартная функция из математической библиотеки, возвращающая по заданным координатным компонентам градиента угол в соответствующем квадранте. Будем полагать, что *atan2* возвращает значение в диапазоне от  $0^\circ$  до  $359^\circ$ . Во многих стандартных библиотеках эта функция возвращает значение угла в радианной мере, тогда его необходимо



**Рис. 10.20.** Параметры  $d$  и  $\theta$ , входящие в уравнение прямой  $d = -r \sin \theta + c \cos \theta$ .



**Рис. 10.21.** Аккумуляторный массив для обнаружения прямых на изображениях с размерами  $256 \times 256$  пикселей.

преобразовать в градусы. Если значение  $d$  отрицательно (например, при  $\theta = 135^\circ$ ), то расстояние между прямой и началом координат вычисляется как модуль  $d$ . Основные действия алгоритма поясняются на рис. 10.22. Отметим, что при использовании оператора градиента с окном  $3 \times 3$  прямые имеют ширину два пикселя.

Процедура *accumulate\_lines* представляет собой версию преобразования Хафа, разработанную О'Горманом и Кловсом. Преобразование Хафа обеспечивает заполнение аккумуляторного массива и массива списков, но не предусматривает стандартного метода выделения отрезков прямых. Поэтому О'Горман и Кловс разработали специальную процедуру *find\_lines*. Эта процедура представлена в виде алгоритма 10.8. Она иллюстрирует ряд проблем, возникающих на этапе выделения отрезков прямых.

Функция *pick\_greatest\_bin* возвращает максимальное значение из аккумуляторного массива, присваивая параметрам **DQ** и **ТНЕТАQ** соответствующие дискретные значения  $d$  и  $\theta$ . Функция *reorder* упорядочивает список точек в элементе массива по координате столбца при  $\theta < 45$  или  $\theta > 135$  и по координате строки при  $45 \leq \theta \leq 135$ . Предполагается, что в массивах **D** и **ТНЕТА** для пикселей изображения содержатся дискретные значения **D** и **ТНЕТА**, вычисленные на этапе заполнения аккумуляторного массива. Аналогично, в массиве **GRADIENT** должны находиться вычисленные значения величины



градиента. Перечисленные массивы могут храниться в виде вспомогательных изображений.

Процедура *merge* объединяет список точек соседнего пиксела со списком точек для данного пиксела. При этом сохраняется пространственное упорядочение точек. Процедура *set\_to\_zero* обнуляет элемент аккумуляторного массива, чтобы он не был найден повторно. Наконец, процедура *create\_segments* просматривает окончательный упорядоченный список точек и ищет в нем промежутки длиннее одного пиксела. Эта процедура сохраняет множество отрезков прямых, разделенных промежутками. Для большей точности списки точек можно аппроксимировать отрезками с помощью метода наименьших квадратов. Важно учитывать, что преобразование Хафа может обнаружить посторонние разрывные или фиктивные линии, например образованные образами россыпи камней или тенями от ветвей деревьев на дороге.

### Упражнение 10.10

Это упражнение основано на работе [Kasturi и др., 1990]. Задача заключается в обнаружении строк текста с помощью преобразования Хафа. При выполнении упражнения воспользуйтесь какими-либо существующими программами или напишите собственные. (а) Напечатайте несколько строк текста в различных направлениях и получите соответствующее бинарное изображение. Добавьте на изображение несколько посторонних образов, например, пятен или кривых. (б) Примените маркировку связных компонент и вычислите центры тяжести всех компонент, у которых описывающие прямоугольники похожи на области символов. (в) Передайте множество вычисленных центров тяжести в процедуру обнаружения прямых методом преобразования Хафа и выясните, насколько хорошо удается обнаружить строки текста.

*Обнаружение окружностей.* Преобразование Хафа может применяться для обнаружения окружностей и других параметрических кривых. Стандартное уравнение окружности содержит три параметра. Если точка  $[R, C]$  лежит на окружности, то градиент в точке  $[R, C]$  должен быть направлен к центру окружности (см. рис. 10.23). Поэтому, если дана точка  $[R, C]$ , известен радиус  $d$ , и в точке  $[R, C]$  вычислено направление к центру окружности, то можно определить координаты центра окружности. Радиус  $d$ , вертикальная координата центра  $r_o$  и горизонтальная координата центра  $c_o$  представляют собой три параметра, используемые при обнаружении окружностей с помощью преобразования Хафа. В растровой системе координат окружность описывается уравнениями

$$r = r_o + d \sin \theta \quad (10.18)$$

$$c = c_o - d \cos \theta \quad (10.19)$$

**Накопление данных об отрезках прямых на полутоновом изображении S в аккумуляторном массиве A.**

**S**[R, C] — входное полутоновое изображение.

**NLINES** — количество строк на изображении.

**NPIXELS** — количество пикселей в строке изображения.

**A**[DQ, THETAQ] — аккумуляторный массив.

**DQ** — дискретное расстояние от прямой до начала координат.

**THETAQ** — дискретный угол между направлением строк и перпендикуляром к прямой, опущенным из начала координат.

```

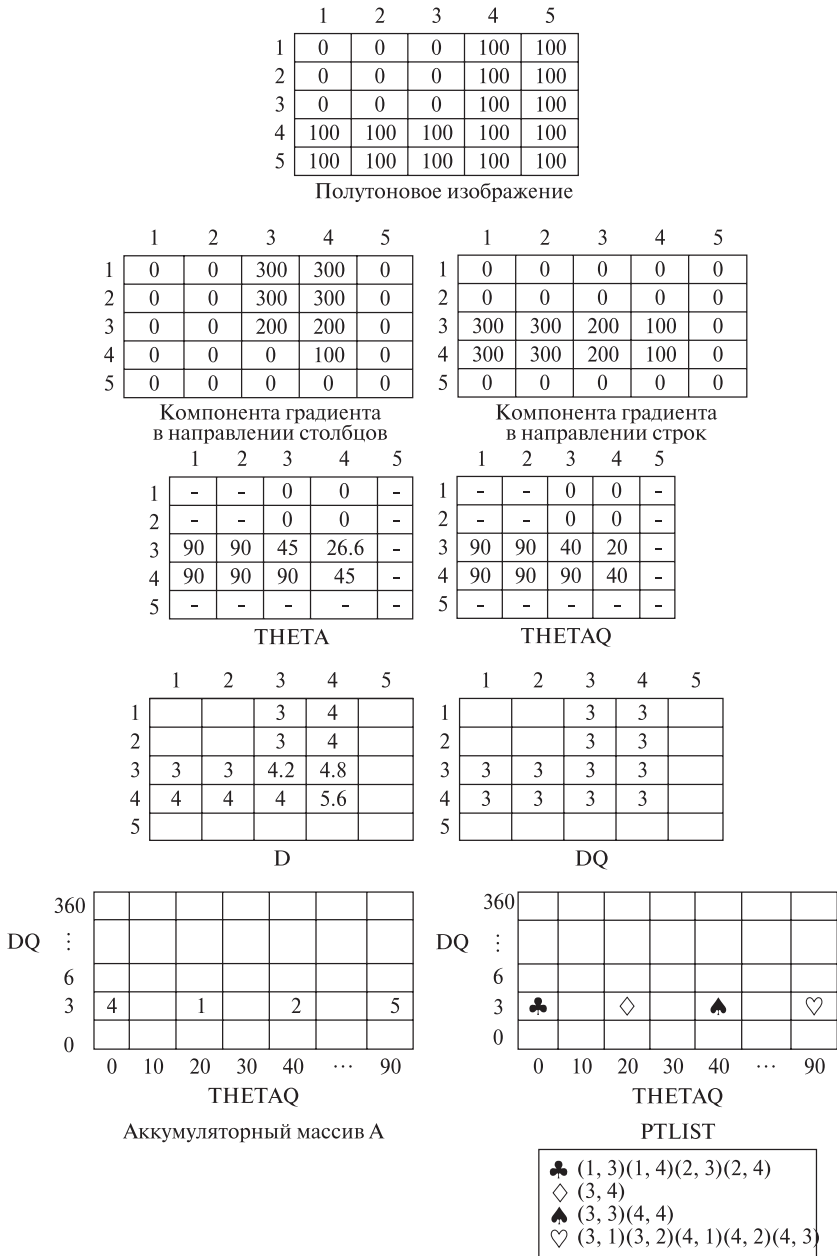
procedure accumulate_lines(S,A);
{
  A := 0;
  PTLIST := NIL;
  for R := 1 to NLINES
    for C := 1 to NPIXELS
      {
        DR := row_gradient(S,R,C);
        DC := col_gradient(S,R,C);
        GMAG := gradient(DR,DC);
        if GMAG > gradient_threshold
          {
            THETA := atan2(DR,DC);
            THETAQ := quantize_angle(THETA);
            D := abs(C*cos(THETAQ) - R*sin(THETAQ));
            DQ := quantize_distance(D);
            A[DQ,THETAQ] := A[DQ,THETAQ] + GMAG;
            PTLIST(DQ,THETAQ) :=
              append(PTLIST(DQ,THETAQ),[R,C])
          }
      }
}

```

**Алгоритм 10.7.** Преобразование Хафа для обнаружения прямых.

На основе этих уравнений можно записать алгоритм заполнения аккумуляторного массива для обнаружения окружностей. Этот алгоритм *accumulate\_circles* представлен далее.

Эту процедуру легко модифицировать для учета величины градиента, как это делалось в случае поиска отрезков прямых. Результаты применения этой процедуры для обработки изображения чертежа приведены на рис. 10.24.



**Рис. 10.22.** Результаты применения процедуры *accumulate* к простому полутоновому изображению с использованием масок Превитт. В этом примере значения корректных данных не намного превышают некорректные значения. Но на реальных изображениях с длинными отрезками различие будет намного более явным.

**Поиск списков точек, соответствующих отдельным прямолинейным отрезкам.**

**A[DQ, THETAQ]** — аккумуляторный массив, заполненный данными о прямых на изображении.

**DQ** — дискретное расстояние от прямой до начала координат.

**THETAQ** — дискретный угол между направлением строк и перпендикуляром к прямой, опущенным из начала координат.

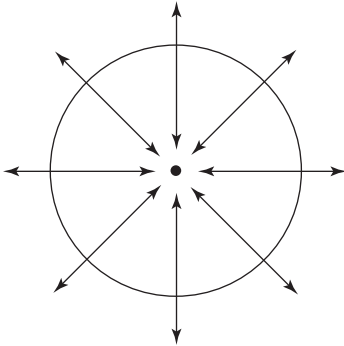
```

procedure find_lines;
{
  V := pick_greatest_bin(A,DQ,THETAQ);
  while V > value_threshold
  {
    list_of_points := reorder(PTLIST[DQ,THETAQ]);
    for каждая точка [R, C] в списке list_of_points
      for каждый соседний пиксел [R',C'] для [R, C],
        не входящий в list_of_points
      {
        DPRIME := D[R',C'];
        THETAPRIME := THETA[R',C'];
        GRADPRIME := GRADIENT[R',C'];
        if GRADPRIME > gradient_threshold
          and abs(THETAPRIME-THETAQ) ≤ 10
        then {
          merge(PTLIST[DQ,THETAQ],PTLIST[DPRIME,
            THETAPRIME]);
          set_to_zero[A,DPRIME,THETAPRIME];
        }
      }
    final_list_of_points := PTLIST[DQ,THETAQ];
    create_segments(final_list_of_points);
    set_to_zero[A,DQ,THETAQ];
    V := pick_greatest_bin[A,DQ,THETAQ];
  }
}

```

**Алгоритм 10.8.** Процедура О'Гормана и Кловса для выделения отрезков прямых.

**Дополнения.** Метод преобразования Хафа может быть расширен для обнаружения любых кривых, аналитически описываемых с помощью уравнения вида  $f(x, a) = 0$ , где  $x$  обозначает точку изображения, а  $a$  — вектор параметров. Процедура заключается в следующем.



**Рис. 10.23.** Направление градиента в граничных точках окружности. Точки, в которых градиент направлен к центру окружности, учитываются при увеличении счетчика, соответствующего центру окружности.

**Накопление данных об окружностях на полутоновом изображении  $S$  в аккумуляторном массиве  $A$ .**

$S[R, C]$  – входное полутоновое изображение.

$NLINES$  – количество строк на изображении.

$NPIXELS$  – количество пикселей в строке.

$A[R, C, RAD]$  – аккумуляторный массив.

$R$  – вертикальная координата центра окружности (индекс строки).

$C$  – горизонтальная координата центра окружности (индекс столбца).

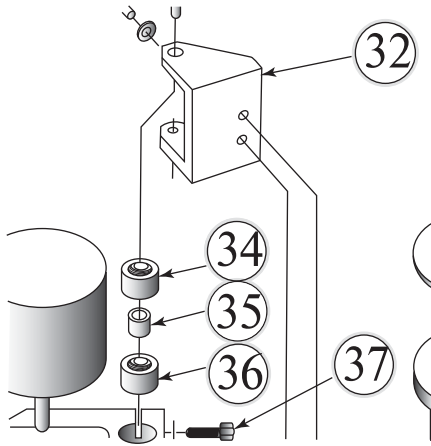
$RAD$  – радиус окружности.

```

procedure accumulate_circles(S,A);
{
  A := 0;
  PTLIST := 0;
  for R := 1 to NLINES
    for C := 1 to NPIXELS
      for каждое возможное значение радиуса RAD
        {
          THETA := compute_theta(S,R,C,RAD);
          R0 := R - RAD*cos(THETA);
          C0 := C + RAD*sin(THETA);
          A[R0,C0,RAD] := A[R0,C0,RAD]+1;
          PTLIST[R0,C0,RAD] :=
            append(PTLIST[R0,C0,RAD],[R,C])
        }
    }
}

```

**Алгоритм 10.9.** Преобразование Хафа для обнаружения окружностей.



**Рис. 10.24.** Фрагмент технического чертежа с окружностями, найденными с помощью преобразования Хафа. Они показаны оверлейным наложением окружностей, радиусы которых немного превышают радиусы исходных окружностей на изображении.

1. Инициализация аккумуляторного массива  $\mathbf{A}[\mathbf{a}]$  нулевыми значениями.
2. Для каждого краевого пиксела  $\mathbf{x}$  определяется такой вектор  $\mathbf{a}$ , что  $f(\mathbf{x}, \mathbf{a}) = 0$  и выполняется увеличение значения соответствующего элемента массива  $\mathbf{A}[\mathbf{a}] := \mathbf{A}[\mathbf{a}] + 1$ .
3. Локальные максимумы в аккумуляторном массиве  $\mathbf{A}$  соответствуют вероятным кривым  $f$  на изображении.

Если вектор  $\mathbf{a}$  содержит  $m$  параметров и каждый из этих параметров принимает  $M$  дискретных значений, то временная сложность алгоритма составляет  $O(M^{m-2})$ . Преобразование Хафа было обобщено для обнаружения произвольных фигур, заданных последовательностью граничных точек [Ballard, 1981]. Этот метод известен под названием *обобщенного преобразования Хафа* (generalized Hough transform).

**Определитель прямых Барнса.** Существует ряд гибридных методов, в которых используются некоторые свойства преобразования Хафа. Определитель прямых Барнса [Burns и др., 1986] был разработан для обнаружения прямых линий на сложных изображениях естественных сцен. В методе Барнса выполняются следующие основные операции:

1. Вычисление величины и направления градиента для каждого пиксела изображения.
2. Точкам с достаточно большой величиной градиента присваиваются две метки, соответствующие двум различным дискретным значениям направления градиента (например, при восьмиразрядной дискретизации первое дискретное значение берется из диапазонов 0-44, 45-90, 91-134,

и т. д., а второе — из диапазонов  $-22-22$ ,  $23-67$ ,  $68-112$ , и т. д.). Результаты маркировки сохраняются в виде двух маркированных изображений.

**3.** Поиск связанных компонент на каждом маркированном изображении и вычисление длины отрезков каждой компоненты

- Каждый пиксел принадлежит двум компонентам, по одной на каждом маркированном изображении.
- Каждый пиксел используется для голосования за свою *длиннейшую* компоненту.
- Счетчик каждой компоненты равен количеству проголосовавших за нее пикселов.
- Выполняется выбор компонент (отрезков прямых) с наибольшими значениями счетчиков.

В определителе прямых Барнса сочетаются преимущества двух мощных алгоритмов: преобразования Хафа и алгоритма маркировки связанных компонент. В этом алгоритме два различных варианта дискретизации применяются для устранения проблем дискретизации, вынудивших О'Гормана и Кловса выполнять обработку соседних элементов.

Проблемы дискретизации влияют на любой определитель прямых, который оценивает направление по малой окрестности данного пиксела. Растровые отрезки не являются прямолинейными. Диагональные отрезки в действительности состоят из последовательности горизонтальных и вертикальных сегментов. Если при вычислении угла используется слишком маленькая окрестность, то вместо длинного диагонального отрезка будет обнаружено большое количество мелких горизонтальных и вертикальных отрезков.

При практическом использовании определитель прямых Барнса (и любой другой определитель прямых на основе оценки направления) может разбивать отрезки, которые человеком воспринимались бы как один связный отрезок.

---

**Упражнение 10.11.** Сравнение алгоритмов Барнса и Хафа

Разработайте программы с реализацией преобразования Хафа и определителя прямых Барнса. Сравните результаты работы этих программ при обработке естественных изображений, содержащих достаточное количество прямолинейных отрезков.

---

**Упражнение 10.12.** Обнаружение прямых

Напишите программу, выполняющую поиск прямых на полутоновом изображении **I** на основе следующего подхода:

```

for все пикселы изображения  $I[R,C]$ 
  {
  Вычислить величину и направление градиента  $G_{mag}$  и  $G_{dir}$ 
  if  $G_{mag} > \text{threshold}$ 
    then поместить  $[G_{mag}, G_{dir}]$  в множество  $H$ 
  }
Обнаружить кластеры на множестве  $H$ ;

```

Наиболее вероятным прямолинейным отрезкам на изображении  $I$  должны соответствовать «существенные» кластеры.

## 10.4 ПОДБОР МОДЕЛЕЙ СЕГМЕНТОВ

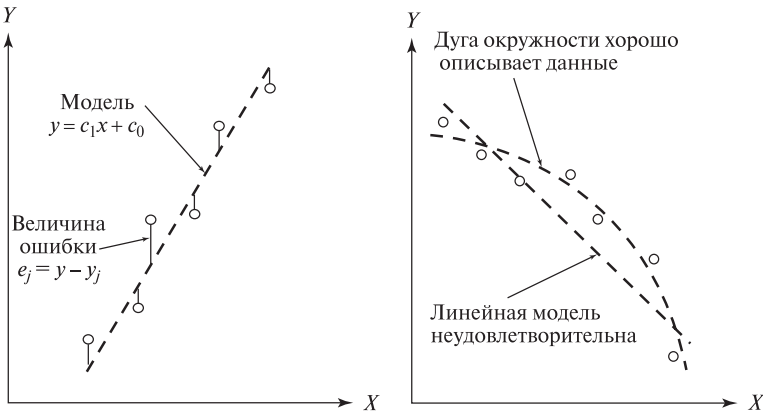
Математические модели для аппроксимации (приближенного представления) данных не только помогают выявить важные структурные свойства данных, но также могут использоваться для их представления в процессе дальнейшей обработки. Например, для аппроксимации контурных пикселов изображения здания может использоваться модель прямолинейного отрезка, а для представления данных, соответствующих стороне здания — модель плоскости. Существуют удобные математические модели окружностей, цилиндров и многих других фигур.

Далее будет рассмотрен *метод наименьших квадратов*, позволяющий определить параметры математической модели, которая наилучшим образом аппроксимирует наблюдаемые данные. Исходные данные могут быть получены с помощью одного из описанных выше алгоритмов сегментации на основе границ или областей. Например, уже отмечалось, что для подтверждения гипотезы о наличии на изображении прямой, соответствующей элементу  $A[THETAQ, DQ]$  в аккумуляторном массиве преобразования Хафа, могла бы применяться проверка на основе аппроксимации всех пикселов  $[r, c]$ , проголосовавших за этот элемент, с помощью модели прямой.

При использовании метода наименьших квадратов должен быть известен способ выбора из бесконечно большого количества моделей одной или нескольких моделей, которые следует пробовать подгонять к исходным данным. После выбора модели и определения ее параметров можно решить, удовлетворительно ли данная модель описывает данные. Если качество аппроксимации удовлетворительно, то это может означать факт обнаружения объекта некоторой заданной формы или возможность использования параметров модели в качестве компактного представления данных в процессе дальнейшей обработки.

**Подбор прямой** Метод наименьших квадратов рассмотрим на простом примере аппроксимации точек данных с помощью прямой. В качестве модели прямой выберем уравнение с двумя параметрами  $y = f(x) = c_1x + c_0$ . Предположим, что требуется проверить, лежат ли исходные точки  $\{(x_j, y_j), j = 1, n\}$





**Рис. 10.25.** (Слева) подгонка модели  $y = f(x)$  к шести точкам данных. (Справа) сравнение моделей прямой и окружности. Знаки остаточных ошибок показывают, что линейная модель систематически смещена относительно точек данных, а окружность — нет.

на одной прямой. Для этого определим параметры  $c_1$  и  $c_0$  линейной функции, наилучшим образом аппроксимирующей данные, и проверим, насколько близки исходные точки к найденной прямой. Для оценки близости точек данных к модели могут применяться несколько различных критериев. На рис. 10.25 представлена прямая, подобранная по шести точкам данных. Вероятно, при небольшом смещении прямой она все равно «достаточно хорошо» представляла бы исходные точки. В Определении 74 приведен критерий наименьших квадратов для определения наилучшей аппроксимирующей прямой.

**74 Определение. Критерий наименьших квадратов.** В качестве величины, характеризующей согласование модели  $y = f(x)$  с множеством  $n$  исходных точек  $\{(x_j, y_j), j = 1, n\}$ , может использоваться сумма квадратов отклонений

$$LSE = \sum_{j=1}^n (f(x_j) - y_j)^2$$

Наилучшей моделью  $y = f(x)$  считается модель с такими параметрами, при которых достигается минимум суммы квадратов отклонений  $LSE$ .

**75 Определение. Среднеквадратическая ошибка** (root-mean-square error, RMSE) равна средней разности между исходными данными и значениями функции модели:

$$RMSE = \left[ \sum_{j=1}^n (f(x_j) - y_j)^2 / n \right]^{1/2}$$

**Таблица 10.1.** Применение метода наименьших квадратов для аппроксимации точек данных, вычисленных по формуле  $y = 3x - 7$  с добавлением шума. Была подобрана модель  $y = 2.971x - 6.962$ .

Точки данных						
$(x_j, y_j)$	(0.0, -6.8)	(1.0, -4.1)	(2.0, -1.1)	(3.0, 1.8)	(4.0, 5.1)	(5.0, 7.9)
Невязки $y - y_j$ :	-0.162	0.110	0.081	0.152	-0.176	-0.005

Отметим, что при подборе прямой разности, стоящие под знаком суммы в этом выражении, не равны евклидову расстоянию между точками данных и прямой. Они равны расстояниям между исходными точками и подобранной прямой вдоль направления, параллельного оси  $OY$  (см. рис. 10.25).

**76 Определение. Критерий максимальной ошибки.** В качестве величины, характеризующей согласование модели  $y = f(x)$  с множеством  $n$  исходных точек  $\{(x_j, y_j), j = 1, n\}$ , может использоваться максимальное отклонение

$$MAXE = \max(\{|f(x_j) - y_j|\}_{j=1,n})$$

Отметим, что эта величина определяется точкой, наиболее отклоняющейся от модели, в то время как среднеквадратическая ошибка зависит от качества аппроксимации всех исходных точек.

**Аналитические выражения параметров моделей** Критерий наименьших квадратов популярен по двум причинам. Во-первых, его логично применять в условиях гауссовского шума и, во-вторых, данный критерий позволяет достаточно просто получить аналитическое выражение параметров наилучшей модели. Сначала запишем аналитическое выражение для параметров наилучшей аппроксимирующей прямой. Выражения параметров других моделей выводятся аналогичным образом.

Запишем выражение ошибки аппроксимации при подборе модели прямой по критерию наименьших квадратов. Отметим, что исходные точки данных  $x_j, y_j$  в это выражение входят в качестве констант:

$$LSE = \varepsilon(c_1, c_0) = \sum_{j=1}^n (c_1 x_j + c_0 - y_j)^2 \quad (10.20)$$

Функция ошибки  $\varepsilon$  является непрерывной неотрицательной функцией двух переменных  $c_1$  и  $c_0$ . Эта функция имеет глобальный минимум в точке  $(c_1, c_0)$ , в которой выполняются равенства  $\partial\varepsilon/\partial c_1 = 0$  и  $\partial\varepsilon/\partial c_0 = 0$ . Вычислим эти частные производные с использованием выражения (10.20) и, учитывая тот факт, что производная суммы функций равна сумме производных каждой

функции.

$$\partial\varepsilon/\partial c_1 = \sum_{j=1}^n 2(c_1 x_j + c_0 - y_j)x_j = 0 \quad (10.21)$$

$$= 2 \left( \sum_{j=1}^n x_j^2 \right) c_1 + 2 \left( \sum_{j=1}^n x_j \right) c_0 - 2 \sum_{j=1}^n x_j y_j \quad (10.22)$$

$$\partial\varepsilon/\partial c_0 = \sum_{j=1}^n 2(c_1 x_j + c_0 - y_j) = 0 \quad (10.23)$$

$$= 2 \left( \sum_{j=1}^n x_j \right) c_1 + 2 \sum_{j=1}^n c_0 - 2 \sum_{j=1}^n y_j \quad (10.24)$$

Эти уравнения можно записать в матричной форме. Для определения параметров наилучшей аппроксимирующей прямой необходимо решить эти уравнения. В общем случае полиномиальной аппроксимации (когда исходные данные аппроксимируются с помощью многочленов, а не прямых) получается набор уравнений характерного вида, называемых *нормальными уравнениями*.

$$\begin{bmatrix} \sum_{j=1}^n x_j^2 & \sum_{j=1}^n x_j \\ \sum_{j=1}^n x_j & \sum_{j=1}^n 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_0 \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^n x_j y_j \\ \sum_{j=1}^n y_j \end{bmatrix} \quad (10.25)$$

### Упражнение 10.13. Построение прямой по трем точкам

С помощью уравнений (10.25) вычислите параметры  $c_1$  и  $c_0$  наилучшей аппроксимирующей прямой для точек  $(0, -7)$ ,  $(2, -1)$  и  $(4, 5)$ .

### Упражнение 10.14. Нормальные уравнения

(а) Запишите в матричной форме уравнения для определения 4 параметров аппроксимирующего кубического полинома  $c_3 x^3 + c_2 x^2 + c_1 x + c_0$  для точек данных  $(x_j, y_j)$ ,  $j = 1, n$ . (б) Учитывая структуру полученных элементов матриц, попробуйте определить вид матрицы для аппроксимирующего полинома четвертой степени.

**Эмпирическая интерпретация ошибок** Эмпирическая интерпретация критерия ошибок и отдельных ошибок в задачах машинного зрения часто имеет очевидный смысл. Например, результаты аппроксимации могут признаваться удовлетворительными, если все исходные точки лежат на расстоянии одного-двух пикселей от подобранной модели. Можно представить задачу, в которой в управляемой среде формируются двумерные изображения объектов с плоскими гранями. Для оценки ошибок можно было бы вычислить разброс найденных краевых точек относительно идеальной прямой. Отдельные точки

данных, которые лежат далеко от аппроксимирующей прямой (такие точки называются *выбросами*), могут соответствовать ошибочно обнаруженным признакам, или дефектам объекта, или потребовать поиска другого объекта или подбора другой модели. В таких случаях, возможно, целесообразно удалить выбросы из набора точек данных и затем повторить подбор модели, чтобы на ее параметры не влияли некорректные точки. Исходные точки данных интерпретируются относительно обновленной модели. При использовании аппроксимирующих моделей для сегментации кривых обычно удаляются крайние точки сегментов, так как они наиболее вероятно могут принадлежать другим объектам или кривым.

**Статистическая интерпретация ошибок\*** Для интерпретации ошибок можно пользоваться формальной статистической гипотезой. Обычно предполагается, что наблюдаемое значение  $y_j$  можно представить в виде суммы модельного значения  $f(x_j)$  и гауссовского шума. Шум описывается нормальным распределением  $N(0, \sigma)$ , где  $\sigma$  определяется по результатам анализа ошибок измерений (возможно, эмпирическим путем, как было описано выше). Также предполагается, что шумовые составляющие различных измерений независимы друг от друга. Следовательно, переменная  $S_{sq} = \sum_{j=1}^n ((f(x_j) - y_j)^2) / \sigma^2$  описывается распределением  $\chi^2$ , вероятности которого можно вычислить по формуле или найти в таблице. Для прямой распределение  $\chi^2$  будет иметь  $(n-2)$  степени свободы, так как два параметра прямой оцениваются по результатам  $n$  измерений. Например, гипотезу о корректной аппроксимации исходных данных моделью можно отбрасывать, если 95% вероятности  $\chi^2$  меньше наблюдаемого значения  $S_{sq}$ . Могут использоваться и другие доверительные уровни. Критерий «хи-квадрат» полезен не только для принятия/отклонения данной гипотезы, но и для выбора наиболее вероятной модели на множестве возможных альтернатив. Например, наряду с моделью прямой может рассматриваться параболическая модель. Отметим, что параболическая модель  $y = c_2x^2 + c_1x + c_0$  содержит три параметра, так что распределение  $\chi^2$  будет иметь  $(n-3)$  степеней свободы.

Интуитивно можно предположить, что ошибка измерений в точке  $j$  не всегда будет независима от ошибок измерений в точках  $j-1$  или  $j+1$ . Например, если некоторое изделие имеет производственный дефект, то на его изображении в области дефекта может быть искажена некоторая окрестность пикселей, и все они будут отклоняться от идеальной модели. Для проверки гипотезы независимости ошибок можно проверить знаки отклонений от модели у последовательности ошибочных значений. Эта проверка позволяет обнаружить систематическую ошибку, что, в свою очередь, может свидетельствовать о необходимости перейти к подбору другой модели. При наличии в измерениях случайного шума знаки ошибок должны меняться случайным образом и поэтому часто чередоваться. На рис. 10.25 (справа) показана смещенная линейная модель и несмещенная модель окружности. Знаки ошибок свидетельствуют о систематическом отклонении линейной модели. Более подробные сведения о

проверке статистических гипотез содержатся в книгах, перечисленных в списке литературы в конце главы.

---

### Упражнение 10.15. Аппроксимация трехмерных точек плоскостью

(а) Вычислите параметры  $a, b, c$  модели  $z = f(x, y) = ax + by + c$  для аппроксимации методом наименьших квадратов пяти точек  $(20, 10, 130)$ ,  $(25, 20, 130)$ ,  $(30, 15, 145)$ ,  $(25, 10, 140)$ ,  $(30, 20, 140)$ . (б) Повторите задание (а) после добавления случайного шума к каждой из трех координат всех пяти точек. Для определения добавляемых значений можно подбрасывать монету: если выпадет орел, добавляйте 1, а если решка, то вычитайте 1.

---

### Упражнение 10.16. Оптимальность оператора Превитт

Покажите, что градиентный оператор Превитт (этот оператор рассматривался в гл. 5) можно получить на основе аппроксимации плоскостью функции интенсивности в окрестности  $3 \times 3$  методом наименьших квадратов. Для вычисления градиента в точке  $I[x, y]$  подберите плоскость для девяти точек  $(x + \Delta x, y + \Delta y, I[x + \Delta x, y + \Delta y])$ , где  $\Delta x$  и  $\Delta y$  принимают значения  $-1, 0, +1$ . После определения параметров модели плоскости  $z = ax + by + c$ , лучше всего аппроксимирующей поверхность точек интенсивности, покажите, что две маски Превитт выполняют вычисление  $a$  и  $b$ .

**Проблемы аппроксимации** Перечислим ряд проблем, возникающих при аппроксимации данных измерений.

*Выбросы.* На значение среднеквадратической ошибки влияют все исходные точки. Поэтому при большом количестве выбросов подобранная модель может оказаться бесполезной. Параметры модели могут быть настолько далеки от идеальных, что найти и удалить выбросы не удастся. В таких случаях могут применяться робастные статистические методы. Этот вопрос рассматривается в работе [Boyer, Mirza, Ganguly, 1994] (см. список литературы в конце главы).

*Определение отклонения.* Выше для оценки отклонения исходных точек от модели использовалось расстояние вдоль оси  $OY$ , а не обычное геометрическое расстояние. При такой оценке отклонения метод наименьших квадратов не обязательно даст кривую или поверхность, лучше всего аппроксимирующую данные в геометрическом пространстве. Для пояснения этой проблемы рассмотрим крайнюю правую точку на рис. 10.25 (справа). Геометрически эта точка близка к окружности, но ее отклонение от окружности вдоль оси  $OY$  довольно велико. Этот эффект становится даже более явным при подборе сложных поверхностей для аппроксимации трехмерных точек. К сожалению, использование геометрического расстояния для оценки отклонений усложняет вычисления. При подборе прямой, близкой к вертикальной, вместо метода наименьших квадратов может быть удобнее использовать метод вычисления

главной оси, рассмотренный в гл. 3. При вычислении главной оси производится минимизация геометрического расстояния между прямой и точками.

*Нелинейная оптимизация.* Аналитическое выражение параметров модели удается получить не для всех моделей. В подобных случаях для минимизации критерия ошибок можно воспользоваться численным методом, выполняющим поиск наилучших значений в пространстве параметров. Для этой цели могут применяться градиентные методы поиска, а в некоторых задачах даже поиск методом полного перебора. По этому и предыдущему вопросу см. работы [Chen, Medioni, 1994] и [Sullivan, Sandford, Ponce, 1994].

*Большая размерность.* При большом количестве измерений исходных данных и/или при большом количестве параметров модели часто оказывается сложно дать эмпирическую или статистическую интерпретацию качества аппроксимации. Кроме того, при определении параметров модели каким-либо поисковым алгоритмом, может даже быть неизвестно, являются ли найденные параметры оптимальными или соответствуют некоторому локальному минимуму критерия ошибок.

*Ограничения аппроксимации.* Иногда подобранная модель должна удовлетворять дополнительным ограничениям. Например, может потребоваться найти наилучшую прямую, проходящую через точки данных и при этом перпендикулярную другой прямой. Методы оптимизации при наличии ограничений описываются в учебниках, приведенных в списке литературы.

### **Сегментация кривых с использованием методов аппроксимации**

В описанном выше методе подбора моделей предполагается, что известны и вид модели, и набор исходных точек. Для получения исходных точек в виде длинных контурных цепочек могут применяться алгоритмы прослеживания границ.

Контурные цепочки можно сегментировать следующим образом. Во-первых, найдем контурные точки с большой кривизной (заострения контуров). Затем можно выделить сегменты контуров, ограниченные точками разрывов цепочки. Эти сегменты аппроксимируются с использованием одной или нескольких известных моделей. В результате этой обработки будет определено множество криволинейных сегментов, математическая модель и параметры, описывающие форму каждого сегмента.

Альтернативный метод предполагает использование процедуры аппроксимации для сегментации исходной контурной кривой. На первом этапе каждый фрагмент кривой из  $k$  последовательных точек аппроксимируется с использованием каждой из рассматриваемых моделей. Для каждого удовлетворительного варианта аппроксимации в массиве результатов сохраняется соответствующее значение  $\chi^2$ . На втором этапе производится попытка расширения множеств аппроксимируемых точек путем последовательного добавления конечных точек соседних фрагментов кривой. Аппроксимируемые сегменты растут до тех пор, пока очередная конечная точка не приведет к уменьшению значения  $\chi^2$ , описывающего результаты аппроксимации. Полученное множество сегментов передается высокоуровневым процедурам обработки, которые

могут обнаружить образы объектов предметной области по найденным сегментам контуров. Этот процесс напоминает сегментацию методом наращивания областей, которая рассматривалась в разд. 10.1.2. Только для наращивания отрезков прямых здесь используется оценка направления в каждом контурном пикселе, а не значение интенсивности, которое использовалось при наращивании областей.

## 10.5 ОБНАРУЖЕНИЕ ВЫСОКОУРОВНЕВЫХ СТРУКТУР

В процессе анализа изображений часто требуется обрабатывать комбинации разнотипных сегментов. Например, при обнаружении образа здания могут совместно учитываться четырехугольные области и отрезки прямых; пересекающиеся контурные сегменты могут определять угол здания, или, зеленая область внутри синей может означать наличие острова на карте. Способы комбинирования сегментов изображений безграничны. Ниже будут рассмотрены два общих случая объединения контурных сегментов для построения более информативных структур — *лент* и *углов*.

### 10.5.1 Ленты

К лентам (*ribbon*) относятся различные сегменты изображений, образы которых обладают осевой симметрией. Ленты часто встречаются на изображениях продолговатых двумерных и трехмерных объектов. Примерами подобных изображений являются изображения проводников на печатной плате, двери дома, ручки на столе, или дороги, проходящей через поля. В этих примерах стороны лент примерно параллельны, но не обязательно прямолинейны. Далее будут рассматриваться ленты с прямолинейными сторонами, хотя ленты могут иметь и другую форму, например, как силуэт бутылки или фонарного столба. В этих случаях форма образов объектов описывается некоторой сложной кривой, но образ является зеркально симметричным относительно оси ленты. Ленточные образы на изображениях формируют многие объекты, например, электрический шнур, канат, извилистая река или дорога, тень от каната или фонарного столба. В гл. 14 рассматриваются трехмерные объекты, называемые *обобщенными цилиндрами*. Образы этих объектов на изображениях имеют форму лент. Символ на рис. 10.16 (слева) хорошо описывается четырьмя лентами, две из которых имеют большую кривизну. Обнаружение лент общего вида оставим в качестве материала для дополнительного изучения и сосредоточимся далее на лентах с прямолинейными сторонами.

**77 Определение.** **Лентой** называется продолговатая область, форма которой приблизительно симметрична относительно главной оси. Часто, но не всегда, края ленты характеризуются симметричным контрастом с фоном.

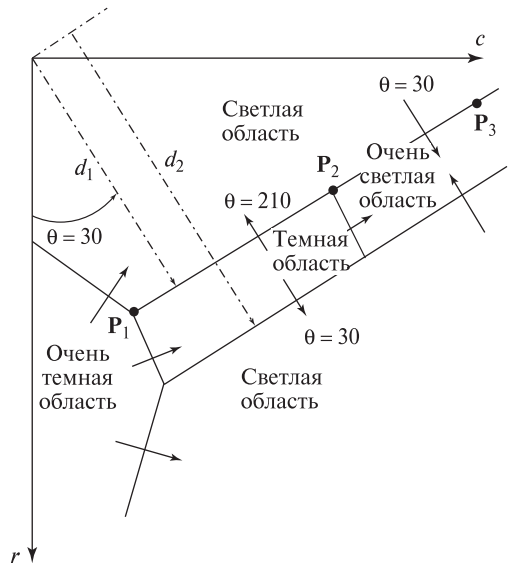
Алгоритм преобразования Хафа позволяет определить ориентацию и местоположение краев на изображении. На рис. 10.26 поясняется, как этот алгоритм можно расширить для определения градиента края. В гл. 5 и выше в данной

главе отмечалось, что направление градиента  $\theta$  в пикселе  $[r, c]$ , которому соответствует значительная величина градиента, можно вычислить как угол в полуинтервале  $[0, 2\pi)$  с помощью операторов типа оператора Собеля. Вектор, проведенный из начала растровой системы координат к данному пикселу, равен  $[r, c]$ . Проекция этого пиксела на единичный вектор с направлением  $\theta$  равна расстоянию со знаком  $d$ :

$$d = [r, c] \circ [-\sin \theta, \cos \theta] = -r \sin \theta + c \cos \theta \quad (10.26)$$

Положительные значения  $d$  можно вычислять на основе представления пиксела  $[r, c]$  в полярной системе координат. Отрицательные значения  $d$  соответствуют случаям, когда градиент противоположен направлению от начала координат до края. Например, на изображении шахматной доски для каждой прямой формируются два отдельных кластера, соответствующих двум различным направлениям градиента.

Рассмотрим прямолинейный край  $P_2P_3$  на рис. 10.26. Все пикселы вдоль края будут характеризоваться градиентом с направлением примерно 30 градусов. Перпендикуляр из начала координат к прямой  $P_2P_3$  имеет такое же направление, поэтому пикселы вдоль  $P_2P_3$  будут преобразованы (приблизительно) в точку  $[d_1, 30^\circ]$  в параметрическом пространстве преобразования Хафа. Однако пикселы вдоль отрезка  $P_1P_2$  характеризуются градиентом с направлением 210 градусов. В этом случае градиент противоположен направлению перпендикуляра из начала координат к прямой  $P_1P_2$ . Значит, пикселы отрезка  $P_1P_2$  преобразуются (приблизительно) в точку  $[-d_1, 210^\circ]$ .



**Рис. 10.26.** Преобразование Хафа позволяет определить местоположение и ориентацию края и направление его градиента. Градиент, соответствующий переходу от темной области к светлой, направлен противоположно градиенту перехода в обратном направлении, от светлой области к темной.



### Упражнение 10.17

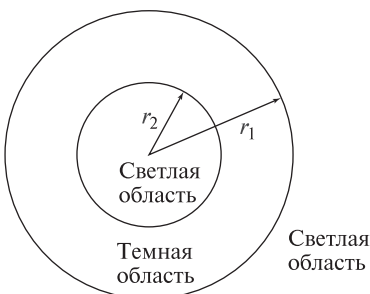
На рис. 10.27 изображено темное кольцо на светлом фоне с центром в начале координат изображения. Схематично изобразите параметрическое пространство, которое могло бы получиться при применении к этому изображению преобразования Хафа, дополненного оценкой направления градиента (см. рис. 10.26).

**Обнаружение прямолинейных лент** Параметры преобразования Хафа вместе со списками точек, полученными с помощью алгоритма *accumulate\_lines*, позволяют обнаружить на изображении более сложные структуры. Например, два края с направлениями, отличающимися на  $180^\circ$ , свидетельствуют о возможном наличии ленты. Если при этом точки из списков двух краев расположены близко друг к другу, то это указывает на присутствие на изображении крупного характерного признака в виде области линейной формы (например, образы колонн на рис. 10.17).

На рис. 10.28 показано изображение части здания с водосточной трубой. Изображение было получено при ярком солнечном освещении и поэтому содержит резкие тени. Путем применения градиентного оператора и накопления пикселей краевых сегментов с помощью *accumulate\_lines* можно обнаружить очевидное присутствие яркой ленты на темном фоне, которая соответствует водосточной трубе (стороны АВ и ED). Тень от этой трубы **S** создает образ темной ленты на светлом фоне.

### Упражнение 10.18

Напишите программу для изучения возможностей использования преобразования Хафа для обнаружения лент. (а) Для вычисления величины и направления градиента во всех пикселах используйте оператор Собеля. (б) Найдите кластеры в параметрическом пространстве  $[d, \theta]$ . (в) Найдите пары кластеров  $([d_1, \theta_1], [d_2, \theta_2])$ , у которых углы  $\theta_1$  и  $\theta_2$  отличаются на  $\pi$ . (г) Удалите пары, которые не являются примерно симметричными относительно оси, проходящей посередине между ними.



**Рис. 10.27.** На светлом фоне расположена темная кольцевая область с центром в начале координат изображения. Внутри темной области содержится светлый круг.



**Рис. 10.28.** (Слева) область изображения дома, на которой видны водосточная труба и резкие тени. (В центре) точки, в которых значения градиентов превышают 90% от максимального значения. Для вычисления градиента применялся оператор Превитт с маской  $3 \times 3$ . (Справа) схема расположения явно заметных лент и углов.

### 10.5.2 Обнаружение углов

Области углов (corner) на изображении можно обнаружить посредством поиска пар выделенных краевых сегментов  $E_1$  и  $E_2$ , удовлетворяющих перечисленным ниже условиям.

1. Аппроксимирующие прямые для множеств краевых точек  $E_1$  и  $E_2$  пересекаются в точке  $[u, v]$ , координаты которой определены в действительной системе координат изображения.
2. Точка  $[u, v]$  лежит близко от крайних точек множеств  $E_1$  и  $E_2$ .
3. Направления градиентов  $E_1$  и  $E_2$  симметричны относительно оси симметрии двух этих краев.

Приведенные условия определяют только углы типа 'L'. Условие (2) исключает из рассмотрения углы типа 'T', 'X' и 'Y'. Координаты точки пересечения аппроксимирующих прямых  $[u, v]$  вычисляются с субпиксельной точностью. На рис. 10.29 приведена схема геометрической структуры угла. Исходные краевые сегменты выделяются с помощью преобразования Хафа, путем прослеживания границ и последующей аппроксимации прямыми, или каким-либо другим подходящим алгоритмом. Для каждой пары отрезков  $([d_1, \theta_1], [d_2, \theta_2])$ , удовлетворяющей перечисленным условиям, четверка элементов  $([d_1, \theta_1], [d_2, \theta_2], [u, v], \alpha)$  добавляется в множество кандидатов на углы (где  $\alpha$  — это величина угла). Сформированное множество угловых характерных признаков применяется в различных целях. Например, для построения каких-либо высокоуровневых описаний, для непосредственного сопоставления изображений или в методах деформации изображений (см. гл. 11).

На изображении деталей на рис. 10.2 заметны несколько углов. Однако большинство из них возникают из-за перекрытий образов различных объектов при наблюдении из выбранной точки, и не соответствуют реальным соединениям граней трехмерных объектов. Четыре объективно существующих угла явно заметны на верхней грани арочной детали. На рис. 10.28 углы треугольника

ABC являются фиктивными углами, возникшими из-за особенностей освещения и выбранной точки наблюдения. В завершение рассмотрения данного вопроса отметим, что хотя представления с использованием краевых сегментов применяются для решения некоторых задач, но в общем случае они могут быть весьма неоднозначны. Для интерпретации высокоуровневых структур обычно необходимо использовать дополнительные сведения из предметной области задачи.

### Упражнение 10.19

Предложите способ модификации алгоритма обнаружения ленты, чтобы (а) алгоритм обнаруживал только ленты с примерно вертикальным направлением, (б) алгоритм обнаруживал ленты шириной не больше  $W$ .

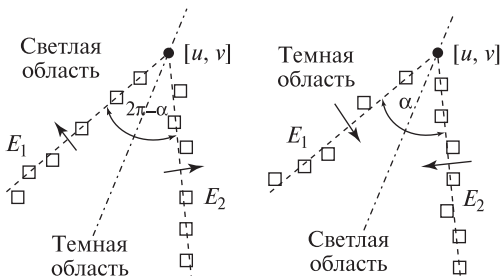
## 10.6 СЕГМЕНТАЦИЯ НА ОСНОВЕ СОГЛАСОВАННОГО ДВИЖЕНИЯ

Как уже отмечалось ранее, информация о движении важна для определения содержания сцены и поведения наблюдаемых объектов. В гл. 9 рассматривались методы для обнаружения изменений в сцене и для слежения за движущимися объектами на последовательности кадров.

### 10.6.1 Границы движущихся объектов

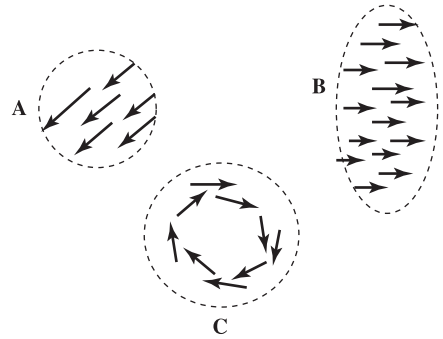
Для обнаружения контуров движущихся объектов можно использовать данные как о пространственном, так и о временном контрасте. В рассматривавшихся ранее примерах учитывался только пространственный контраст некоторого характерного признака, например, интенсивности или текстуры. Оценка контраста производилась по данным одного изображения. Если имеются два изображения сцены  $I[x, y, t]$  и  $I[x, y, t + \Delta t]$ , то по ним можно вычислить и совместно рассматривать пространственный и временной градиенты.

Определим *величину пространственно-временного градиента* (spatio-temporal gradient magnitude) как произведение модулей пространственного и временного градиента (см. выражение (10.27)). Значения этой величины для всех пикселей пары изображений можно представить в виде изображения  $STG[]$ . К этому изображению можно применять все рассматривавшиеся выше



**Рис. 10.29.** Углы определяются парой выделенных краевых прямолинейных сегментов, которые расположены в соответствующем пространственном взаимоотношении.

**Рис. 10.30.** Учет данных о векторах перемещения в пределах кластеров поля движения. Кластеры выделяются с учетом местоположения векторов перемещения, скорости и направления. Обнаружить поступательно движущиеся объекты (А,В) проще, чем вращающиеся (С).



методы выделения контуров. Но при этом выделенные контуры будут соответствовать границам движущихся объектов, а не неподвижных.

$$STG[x, y, t] = \text{Mag}[x, y, t] (|I[x, y, t] - I[x, y, t + \Delta t]|) \quad (10.27)$$

### 10.6.2 Накопление траекторий движения

Предположим, что по двум кадрам видеопоследовательности были вычислены векторы перемещения. Возможно, они определены только для некоторых точек или областей интереса (см. гл. 9). Векторы перемещений можно использовать для сегментации изображения на области. Кластеризация при этом должна выполняться с учетом местоположения векторов перемещения на изображении, соответствующей им скорости и направления движения (см. рис. 10.30). Плотные кластеры будут соответствовать поступательно движущимся объектам, так как все точки таких объектов движутся с одинаковой скоростью. С помощью более сложной обработки можно обнаружить вращающиеся объекты.

#### Упражнение 10.20

Даны параметры двух прямых  $([d_1, \theta_1], [d_2, \theta_2])$ . (а) Запишите формулу для вычисления их точки пересечения  $[x, y]$  и (б) формулу для определения их оси симметрии  $[d_a, \theta_a]$ .

#### Упражнение 10.21

Получите два последовательных изображения сцены с движущимися объектами и с помощью уравнения (10.27) вычислите для них пространственно-временное изображение. (В качестве изображений можно взять два кадра из видеопоследовательности формата Motion JPEG. Или получите изображение нескольких темных объектов с помощью планшетного сканера, а затем слегка сместите эти объекты и получите второе изображение.)

На рис. 10.31 поясняется ход обработки изображений в задаче, в которой информация о движении использовалась для организации пользовательского

интерфейса. Цель заключалась в обеспечении ввода данных в компьютер посредством языка жестов American Sign Language (ASL). На рисунке приведены несколько кадров из видеопоследовательности длительностью примерно 2 секунды, на которой представлена жестикуляция человека.

Для получения результатов, показанных на рис. 10.31, выполнялась сегментация по цвету в пределах одного кадра и сегментация по движению на паре кадров. Основные операции обработки представлены в виде алгоритма 10.10. Более подробное описание алгоритма, результаты работы которого показаны на рис. 10.31, можно найти в работе [Yang, Ahuja, 1999].

Первые несколько шагов алгоритма могут оказаться полезными и во многих других задачах обработки видеопоследовательностей. Для каждого изображения выполняется цветовая сегментация. Затем выполняется сопоставление сегментов, выделенных на двух последовательных изображениях. Результаты успешного сопоставления используются для организации вычислений плотного поля движения для каждой пары последовательных изображений. Затем поле движения сегментируется с целью определения траекторий движения на уровне отдельных пикселей. На основе этих данных выделяются области с примерно одинаковыми параметрами движения. Только на этом этапе учитываются специфические правила предметной области, позволяющие обнаружить образы рук и лица. Модель цвета кожи (см. гл. 6) применяется для выделения областей человеческой кожи. Крупнейшая из этих областей считается образом лица человека. Местоположение центров обеих рук прослеживается на всех кадрах видеопоследовательности. В результате определяются две траектории, которые могут использоваться для распознавания жестовой информации. В работе [Yang, Ahuja, 1999] сообщается, что в экспериментах по обработке большого количества образцов, состоящих из 40 знаков языка жестов American Sign Language, был достигнут коэффициент распознавания более 90 процентов.

---

### Упражнение 10.22

Предложите способы модификации алгоритма 10.10, так чтобы его можно было упростить и ускорить за счет специализации всех операций обработки применительно к задаче распознавания жестов языка ASL.

---

### Упражнение 10.23

Допустим, что известны две траектории движения  $\mathbf{P}_j, j = 1, N$  и  $\mathbf{Q}_k, k = 1, M$ , где  $\mathbf{P}_j$  и  $\mathbf{Q}_k$  являются двумерными точками на кадрах видеопоследовательности. Разработайте алгоритм для сопоставления двух траекторий, так чтобы идентичным траекториям соответствовало значение 1.0, а сильно различным — значение 0.0. Отметим, что  $M$  и  $N$  могут иметь разные значения.



(a) Кадр 14

(b) Кадр 16

(c) Кадр 19

(d) Кадр 22

(I) Четыре кадра 55-кадровой видеопоследовательности со съемкой слова *cheerleader* на языке ASL.



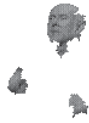
(e) Кадр 14

(f) Кадр 16

(g) Кадр 19

(h) Кадр 22

(II) Сегментация по движению последовательности изображений с жестовым представлением слова *cheerleader*. (Пиксели одной и той же области движения отображены одним оттенком серого цвета. Различным областям соответствуют разные оттенки серого цвета.)



(i) Кадр 14



(j) Кадр 16

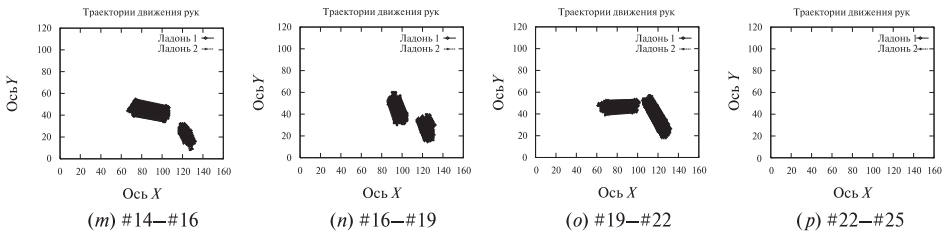


(k) Кадр 19



(l) Кадр 22

(III) Обнаруженные области головы и рук человека на видеопоследовательности с жестовым представлением слова *cheerleader*.



(IV) Выделенные траектории движения, соответствующие фрагментам жестового представления слова *cheerleader*. Одновременно показаны все пиксели траекторий, поэтому они образуют крупные пятна

**Рис. 10.31.** Выделение траекторий движения на последовательности изображений. (I) Примеры кадров видеопоследовательности. (II) Сегментация кадров на основе информации о движении. (III) Области ладоней и головы человека, обнаруженные по цвету и размерам. (IV) Траектории движения центров областей ладоней. (С любезного разрешения Ming-Hsuan Yang, Narendra Ahuja.)

**Входные данные:** видеопоследовательность со съемкой человеческой жестикуляции на языке ASL.

**Выходные данные:** траектории движения образов двух ладоней.

1. Цветовая сегментация каждого кадра  $I_t$  видеопоследовательности на области примерно одинакового цвета.
2. Сопоставление областей на каждой паре изображений ( $I_t, I_{t+1}$ ) по цвету и местоположению.
3. Вычисление параметров аффинного преобразования, переводящего каждую область изображения  $I_t$  в сопоставленную ей область изображения  $I_{t+1}$ .
4. Вычисление векторов перемещений для отдельных пикселей с использованием параметров преобразований сопоставленных областей.
5. Сегментация вычисленного поля движения на основе данных о согласованном движении и местоположении векторов движения.
6. Обнаружение областей двух рук и лица с использованием цветовой модели кожи.
7. Слияние смежных областей телесного цвета, которые до сих пор были фрагментированы.
8. Вычисление аппроксимирующих эллипсов для образа каждой руки и лица.
9. Формирование траекторий движения путем прослеживания положения центра каждого эллипса на всей видеопоследовательности.
10. (Распознавание жестикуляции с использованием данных о траекториях образов двух рук.)

**Алгоритм 10.10.** Алгоритм, учитывающий информацию о цвете и движении для прослеживания знаков языка жестов ASL. (По материалам работы [Yang, Ahuja, 1999].)

## 10.7 ЛИТЕРАТУРА

Сегментация представляет собой одну из старейших, но еще не окончательно изученную область компьютерного зрения. В обзоре [Haralick, Shapiro, 1985] приведено рассмотрение ранних работ по сегментации, большинство из которых предназначались для полутоновых изображений. Одной из первых работ по цветовой сегментации естественных цветных изображений является статья [Ohlander, Price, Reddy, 1978]. Значительный прогресс в этой области достигнут только в последние годы. Катализатором новых исследований в области сегментации произвольных цветных изображений можно назвать работу [Shi, Malik, 1997] по нормированным разрезам. В области анализа контурных изоб-

ражений Фриман (Freeman) первым предложил цепной код в 1960-х гг.; в его статье 1974 г. обсуждаются способы использования этого кода. Преобразование Хафа первоначально было опубликовано в виде патента. Оно стало широко известным благодаря статье [Duda, Hart, 1972]. Применение преобразования Хафа для выделения отрезков прямых было продемонстрировано в работе [O’Gorman, Clowes, 1976], а для выделения окружностей — в работе [Kimme, Ballard, Sklansky, 1975]. Десятью годами позже Барнс (Burns) предложил свой определитель прямых в качестве более робастной и надежной модификации преобразования Хафа. В книге [Samet, 1990] о пространственных структурах данных содержится отличное рассмотрение квадрантных деревьев.

В работе Boyer и др. (1994) рассмотрено применение робастных статистик для аппроксимации данных измерений в задачах сегментации. В принципе, любая предполагаемая модель может быть подогнана в любой области изображения. Робастная аппроксимация значительно уменьшает количество выбросов, возникающих в областях успешной подгонки известной модели. Изображение может считаться сегментированным, если на нем успешно удалось подобрать параметры всех ожидаемых аппроксимирующих моделей. В таком случае сегментами изображений называются множества точек, которые использовались при аппроксимации.

1. Ballard, D. H., 1981, Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recog.*, v. 13(2):111–122.
2. Boyer, K., K. Mirza, and G. Ganguly. 1994. The robust sequential estimator: a general approach and its application to surface organization in range data, *IEEE Trans. Pattern Analysis and Machine Intelligence*, v. 16(10) (Oct. 1994), 987–1001.
3. Burns, J. R., A. R. Hanson, and E. M. Riseman. 1986. Extracting straight lines. *IEEE Trans. Pattern Analysis and Machine Intelligence*, v. PAMI-8:425–455.
4. Chen, Y., and G. Medioni. 1994. Surface description of complex object from multiple range images. *Proc. IEEE Conf. Comput. Vision and Pattern Recog.*, Seattle, WA (June 1994), 513–518.
5. Duda, R. O., and P. E. Hart. 1972. Use of the Hough transform to detect lines and curves in pictures. *Communications of the ACM*, v. 15:11–15.
6. Freeman, H. 1974. Computer processing of line-drawing images. *Computing Surveys*, v. 6:57–97.
7. Haralick, R. M., and L. G. Shapiro. 1985. Image segmentation techniques. *Comput. Vision, Graphics, and Image Proc.*, v. 29(1) (January 1985), 100–132.
8. Kasturi, R., S. Bow, W. El-Masri, J. Shah, J. Gattiker, and U. Mokate. 1990. A system for interpretation of line drawings. *IEEE Trans. Pattern Analysis and Machine Intelligence*, v. PAMI-12:978–992.



9. Kimme, C., D. Ballard, and J. Sklansky. 1975. Finding circles by an array of accumulators. *Communications of the ACM*, v. 18:120–122.
10. O’Gorman, F., and M. B. Clowes. 1976. Finding picture edges through collinearity of feature points. *IEEE Trans. Comput.*, v. C-25:449–454.
11. Ohlander, R., K. Price, and D. R. Reddy. 1978. Picture segmentation using a recursive region splitting method. *Comput. Graphics and Image Proc.*, v. 8:313–333.
12. Ohta, Y., T. Kanade, and T. Sakai. 1980. Color information for region segmentation. *Comput. Graphics and Image Proc.*, v. 13:222–241.
13. Rao, K. 1988. *Shape Description from Sparse and Imperfect Data*. Ph.D. thesis, Univ. of Southern California.
14. Samet, H. 1990. *Design and Analysis of Spatial Data Structures*. Addison-Wesley, Reading, MA.
15. Shi, J., and J. Malik. 1997. Normalized cuts and image segmentation. *IEEE Conf. Comput. Vision and Pattern Recog.*, 731–737.
16. Sullivan, S., L. Sandford, and J. Ponce. 1994. Using geometric distance for 3D object modeling and recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, v. 16(12) (Dec. 1994), 1183–1196.
17. Yang, M.-H., and N. Ahuja. 1999. Recognizing hand gesture using motion trajectories. *Proc. IEEE Conf. Comput. Vision and Pattern Recog. 1999*, Ft. Collins, CO (23–25 June 1999), 466–472.

# Сопоставление в двумерном пространстве

В данной главе изучаются способы определения и использования информации о соответствии между изображениями, между изображениями и картами, а также между изображениями и моделями объектов. Все операции рассматриваются в двумерном пространстве. В гл. 14 они будут расширены на задачи сопоставления типа 3D-2D и 3D-3D. Существует большое количество задач, в которых требуется непосредственное применение методов двумерного сопоставления и не возникает необходимости использования более сложных алгоритмов обработки трехмерных данных.

Рассмотрим задачу учета городских построек для строительного планирования или в целях налогообложения. В ясный день с самолета была проведена аэрофотосъемка города. Полученные изображения сравниваются с последними картами соответствующих городских районов, чтобы в результате построить обновленные карты. Кроме того, модифицируются различные базы данных, чтобы в них хранилась актуальная информация о существующих зданиях, дорогах, нефтяных резервуарах и т. п. Возможно, потребуется даже сохранить данные о видах посевов на полях около города. Вся эта работа может быть проделана вручную, но в настоящее время обычно выполняется с помощью компьютера.

Второй пример относится к медицине. Предположим, что требуется исследовать потоки крови, проходящие через сердце и легкие пациента. Сначала в обычных условиях делается рентгеновский снимок. Затем в кровь пациента добавляется специальный краситель и делается второй снимок. На втором изображении будут заметны потоки крови, но также будет присутствовать шум и различные посторонние фрагменты, соответствующие различным тканям человеческого тела, например, костям. Вычитание первого изображения из второго позволяет уменьшить шум и количество посторонних деталей. Это облегчает восприятие фрагментов двух изображений, отличающихся из-за применения красителя. Но перед вычитанием изображений требуется, чтобы первое изображение было *геометрически преобразовано* (или *деформировано*)

для совмещения со вторым изображением. Это преобразование необходимо для компенсации небольших сдвигов образов тканей тела, вызванных изменением позы пациента, биением сердца, дыханием и т. п.

### 11.1 СОВМЕЩЕНИЕ ДВУМЕРНЫХ ДАННЫХ

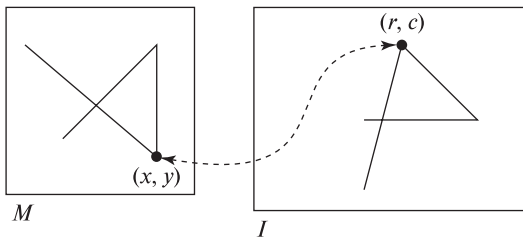
Ко всем задачам сопоставления из данной главы, а также ко многим другим, применима общая математическая модель. Эта простая модель описывает обратимое отображение одного двумерного координатного пространства на другое. В форме уравнений (11.1) и на рис. 11.1 представлено обратимое отображение точек модели  $\mathbf{M}$  на точки изображения  $\mathbf{I}$ . Но в качестве  $\mathbf{M}$  и  $\mathbf{I}$  могут рассматриваться любые двумерные координатные пространства. Например, это может быть карта, модель объекта или изображение.

$$\begin{aligned} M[x, y] &\cong I[g(x, y), h(x, y)] \\ I[r, c] &\cong M[g^{-1}(r, c), h^{-1}(r, c)] \end{aligned} \quad (11.1)$$

**78 Определение.** Отображение одного двумерного координатного пространства на другое называется **двумерным преобразованием** (2D transformation).

Преобразование, определяемое уравнением (11.1), иногда называется пространственным преобразованием, геометрическим преобразованием или деформацией (термин «деформация» обычно применяется только для нелинейных преобразований). Отображающие функции  $g$  и  $h$  задают соответствие между модельными точками  $[x, y]$  и точками изображения  $[r, c]$ . Эти функции позволяют для любого точечного признака модели вычислить координаты соответствующей точки изображения. Будем полагать, что отображающие функции являются обратимыми. В таком случае с помощью обратных функций можно выполнять преобразование в двух направлениях.

В упоминавшейся задаче учета городских построек отображающие функции позволили бы преобразовывать контуры объектов на карте в контуры



**Рис. 11.1.** Отображение двумерного пространства  $M$  на  $I$ .  $M$  может быть моделью, а  $I$  — изображением, но в общем случае это могут быть два любых двумерных координатных пространства.



**Рис. 11.2.** (Слева) изображение сцены с дорожным знаком, использовавшееся в гл. 8. (Справа) новое изображение, полученное из исходного с помощью выборочно-го преобразования.

объектов на изображении аэрофотосъемки. Область изображения, соответствующая объекту карты, затем может анализироваться для обнаружения новых зданий, видов сельскохозяйственных посевов и т. п. (В настоящее время такие задачи обычно решаются человеком с использованием интерактивных программ на графических станциях.)

Если отображающие функции известны в рассмотренной выше медицинской задаче, то для анализа рентгенологом можно сформировать разностное изображение  $I_2[r_2, c_2] - I[g(r_2, c_2), h(r_2, c_2)]$ . В данном примере отображающие функции позволяют совместить точки двух изображений.

**79 Определение.** Совмещением изображений называется такой процесс геометрического преобразования точек двух изображений, в результате которого соответствующие признаковые точки двух изображений получают одинаковые координаты. При этом предполагается, что на двух изображениях представлена одна и та же сцена, видимая с близких точек наблюдения.

Еще одной распространенной и важной операцией, хотя и отличающейся от операции сопоставления, является формирование нового изображения посредством выборки некоторых пикселей исходного изображения. Например, на рис. 11.2 показано, как из изображения  $I_1$  (слева) можно вырезать его часть  $I_2$  (справа). Хотя содержимое нового изображения  $I_2$  представляет фрагмент исходного изображения  $I_1$ , возможно, что  $I_2$  будет содержать столько же пикселей, как и  $I_1$  (или даже больше).

Ряд теоретических аспектов, касающихся двумерных преобразований, также имеют большое практическое значение. Какой вид имеют функции  $g$  и  $h$ , являются ли они линейными, непрерывными и т. п.? Преобразуются ли прямые линии из одного пространства в прямые или кривые линии другого пространства? Одинаковы ли расстояния между парами точек в обоих пространствах? Как можно использовать свойства различных функций для построения требуе-

мого отображения? Будет ли двумерное пространство модели или изображения непрерывным или дискретным? Если по крайней мере одно из пространств является цифровым изображением, то на точность и качество визуального представления будут влиять эффекты дискретизации. (Для пояснения последнего замечания типичные эффекты дискретизации были специально сохранены на рис. 11.2 (справа).)

### Упражнение 11.1

Предложите способ повышения качества изображения на рис. 11.2 (справа) для устранения эффектов дискретизации (ступенчатости контуров).

## 11.2 ПРЕДСТАВЛЕНИЕ ТОЧЕК

В данной главе операции с точками рассматриваются только в двумерных координатных пространствах (на плоскости). Обобщение определений и результатов на трехмерный случай будет сделано позже в гл. 13. Большинство этих обобщений, хотя и не все, выполняются непосредственным образом. При первоначальном изучении предмета обычно оказывается проще изучить основные понятия и систему обозначений в двумерном случае, а затем переходить к трехмерному пространству, для которого излагаемый материал иногда оказывается более сложным.

Двумерная точка задается двумя координатами и может быть представлена в виде вектора-строки  $\mathbf{P} = [x, y]$  или вектора-столбца  $\mathbf{P} = [x, y]^t$ . Далее при записи уравнений будет использоваться представление точек в виде вектора-столбца. Такая форма записи применяется в большинстве технических книг. При этом обозначения преобразований  $\mathbf{T}$  записываются слева от точек  $\mathbf{P}$ , к которым применяются эти преобразования. Непосредственно в тексте для удобства восприятия будет использоваться представление точек в виде вектора-строки без указания индекса транспонирования  $t$ . В такой записи координаты отделяются друг от друга запятыми. При записи точки в виде вектора-столбца с вертикально расположенными компонентами запятые не указываются.

$$\mathbf{P} = [x, y]^t = \begin{bmatrix} x \\ y \end{bmatrix}$$

Иногда точки будут снабжаться метками, обозначающими тип характерных признаков, которым соответствуют эти точки. Например, точка может быть центром отверстия, вершиной многоугольника или вычисленной точкой пересечения двух отрезков. Тип точек будет учитываться в алгоритмах автоматического сопоставления, которые рассматриваются далее в данной главе.

**Системы координат** Координаты точек всегда задаются в некоторой системе координат. В системах машинного зрения для анализа наблюдаемых

сцен часто применяются несколько систем координат (они рассматривались в гл. 2). При использовании нескольких систем координат будем обозначать систему координат, в которой задана данная точка, с помощью верхнего индекса.

**80 Определение.** Координаты некоторой признаковой точки  $\mathbf{P}_j$  в системе координат  $\mathbf{C}$  будем обозначать как  ${}^c\mathbf{P}_j$ .

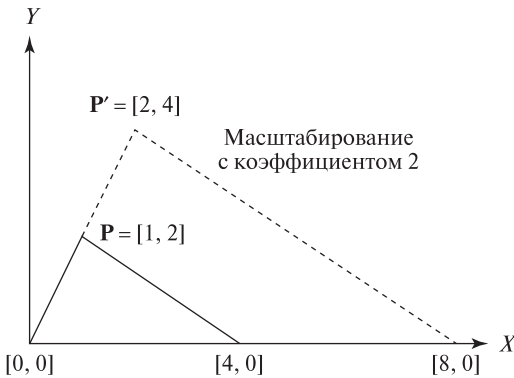
**Однородные координаты** Как будет показано далее, для упрощения системы обозначений и компьютерной обработки бывает удобно пользоваться *однородными координатами точек*. Особенно часто они применяются в операциях с аффинными преобразованиями.

**81 Определение.** **Однородные координаты** двумерной точки  $\mathbf{P} = [x, y]^t$  записываются в виде  $[sx, sy, s]^t$ , где  $s$  — масштабный коэффициент, обычно равный 1.0.

В заключение раздела обсудим соглашения относительно координатных систем, используемых на иллюстрациях в этой книге и в программах визуализации изображений. Системы координат на иллюстрациях в данной главе обычно соответствуют прямоугольной правой системе координат, применяемой в учебниках по математике. Одна координатная ось (обозначается  $x$ ,  $u$  или  $r$ ) направлена из начала координат вправо, а другая ( $y$ ,  $v$  или  $c$ ) — вверх. Однако в программах, обрабатывающих изображения, изображение из  $n$  строк и  $m$  столбцов часто рассматривается в пиксельной системе координат. При этом первая строка (с координатой  $r = 0$ ) считается расположенной сверху изображения, а последняя строка (с координатой  $r = n - 1$ ) — внизу. Таким образом, в пиксельной системе координат значения координат  $r$  увеличиваются сверху-вниз, а значения координат  $c$  — слева-направо относительно изображения. Указанное расхождение не составляет сложностей для записи алгебраических выражений, но может затруднять восприятие результатов, так как выводимое на экран компьютера изображение нужно представлять себе повернутым на 90 градусов против часовой стрелки. Тогда его ориентация будет согласоваться с ориентацией применяемой в математике правой системы координат.

## 11.3 АФФИННЫЕ ГЕОМЕТРИЧЕСКИЕ ПРЕОБРАЗОВАНИЯ

Большое семейство практически полезных пространственных преобразований можно представить в виде произведения матрицы и точки в однородных координатах. Далее будут кратко рассмотрены основные преобразования в однородных координатах. Более подробное описание можно найти в одной из книг по компьютерной графике или робототехнике из списка литературы в конце главы. Свойства векторных пространств обсуждались в гл. 5.



**Рис. 11.3.** Масштабирование обеих координат двумерного вектора с масштабным коэффициентом, равным 2.

**Масштабирование** Одним из наиболее часто применяемых преобразований является масштабирование (scaling). Однородное масштабирование одинаковым образом изменяет значения всех координат, т. е. одинаково изменяются размеры всех объектов. На рис. 11.3 показана двумерная точка  $\mathbf{P} = [1, 2]$ . После масштабирования с масштабным коэффициентом 2 была получена новая точка  $\mathbf{P}' = [2, 4]$ . Масштабирование с таким же коэффициентом было выполнено для всех трех вершин треугольника. Полученный треугольник показан пунктирными отрезками.

Масштабирование является *линейным преобразованием*. Поэтому его можно представить в виде суммы слагаемых, описывающих применение масштабного коэффициента к двум базисным векторам двумерного евклидова пространства. Например,  $[1, 2] = 1[1, 0] + 2[0, 1]$  и  $2[1, 2] = 2(1[1, 0] + 2[0, 1]) = 2[1, 0] + 4[0, 1] = [2, 4]$ .

В уравнении (11.2) масштабирование двумерной точки представлено в виде произведения координат исходной точки и матрицы, содержащей в диагональных элементах значения масштабных коэффициентов. Уравнение (11.3) представляет преобразование масштабирования общего вида, при котором единичные вектора в направлениях  $x$  и  $y$  могут изменяться по-разному.

В гл. 2 были описаны пять различных координатных систем. Переход от координат в действительной системе координат изображения, измеряемых в миллиметрах, к координатам в растровой системе координат, измеряемым в пикселах, выполняется с помощью масштабирования. При использовании камер с квадратными пикселями масштабные коэффициенты имеют одинаковые значения  $c_x = c_y = c$ . Для камер телевизионного стандарта отношение масштабных коэффициентов должно быть равно  $4/3$ .

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} c & 0 \\ 0 & c \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} cx \\ cy \end{bmatrix} = c \begin{bmatrix} x \\ y \end{bmatrix} \quad (11.2)$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} c_x & 0 \\ 0 & c_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} c_x x \\ c_y y \end{bmatrix} \quad (11.3)$$

---

**Упражнение 11.2.** Масштабирование для камеры с неквадратными пикселями

Предположим, что имеется квадратная ПЗС-матрица со стороной 1.25 см. На поверхности матрицы расположены 480 строк, каждая из которых содержит 640 пикселей. Запишите матрицу масштабирования для преобразования пиксельных координат  $[r, c]$  в координаты  $[x, y]$ , измеряемые в миллиметрах. Центр пикселя  $[0, 0]$  соответствует точке  $[0, 0]$  в мм. С помощью записанной матрицы преобразования вычислите вещественные координаты центра пикселя, расположенного в строке 100, столбце 200.

**Поворот** Еще одним известным преобразованием на плоскости является поворот относительно точки. На рис. 11.4 (слева) показана двумерная точка  $\mathbf{P} = [x, y]$ , которая была повернута относительно начала координат против часовой стрелки на угол  $\theta$ . В результате была получена новая точка  $\mathbf{P}' = [x', y']$ .

Преобразование поворота двумерной точки относительно начала координат записано в виде уравнения (11.4). Как и для любого линейного преобразования, столбцы матрицы поворота можно определить посредством применения этого преобразования к базисным векторам (рис. 11.4(справа)). Преобразование любого другого вектора можно представить в виде линейной комбинации базисных векторов.

$$\begin{aligned}
 R_\theta([x, y]) &= R_\theta(x[1, 0] + y[0, 1]) \\
 &= xR_\theta([1, 0]) + yR_\theta([0, 1]) = x[\cos \theta, \sin \theta] + y[-\sin \theta, \cos \theta] \\
 &= [x \cos \theta - y \sin \theta, x \sin \theta + y \cos \theta] \\
 \begin{bmatrix} x' \\ y' \end{bmatrix} &= \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x \cos \theta - y \sin \theta \\ x \sin \theta + y \cos \theta \end{bmatrix} \quad (11.4)
 \end{aligned}$$

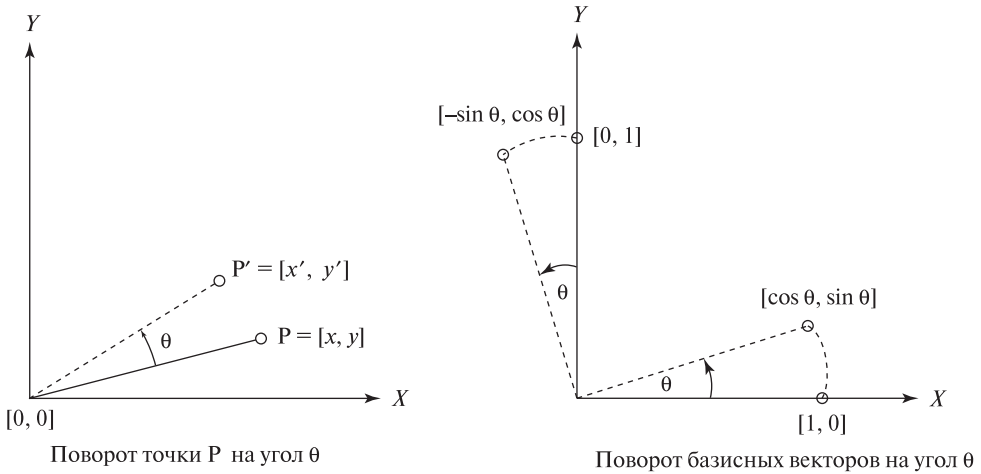
Повороты на плоскости можно выполнять относительно произвольной точки плоскости, а не только относительно начала координат. Этот вопрос рассматривается в упражнении в конце данного раздела.

---

**Упражнение 11.3**

- (а) Изобразите на координатной плоскости три точки  $[0, 0]$ ,  $[2, 2]$  и  $[0, 2]$ .  
 (б) С помощью уравнения (11.2) выполните масштабирование этих точек с масштабным коэффициентом 0.5 и нанесите результаты на график. (в) Постройте новый график, представляющий результаты поворота трех точек на 90 градусов относительно начала координат. Для выполнения поворота используйте уравнение (11.4). (г) Обозначим матрицу масштабирования  $S$  и матрицу поворота  $R$ . Для получения матрицы  $SR$  матрица  $R$  умножается слева на матрицу  $S$ . Различаются ли результаты преобразования трех точек с использованием матриц  $SR$  и  $RS$ ?





**Рис. 11.4.** Поворот произвольной точки на плоскости можно представить в виде комбинации поворотов базисных векторов.

### Ортогональные и ортонормированные преобразования\*

**82 Определение.** Множество векторов называется **ортогональным**, если все векторы этого множества попарно перпендикулярны (или, в эквивалентной формулировке, скалярные произведения всех пар векторов равны нулю).

**83 Определение.** Множество векторов называется **ортонормированным**, если оно ортогонально и все векторы множества имеют единичную длину.

Преобразование поворота сохраняет длину и ортогональность базисных векторов. Это можно проверить геометрически или алгебраически. В качестве непосредственного результата получается, что расстояние между двумя преобразованными точками равно расстоянию между исходными точками. Это свойство *жесткого преобразования*. Жесткое преобразование можно представить в виде композиции преобразований поворота и переноса. Жесткие преобразования часто используются в операциях с твердыми телами и для перехода между системами координат. Однородное масштабирование с коэффициентом, отличным от 1.0, не сохраняет длину, но сохраняет угол между векторами. Свойства преобразований важны для выбора свойств объектов, не зависящих от расположения камеры или объектов в пределах сцены.

**Перенос** Часто требуется выполнить сдвиг точек на заданное расстояние. Это преобразование эквивалентно замене начала координат. Например, преобразование переноса может потребоваться для преобразования координат (строка, столбец) пиксела изображения в координаты (широта, долгота) на

карте. Перенос не отображает начало координат  $[0, 0]$  на себя, поэтому его не удастся представить с помощью матрицы  $2 \times 2$ , как это делалось для масштабирования и поворота. Другими словами, перенос не является линейным преобразованием. Для матричного представления переноса и некоторых других операций можно перейти к матрицам размерности  $3 \times 3$ . Соответственно, к двумерным координатам точки потребуется добавить еще одну координату. Для этого перейдем к однородным координатам. Добавляемая координата обычно выбирается равной 1.0, но иногда используются и другие значения.

$$\mathbf{P} = [x, y] \simeq [wx, wy, w] = [x, y, 1] \quad w = 1$$

Преобразование переноса можно представить в виде произведения матрицы переноса и вектора-столбца с однородными координатами точки (см. уравнение (11.5)). В результате преобразования точки  $[x, y]$  с использованием матрицы переноса  $\mathbf{D}$  получается точка  $[x', y'] = \mathbf{D}([x, y]) = [x + x_0, y + y_0]$ .

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x + x_0 \\ y + y_0 \\ 1 \end{bmatrix} \quad (11.5)$$

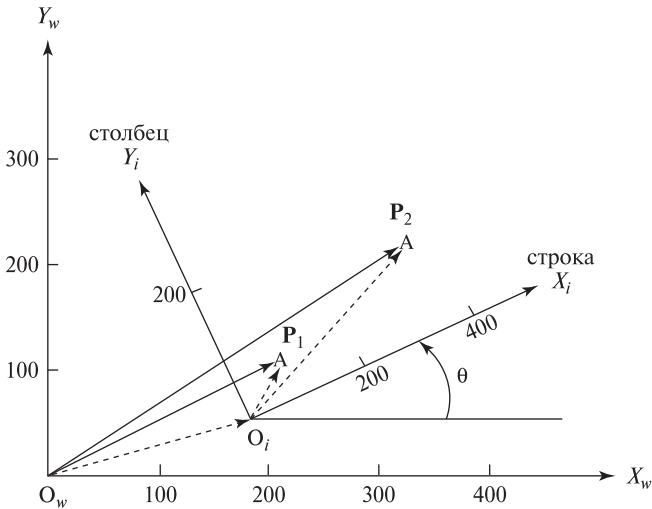
#### Упражнение 11.4. Поворот относительно точки

Запишите матрицу  $3 \times 3$ , представляющую поворот на плоскости относительно точки  $[5, 8]$  на угол  $\pi/2$ . *Рекомендации:* Сначала получите матрицу  $\mathbf{D}_{-5, -8}$  для переноса точки  $[5, 8]$  в начало новой системы координат. Матрица искомого преобразования равна произведению трех матриц  $\mathbf{D}_{5,8} \mathbf{R}_{\pi/2} \mathbf{D}_{-5, -8}$ . Проверьте, что эта матрица позволяет получить корректные результаты преобразования точек  $[5, 8]$ ,  $[6, 8]$ , и  $[5, 9]$ .

#### Упражнение 11.5. Симметрия относительно координатной оси

При *симметричном отражении* относительно оси  $OY$  базисный вектор  $[1, 0]$  преобразуется в  $[-1, 0]$ , а базисный вектор  $[0, 1]$  — в вектор  $[0, 1]$ . (а) Запишите матрицу этого преобразования. (б) Убедитесь, что с помощью полученной матрицы можно получить корректные результаты преобразования точек  $[1, 1]$ ,  $[1, 0]$  и  $[2, 1]$ .

**Поворот, масштабирование и перенос** Типичный пример использования этих преобразований показан на рис. 11.5. Изображение  $\mathbf{I}[\mathbf{r}, \mathbf{c}]$  получено с помощью камеры с квадратными пикселями. Оптическая ось камеры направлена перпендикулярно плоскости рабочего пространства  $\mathbf{W}[\mathbf{x}, \mathbf{y}]$ . Требуется получить формулу преобразования пиксельных координат  $[\mathbf{r}, \mathbf{c}]$  в координаты  $[x, y]$ , измеряемые, например, в миллиметрах. Это преобразование может быть выражено с помощью комбинации матриц поворота  $\mathbf{R}$ , масштабирования  $\mathbf{S}$  и переноса  $\mathbf{D}$ . Искомое преобразование в матричной форме представлено



**Рис. 11.5.** Изображение, полученное с помощью камеры с квадратными пикселями, оптическая ось которой направлена перпендикулярно наблюдаемой плоскости в рабочем пространстве. Для перехода из пиксельной системы координат изображения в мировую систему координат необходимо применить к координатам признаков точек преобразования поворота, масштабирования и переноса.

в уравнении 11.6. Формула преобразования координат также может быть записана в виде  ${}^w\mathbf{P}_j = \mathbf{D}_{x_0, y_0} \mathbf{S}_s \mathbf{R}_\theta {}^i\mathbf{P}_j$ .

Отображение пиксельных координат на координаты  $[x, y]$  на плоскости в рабочем пространстве определяется четырьмя параметрами. Это угол поворота  $\theta$ , масштабный коэффициент  $s$  для перехода от пиксельных единиц измерения к миллиметрам, и два параметра переноса  $x_0$  и  $y_0$ . Значения этих четырех параметров можно вычислить по координатам двух *управляющих точек*  $\mathbf{P}_1$  и  $\mathbf{P}_2$ . Управляющие точки соответствуют некоторым явно заметным и легко обнаружимым характерным признакам в рабочем пространстве. На рис. 11.5 эти точки отмечены символами '+'.  
 При обработке фотографий земной поверхности в качестве управляющих точек часто выбираются пересечения дорог, углы зданий, резкие изгибы рек и т. п. Важно отметить, что одна и та же признаковая точка, например,  $\mathbf{P}_1$ , может быть представлена двумя (и более) различными векторами. Например, один из них представляет пиксельные координаты точки на изображении  $\mathbf{I}$ , а второй — координаты  $[x, y]$  в мировой системе координат  $\mathbf{W}$ , измеренные в миллиметрах. Будем обозначать эти представления точки как  ${}^i\mathbf{P}_1$  и  ${}^w\mathbf{P}_1$ , соответственно. Например, на рис. 11.5 имеем  ${}^i\mathbf{P}_1 = [100, 60]$  и  ${}^w\mathbf{P}_1 = [200, 100]$ .

**84 Определение.** **Управляющими точками** (control points) называются явно различимые и легко обнаружимые точки, используемые для уста-

новления известных соответствий между различными координатными пространствами.

Если известны координаты точки  $\mathbf{P}_1$  в обеих системах координат, то с помощью матричного уравнения 11.6 можно записать два уравнения с четырьмя неизвестными (11.7) и (11.8).

$$\begin{bmatrix} x_w \\ y_w \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \quad (11.6)$$

$$x_w = x_i s \cos \theta - y_i s \sin \theta + x_0 \quad (11.7)$$

$$y_w = x_i s \sin \theta + y_i s \cos \theta + y_0 \quad (11.8)$$

Точка  $\mathbf{P}_2$  позволяет получить еще два уравнения. Для определения значений четырех параметров формулы преобразования координат необходимо решить полученную систему четырех уравнений.

Угол  $\theta$  легко вычислить независимо от других параметров следующим образом. В системе координат  $\mathbf{I}$  направление вектора  $\mathbf{P}_1\mathbf{P}_2$  определяется как  $\theta_i = \arctg((y_2 - y_1)/(x_2 - x_1))$ . В системе координат  $\mathbf{W}$  этот вектор направлен под углом  $\theta_w = \arctg((y_w - y_{w1})/(x_w - x_{w1}))$ . Искомый угол поворота равен разности двух этих углов:  $\theta = \theta_w - \theta_i$ . После вычисления  $\theta$  в решаемые уравнения можно подставить значения  $\sin \theta$  и  $\cos \theta$ . В результате останутся три уравнения с тремя неизвестными, которые легко решить относительно неизвестных  $s$ ,  $x_0$  и  $y_0$ . Читатель сможет завершить это решение, выполнив Упражнение 11.6.

---

### Упражнение 11.6. Преобразование пиксельных координат в мировые

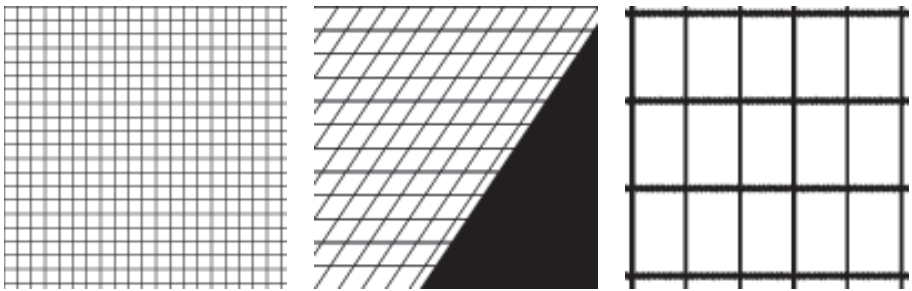
Рассмотрим пример, представленный на рис. 11.5. (Возможно, он относится к разработке зрительной системы манипуляционного робота для определения местоположения объектов в рабочем пространстве.) Запишите в матричной форме преобразование, связывающее координаты в системе координат изображения  $[x_i, y_i, 1]$  с координатами в мировой системе  $[x_w, y_w, 1]$ . Вычислите четыре параметра преобразования с помощью следующих управляющих точек:  ${}^i\mathbf{P}_1 = [100, 60]$ ,  ${}^w\mathbf{P}_1 = [200, 100]$ ;  ${}^i\mathbf{P}_2 = [380, 120]$ ,  ${}^w\mathbf{P}_2 = [300, 200]$ .

**Пример: аффинная деформация** Для извлечения из цифрового изображения фрагмента, имеющего форму параллелограмма, достаточно указать три точки. Первая точка определяет начало выходного изображения и одну из вершин параллелограмма, а вторая и третья точки задают две вершины параллелограмма, смежные первой. Выходное изображение выглядит как прямоугольный пиксельный массив произвольного размера, заполненный выборочными значениями исходного изображения. На рис. 11.6 показаны

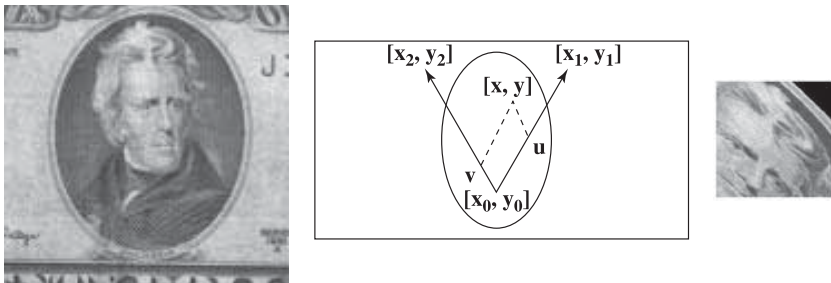
результаты работы программы, в которой была реализована рассмотренная операция деформации изображения. Для получения центрального изображения (рис. 11.6) использовались три точки, определяющие неперпендикулярные оси. Поэтому на выходном изображении заметен сдвиг сетки. На третьем изображении этот сдвиг был удален. Для этого было сформировано изображение, оси которого совмещены с наклонными осями, использовавшимися при формировании центрального изображения.

На рис. 11.7 представлен еще один пример. Это деформированный портрет президента Джексона, извлеченный из изображения банкноты достоинством 20\$ (см. рис. 11.7). В обоих примерах из исходного изображения извлекается только некоторая часть, но размеры исходного и выходного изображений совпадают.

Программа, с помощью которой был получен рис. 11.7, использует три указанные пользователем точки для преобразования фрагмента в форме па-



**Рис. 11.6.** (Слева) цифровое изображение сетки. Размеры изображения равны  $128 \times 128$  пикселей. (В центре) изображение размерами  $128 \times 128$ , которое было получено с помощью аффинной деформации, заданной тремя точками на левом изображении. (Справа) изображение размерами  $128 \times 128$  с восстановленным фрагментом центрального изображения.



**Рис. 11.7.** Деформированный портрет президента Джексона, извлеченный из изображения банкноты достоинством 20\$. Было применено аффинное преобразование сдвига.

параллелограмма, заданного на исходном изображении. Выходное изображение содержит  $n \times m$  (например,  $512 \times 512$ ) пикселей с координатами  $[r, c]$ . Для каждого пикселя  $[r, c]$  выходного изображения производится выборка пикселя исходного изображения с координатами  $[x, y]$ . Координаты выборочного пикселя вычисляются с помощью преобразования (11.9). Это преобразование сначала записано в наглядном виде на основе базисных векторов, а затем в стандартной матричной форме.

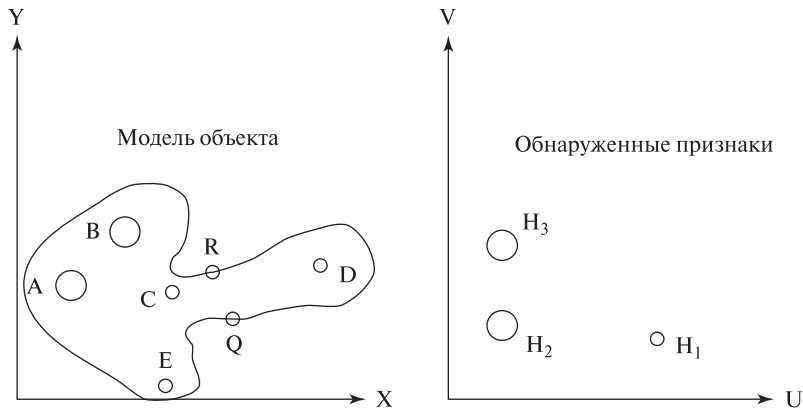
$$\begin{aligned} \begin{bmatrix} x \\ y \end{bmatrix} &= \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \frac{r}{n} \left( \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} - \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \right) + \frac{c}{m} \left( \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} - \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \right) \\ \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} &= \begin{bmatrix} (x_1 - x_0)/n & (x_2 - x_0)/m & x_0 \\ (y_1 - y_0)/n & (y_2 - y_0)/m & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r \\ c \\ 1 \end{bmatrix} \end{aligned} \quad (11.9)$$

В сущности, координаты точки  $[x, y]$  определяются с помощью единичных векторов новых осей, которые были заданы указанными пользователем точками. Вычисленные координаты  $[x, y]$  необходимо округлить для получения целочисленных координат пикселя на цифровом изображении  ${}^1\mathbf{I}$ . Если значения  $x$  или  $y$  выходят за границы изображения, то соответствующему пикселу выходного изображения присваивается значение черного цвета, т. е.  ${}^2\mathbf{I}[r, c] = 0$ . В противном случае  ${}^2\mathbf{I}[r, c] = {}^1\mathbf{I}[x, y]$ . В верхнем правом углу деформированного изображения на рис. 11.7 заметен черный треугольник, так как в данной области выборочный параллелограмм выходит за пределы исходного изображения банкноты.

**Пример: распознавание и определение местоположения объекта**  
Рассмотрим пример вычисления преобразования для сопоставления модели объекта, показанной на рис. 11.8 (слева), и объекта на изображении (рис. 11.8 (справа)).

Будем полагать, что в результате выделения характерных признаков были обнаружены только центры трех отверстий объекта. Пространственное преобразование должно отображать модельные точки  $[x, y]$  на точки изображения  $[u, v]$ . Допустим, что изображения формируются в контролируемой среде и нам известен масштабный коэффициент. Также будем считать, что к пиксельным координатам уже было применено преобразование масштабирования и в результате были получены координаты  $u-v$ .

Для получения параметров поворота и переноса, совмещающих все модельные точки с соответствующими точками изображения, достаточно знать две точки изображения. В таблицах 11.1 и 11.2 приведены координаты и расстояния между точками модели и изображения. Для вычисления параметров преобразования примем гипотезу о соответствии пар точек  $(A, H_2)$  и  $(B, H_3)$ . Отметим, что эта гипотеза согласуется с известными расстояниями между



**Рис. 11.8.** (Слева) модель объекта. (Справа) три отверстия, обнаруженные на изображении.

**Таблица 11.1.** Местоположение и расстояния между точками модели (меткой † обозначены центры отверстий)

Точка	Координаты †	до точки A	до точки B	до точки C	до точки D	до точки E
A	(8,17)	0	12	15	37	21
B	(16,26)	12	0	12	30	26
C	(23,16)	15	12	0	22	15
D	(45,20)	37	30	22	0	30
E	(22,1)	21	26	15	30	0

**Таблица 11.2.** Местоположение и расстояния между точками изображения (меткой † обозначены центры отверстий)

Точка	Координаты †	до точки H <sub>1</sub>	до точки H <sub>2</sub>	до точки H <sub>3</sub>
H <sub>1</sub>	(31,9)	0	21	26
H <sub>2</sub>	(10,12)	21	0	12
H <sub>3</sub>	(10,24)	26	12	0

точками. Алгоритмы для формирования подобных гипотез будут обсуждаться в разд. 11.5.

Направление вектора между модельными точками  $A$  и  $B$  равно  $\theta_1 = \text{arctg}(9.0/8.0) = 0.844$ . На изображении соответствующий вектор между точками  $H_2$  и  $H_3$  направлен под углом  $\theta_2 = \text{arctg}(12.0/0.0) = \pi/2 = 1.571$ . Следовательно, угол поворота равен  $\theta = 0.727$  радиан. Подставим в уравнение 11.6 известные координаты соответствующих точек модели и изображения  $A$  и  $H_2$ .

Получим следующую систему уравнений, где  $u_0, v_0$  являются неизвестными параметрами переноса на плоскости изображения. Отметим, что значения  $\sin \theta$  и  $\cos \theta$  известны, так как угол  $\theta$  уже был вычислен.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} 10 \\ 12 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & u_0 \\ \sin \theta & \cos \theta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 8 \\ 17 \\ 1 \end{bmatrix} \quad (11.10)$$

Полученная система двух линейных уравнений имеет решение  $u_0 = 15.3$  и  $v_0 = -5.95$ . Для проверки можно воспользоваться парой соответствующих точек  $B$  и  $H_3$ . Для них получаются похожие параметры переноса. При вычислении параметров преобразования по различным парам соответствующих точек будут получаться немного отличающиеся значения. Далее в этой главе обсуждаются методы расчета параметров преобразования с использованием большего количества точек, что обеспечивает более точное сопоставление двумерных точек.

После определения параметров пространственного преобразования можно вычислить координаты любых модельных точек на плоскости изображения. Например, модельные точки на «рукоятке» объекта имеют координаты  $R = [29, 19]$  и  $Q = [32, 12]$ . Как показано в выражении (11.11), модельная точка  $R$  преобразуется в точку изображения  ${}^iR = [24.4, 27.4]$ . Для модельной точки  $Q = [32, 12]$  преобразование дает точку изображения  ${}^iQ = [31.2, 24.2]$ .

$$\begin{bmatrix} u_R \\ v_R \\ 1 \end{bmatrix} = \begin{bmatrix} 24.4 \\ 27.4 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 15.3 \\ \sin \theta & \cos \theta & -5.95 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 29 \\ 19 \\ 1 \end{bmatrix} \quad (11.11)$$

В примере с манипуляционным роботом подобная информация могла бы использоваться для выполнения захвата видимого на изображении объекта. Известные параметры преобразований позволяют вычислить по точкам на изображении координаты соответствующих точек объекта в мировой системе координат. Конечно, схват робота должен быть раскрыт немного шире, чем толщина рукоятки объекта, вычисленная по точкам  ${}^iR$  и  ${}^iQ$ . Результаты вычислений могут быть неточными вследствие искажений изображения, неточностей при определении характерных признаков и ошибок вычислений. Операции робота по захвату объекта планируются в непрерывном рабочем пространстве с вещественными координатами, хотя координаты на изображении принимают дискретные значения.

Пиксельные данные изображения определены только в узлах дискретной целочисленной сетки. Допустим, для модели на рис. 11.8 требуется проверить наличие отверстий  ${}^iC$  и  ${}^iD$  посредством поиска светлых пикселей в области отверстий на изображении. Для определения центров отверстий на изображении координаты преобразованных модельных точек необходимо округлить до целых значений. Или, возможно, на изображении следует проверять не



один пиксел в центре отверстия, а некоторую окрестность с центром в точке с вещественными преобразованными координатами. В данном примере можно увидеть возможность распознавания двумерных объектов на основе совмещения точек модели с признаковыми точками на изображении объекта.

**85 Определение.** Распознаванием посредством совмещения (recognition-by-alignment) называется метод распознавания объекта, в котором с помощью преобразований поворота, масштабирования и переноса выполняется сопоставление преобразованных характерных признаков модели с признаками изображения.

**Упражнение 11.7.** Коммутативность преобразований.

Предположим, даны матрицы трех элементарных преобразований:  $\mathbf{R}_\theta$  для поворота относительно начала координат,  $\mathbf{S}_{s_x, s_y}$  для масштабирования и  $\mathbf{D}_{x_0, y_0}$  для переноса. (а) Коммутативны ли преобразования масштабирования и переноса? То есть всегда ли выполняется  $\mathbf{S}_{s_x, s_y} \mathbf{D}_{x_0, y_0} = \mathbf{D}_{x_0, y_0} \mathbf{S}_{s_x, s_y}$ ? (б) Коммутативны ли преобразования поворота и масштабирования? То есть всегда ли выполняется равенство  $\mathbf{R}_\theta \mathbf{S}_{s_x, s_y} = \mathbf{S}_{s_x, s_y} \mathbf{R}_\theta$ ? (в) Коммутативны ли преобразования поворота и переноса? (г) Коммутативны ли преобразования масштабирования и переноса? Для получения ответа и пояснений пользуйтесь геометрическими и алгебраическими методами.

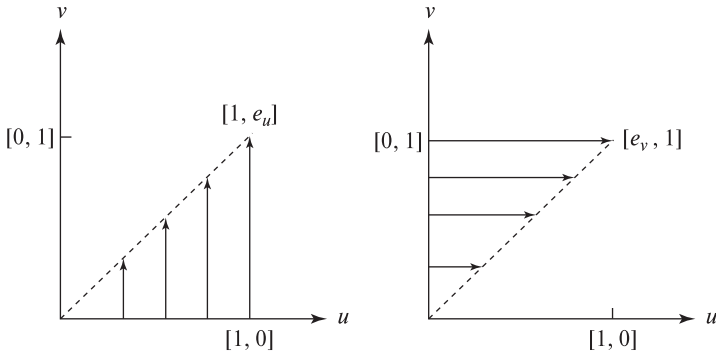
**Упражнение 11.8**

Запишите матрицу симметрии относительно прямой  $y = 3$  в виде композиции матрицы переноса с параметром  $y_0 = -3$  и матрицы симметрии относительно координатной оси  $OX$ . Убедитесь в корректности полученной матрицы на примере преобразования трех точек  $[1, 1]$ ,  $[1, 0]$  и  $[2, 1]$ . Изобразите исходные и преобразованные точки на графике.

**Упражнение 11.9**

Проверьте, что произведение матриц  $\mathbf{D}_{x_0, y_0}$  и  $\mathbf{D}_{-x_0, -y_0}$  равно единичной матрице размерности  $3 \times 3$ . Чем это объясняется?

**Аффинные преобразования общего вида\*** Выше были рассмотрены частные виды двумерных аффинных преобразований — поворот, масштабирование и перенос (rotation, scaling, translation — RST). Еще одной разновидностью преобразований является сдвиг (shear). Действие этого преобразования поясняется на рис. 11.9. Точки координатной плоскости  $u$ - $v$  сдвигаются вдоль оси  $v$  пропорционально их расстоянию от оси  $v$ . Таким образом, точка  $[u, v]$  преобразуется в точку  $[u, e_u u + v]$  при сдвиге вдоль оси  $v$  или в точку



**Рис. 11.9.** (Слева) сдвиг вдоль оси  $v$ . (Справа) сдвиг вдоль оси  $u$ .

$[u + e_v v, v]$  при сдвиге вдоль оси  $u$ . Матричные уравнения этих преобразований имеют вид (11.12) и (11.13). Напомним, что векторы-столбцы матрицы сдвига можно получить посредством применения этих преобразований к базисным векторам.

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ e_u & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad (11.12)$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & e_v & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad (11.13)$$

Выше уже был рассмотрен пример использования преобразования сдвига для получения изображения наклонной сетки на рис. 11.6.

В качестве пятой разновидности аффинных преобразований рассмотрим симметрию относительно оси. Симметрия относительно оси  $u$  отображает базисные векторы  $[1, 0], [0, 1]$  на  $[1, 0], [0, -1]$ , а симметрия относительно оси  $v$  отображает базисные векторы  $[1, 0], [0, 1]$  на  $[-1, 0], [0, 1]$ . Эти преобразования легко записать с помощью матриц размерами  $2 \times 2$  или  $3 \times 3$ .

Любое аффинное преобразование можно представить в виде композиции преобразований рассмотренных пяти типов — поворотов, масштабирований, переносов, сдвигов и симметрий. Каждое из этих преобразований имеет обратное преобразование того же типа. Очевидно, что матрица аффинного преобразования общего вида, которую можно записать в виде композиции перечисленных преобразований, содержит 6 параметров. Аффинное преобразование общего вида показано в уравнении (11.14). Шесть параметров преобразования могут быть определены по трем парам соответствующих неколлинеарных точек. Для этого надо решить систему из шести линейных уравнений, которая получается

путем записи матричного уравнения (11.14) для каждой пары точек.

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad (11.14)$$

#### 11.4 НАИЛУЧШЕЕ ДВУМЕРНОЕ АФФИННОЕ ПРЕОБРАЗОВАНИЕ\*

Двумерное аффинное преобразование общего вида, описывающее отображение одного двумерного пространства на другое, содержит 6 параметров (см. уравнение (11.15)). Значения этих параметров можно вычислить по 3 парам сопоставленных точек  $([x_j, y_j], [u_j, v_j])_{j=1,3}$ .

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (11.15)$$

Ошибка при определении координат любой из этих точек приводит к существенным ошибкам в значениях параметров преобразования. Более надежный подход состоит в вычислении значений 6 параметров преобразования методом наименьших квадратов с использованием большего количества пар сопоставленных управляющих точек. Функцию ошибки определим аналогично задаче аппроксимации точек данных с помощью прямой, рассматривавшейся в гл. 10:

$$\begin{aligned} \varepsilon(a_{11}, a_{12}, a_{13}, a_{21}, a_{22}, a_{23}) = & \sum_{j=1}^n ((a_{11}x_j + a_{12}y_j + a_{13} - u_j)^2 \\ & + (a_{21}x_j + a_{22}y_j + a_{23} - v_j)^2) \end{aligned} \quad (11.16)$$

Вычислим шесть частных производных функции ошибки  $\partial\varepsilon/a_{ij}$  по каждому из 6 неизвестных параметров  $a_{ij}$ . Приравняем эти производные к нулю и получим 6 уравнений, записанных в матричной форме (11.17).

$$\begin{bmatrix} \Sigma x_j^2 & \Sigma x_j y_j & \Sigma x_j & 0 & 0 & 0 \\ \Sigma x_j y_j & \Sigma y_j^2 & \Sigma y_j & 0 & 0 & 0 \\ \Sigma x_j & \Sigma y_j & \Sigma 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \Sigma x_j^2 & \Sigma x_j y_j & \Sigma x_j \\ 0 & 0 & 0 & \Sigma x_j y_j & \Sigma y_j^2 & \Sigma y_j \\ 0 & 0 & 0 & \Sigma x_j & \Sigma y_j & \Sigma 1 \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{21} \\ a_{22} \\ a_{23} \end{bmatrix} = \begin{bmatrix} \Sigma u_j x_j \\ \Sigma u_j y_j \\ \Sigma u_j \\ \Sigma v_j x_j \\ \Sigma v_j y_j \\ \Sigma v_j \end{bmatrix} \quad (11.17)$$

#### Упражнение 11.10

Решите систему уравнений (11.17) с использованием следующих трех пар сопоставленных управляющих точек:  $([0, 0], [0, 0])$ ,  $([1, 0], [0, 2])$ ,  $([0, 1], [-2, 0])$ . Совпадают ли результаты вычислений с оценками на основе преобразования базисных векторов?

---

### Упражнение 11.11

Решите систему уравнений (11.17) с использованием следующих трех пар сопоставленных управляющих точек:  $([0, 0], [1, 2])$ ,  $([1, 0], [3, 2])$ ,  $([0, 1], [1, 4])$ . Совпадают ли результаты вычислений с оценками на основе преобразования базисных векторов?

Обычно при определении соответствия между изображением и картой или между двумя изображениями используется большое количество управляющих точек. На рис. 11.10 показаны два изображения примерно одной и той же сцены. В нижней части рисунка перечислены одиннадцать пар сопоставленных управляющих точек. Управляющие точки соответствуют углам объектов, которые уникальным образом обнаруживаются на обоих изображениях (или картах). В данном примере управляющие точки были указаны пользователем с помощью манипулятора «мышь». Список невязок показывает, что при использовании полученной матрицы преобразования, ни одна из координат  $u$  или  $v$  точек на правом изображении не отклоняется от преобразованных координат точек левого изображения больше чем на два пиксела. Большинство невязок не превышают одного пиксела.

Более точные результаты могли бы быть получены с помощью автоматического обнаружения характерных признаков и вычисления координат признаков точек с субпиксельной точностью. При указании управляющих точек человеком вручную с помощью манипулятора «мышь» координаты управляющих точек обычно задаются с точностью до одного-двух пикселей.

После вычисления параметров аффинного преобразования можно было бы выполнить на правом изображении поиск объектов, местоположение которых известно на левом изображении.

Теперь становится более ясным, в чем могла бы заключаться обработка изображений аэрофотосъемки в задаче учета городских построек. Потребовалось бы определять соответствие между изображением аэрофотосъемки и картой, чтобы найти новые здания и обновить информацию о них в базах данных.

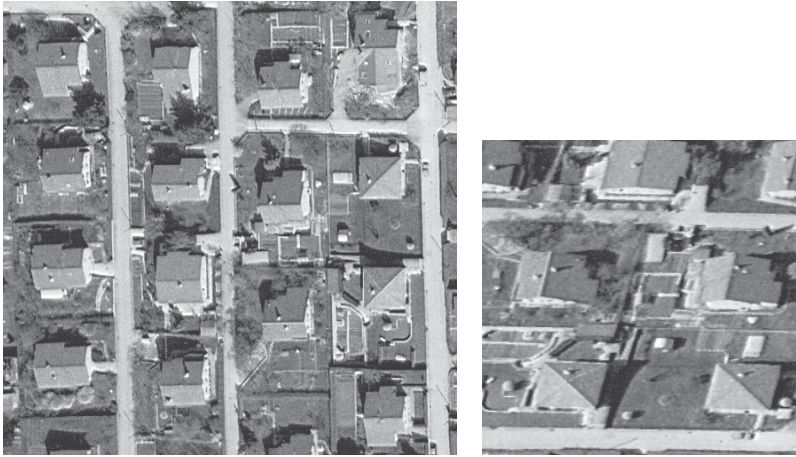
---

### Упражнение 11.12

Выберите на рис. 11.10 три пары сопоставленных управляющих точек (например,  $([288, 210, 1], [31, 160, 1])$ ) и проверьте, что приведенная на рис. 11.10 матрица аффинного преобразования отображает первую точку на вторую.

## 11.5 РАСПОЗНАВАНИЕ ДВУМЕРНЫХ ОБЪЕКТОВ С ИСПОЛЬЗОВАНИЕМ АФФИННЫХ ПРЕОБРАЗОВАНИЙ

В данном разделе будут рассмотрены несколько методов распознавания двумерных объектов посредством отображения точек модели на точки изобра-



====Программа для определения параметров наилучшего двумерного====  
 ====аффинного преобразования=====

Пары сопоставленных управляющих точек:

288	210	31	160	232	288	95	205	195	372	161	229	269	314	112	159
203	424	199	209	230	336	130	196	284	401	180	124	327	428	198	69
284	299	100	146	337	231	45	101	369	223	38	64				

Матрица преобразования:

```
[ -0.0414 ,  0.773 , -119
  -1.120  , -0.213 , 526 ]
```

Невязки (в пикселах) для 22 уравнений:

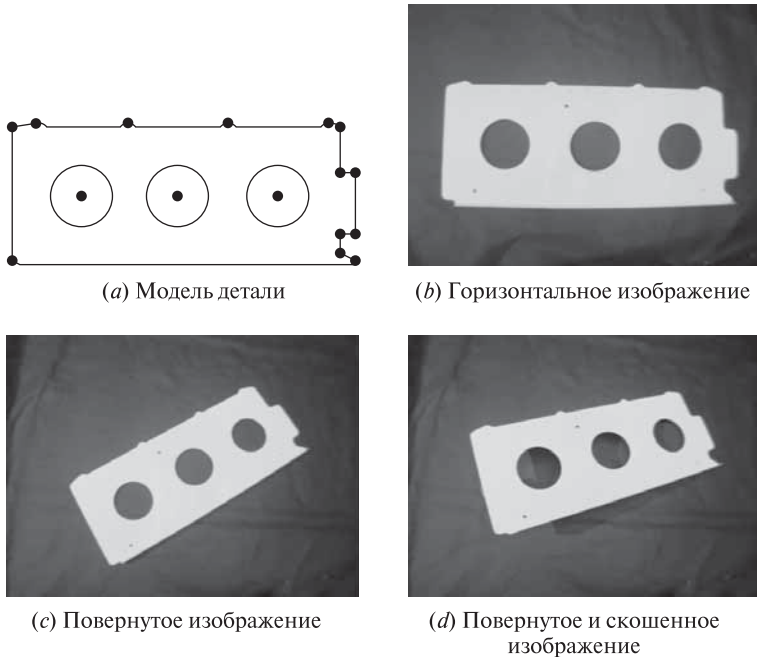
0.18	-0.68	-1.22	0.47	-0.77	0.06	0.34	-0.51	1.09	0.04	0.96
1.51	-1.04	-0.81	0.05	0.27	0.13	-1.12	0.39	-1.04	-0.12	1.81

====Программа завершила работу=====

**Рис. 11.10.** Изображения одной и той же сцены и параметры наилучшего аффинного преобразования, отображающего левое изображение на правое. Параметры преобразования определялись по 11 управляющим точкам. Координаты  $[x, y]$  на левом изображении заданы в системе, в которой значения  $x$  увеличиваются сверху-вниз, а  $y$  — слева-направо. Координаты  $[u, v]$  на правом изображении заданы в системе, в которой значения  $u$  увеличиваются сверху-вниз, а  $v$  — слева-направо. Под изображением приведены 11 пар координат, представляющих управляющие точки для сопоставления  $x, y, u, v$ . Можете ли вы сопоставить характерные признаки на двух этих изображениях? (С любезного разрешения Oliver Faugeras.)

жения. Один из этих методов (распознавание посредством совмещения) уже упоминался в разделе, посвященном аффинным преобразованиям. В методах общего вида используются произвольные признаковые точки. Однако в большинстве предметных областей характерные признаки можно классифицировать и присвоить признаковым точкам метки, указывающие на их тип. На-

пример, в процедуре сортировки частей объектов метки могли бы обозначать углы или центры отверстий, а в задаче обработки карт — точки пересечения контуров или точки с большими значениями кривизны на границах водных поверхностей.



**Рис. 11.11.** Двумерная модель детали самолета и 3 изображения, которые могут быть сопоставлены с моделью.

Принцип распознавания объектов по известным моделям поясняется на рис. 11.11. На рис. 11.11 (a) представлена контурная модель детали самолета. Небольшими черными кружками обозначены признаковые точки, которые можно использовать при сопоставлении. На рис. 11.11 (b) показано изображение настоящей детали самолета, ориентация которой примерно совпадает с ориентацией модели. Деталь на втором изображении на рис. 11.11 (c) повернута на 45 градусов. При получении третьего изображения на рис. 11.11 (d) камера была наклонена относительно плоскости детали, поэтому ее образ выглядит скошенным. Методы, которые рассматриваются в данном разделе, позволяют определить, содержит ли некоторое изображение (например, рис. 11.11 (b), (c), (d)) образ объекта, соответствующего известной модели (рис. 11.11 (a)). В случае успешного обнаружения также могут быть вычислены параметры расположения объекта относительно камеры.

**Метод базовых локальных признаков** При распознавании объекта с помощью метода базовых локальных признаков (local-feature-focus method) учитываются локальные характерные признаки объекта и их взаимное расположение. Заранее формируется множество моделей объектов, по одной модели для каждого распознаваемого объекта. Каждая модель содержит множество *базовых признаков* (focus features). Это главные характерные признаки объекта. Они должны быть легко обнаружимы, если только не перекрываются другими объектами. Для каждого базового признака в модели хранится множество близкорасположенных характерных признаков. Они используются для подтверждения корректного обнаружения базового признака и для определения местоположения и ориентации объекта.

На этапе сопоставления на изображении одного или нескольких объектов производится выделение характерных признаков. Алгоритм сопоставления сначала выполняет поиск базовых признаков. При обнаружении базового признака, принадлежащего некоторой модели, алгоритм пытается обнаружить в окрестности базового признака близкорасположенные признаки изображения. Если на изображении удастся обнаружить такие признаки, похожие на модельные близкорасположенные признаки, то между этим небольшим множеством признаков изображения и модели устанавливается соответствие и алгоритм формирует гипотезу о присутствии на изображении известного объекта. Для подтверждения этой гипотезы выполняется процедура верификации.

Процедура верификации должна определить, достаточно ли оснований для того, чтобы считать, что предполагаемый объект присутствует в наблюдаемой сцене. При распознавании объектов с плоскими гранями в процедуре верификации часто используются границы объекта. С учетом множества обнаруженных соответствующих характерных признаков вычисляются параметры аффинного преобразования, отображающего модельные точки на точки изображения. Это преобразование применяется для преобразования координат каждого прямолинейного сегмента границы модели объекта в координаты концов отрезка на изображении. Преобразованные модельные отрезки должны примерно совпадать с отрезками на изображении, если только объект не перекрывается какими-либо другими объектами. Вследствие шума на изображении и ошибок в процессе выделения и сопоставления признаков, маловероятно, что модельные отрезки после преобразования будут точно совмещены с отрезками на изображении. Поэтому поиск похожих отрезков изображения можно выполнять в прямоугольной области вокруг каждого преобразованного модельного отрезка. Если удастся найти существенные признаки совпадения, то модельный отрезок помечается как подтвержденный. В случае подтверждения достаточного количества модельных отрезков объект считается обнаруженным. Расположение объекта на изображении задается параметрами вычисленного аффинного преобразования.

Ниже приведен алгоритм для сопоставления заданной модели  $\mathbf{F}$  и изображения методом базовых локальных признаков. В модели определено множество базовых признаков  $\{\mathbf{F}_1, \mathbf{F}_2, \dots, \mathbf{F}_M\}$ . Для каждого базового признака

**Применение метода базовых локальных признаков для определения преобразования, отображающего модельные признаки на признаки изображения.**

$G_i, i = 1, I$  — множество обнаруженных характерных признаков изображения.

$F_m, m = 1, M$  — множество базовых характерных признаков модели объекта.

$S(f)$  — множество близкорасположенных признаков для каждого признака  $f$ .

```

procedure local_feature_focus(G, F);
{
  for каждый базовый признак  $F_m$ 
    for каждый признак изображения  $G_i$ , однотипный с признаком  $F_m$ 
      {
        Поиск в  $S(F_m)$  максимального подграфа  $S_m$ ,
        который соответствует подграфу  $S_i$  из  $S(G_i)$ ;
        Вычисление параметров преобразования  $T$ , отображающего
        точки каждого признака из  $S_m$  на точки соответствующего
        признака из  $S_i$ ;
        Применение преобразования  $T$  к сегментам границы модели;
        if на изображении обнаружено достаточное количество
        граничных сегментов then return( $T$ );
      } ;
} ;

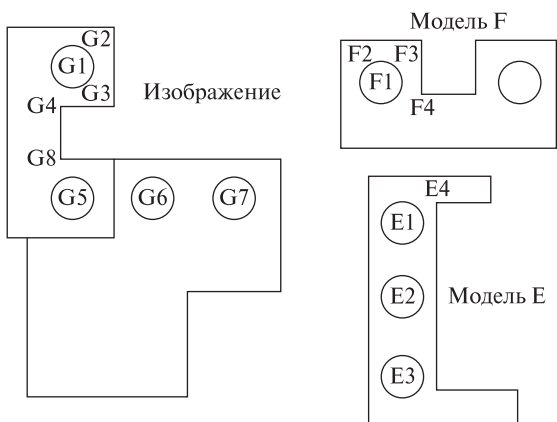
```

**Алгоритм 11.1.** Метод базовых локальных признаков.

$F_m$  также определено множество близкорасположенных признаков  $S(F_m)$ , которые можно использовать для подтверждения корректного обнаружения базового признака. Характерные признаки, обнаруженные на изображении, представлены в виде множества  $\{G_1, G_2, \dots, G_I\}$ . Для каждого признака  $G_i$  определено множество близкорасположенных признаков изображения  $S(G_i)$ .

Метод базовых локальных признаков поясняется на рис. 11.12 на примере поиска на изображении двух моделей E и F. В качестве характерных признаков были обнаружены круглые отверстия и углы. Была получена гипотеза о соответствии локального признака F1 модели F и признака изображения G1. Для близкорасположенных модельных признаков F2, F3 и F4 были найдены соответствующие близкорасположенные признаки изображения G2, G3 и G4. Присутствие на изображении экземпляра модели F подтвердилось на этапе верификации. При обработке второй модели E была сформирована гипотеза о соответствии между признаком E1 и его близкорасположенными признаками





**Рис. 11.12.** Метод базовых локальных признаков. На изображении экземпляр модели F перекрывает другой объект.

$E_2$ ,  $E_3$  и  $E_4$  и признаками изображения  $G_5$ ,  $G_6$ ,  $G_7$ ,  $G_8$ . Однако эта гипотеза была отвергнута, так как в процессе верификации границу модели  $E$  не удалось достаточно хорошо сопоставить с отрезками на изображении.

**Кластеризация по расположению** Выше было показано, что совмещение характерных признаков модели и изображения может быть выполнено с помощью аффинного преобразования, представимого в виде композиции поворота, переноса и масштабирования. Параметры этого RST-преобразования могут быть определены по двум сопоставленным управляющим точкам. Если двум точкам изображения были сопоставлены две точки модели, то параметры преобразования можно вычислить с помощью уравнения (11.6). Автоматическое определение управляющих точек является непростой задачей, так как нередко возможны неоднозначные варианты сопоставления. В методе кластеризации по расположению аффинное преобразование для совмещения модели и изображения вычисляется для всех возможных пар управляющих точек. Затем на множестве параметров преобразований выполняется поиск кластера, содержащего параметры с близкими значениями. Кластер в параметрическом пространстве удастся обнаружить, если у модели и изображения имеется много сопоставленных признаков точек. Основные действия алгоритма кластеризации по расположению будут описаны далее.

**86 Определение.** Предположим, что задано пространственное преобразование  $T$ , совмещающее модель  $M$  с образом объекта  $O$  на изображении  $I$ . Параметры местоположения и ориентации объекта  $O$ , определяемые параметрами  $\alpha$  преобразования  $T$ , называются **расположением** (pose) объекта.

Обработка всех возможных пар признаков точек приводит к значительным вычислительным затратам. В задаче сопоставления изображений аэрофотосъемки с картами могут использоваться точки пересечения дорог или углы некоторых областей (например, полей). Пересечения сегментов можно

классифицировать и учитывать их тип при сопоставлении. Например, на изображениях часто встречаются пересечения сегментов типа 'L', 'Y', 'T', 'стрелка' и 'X' (рис. 11.13).

Предположим, что при сопоставлении учитываются только пары типа LX или TY. На рис. 11.14 приведен пример данных для сопоставления, среди которых имеется 5 пар модельных точек и 4 пары точек изображения. Хотя среди них можно выбрать  $4 \times 5 = 20$  различных пар, только в 10 парах обе точки будут иметь допустимые типы. В таблице 11.3 перечислены параметры преобразования, вычисленные для каждой из этих пар. Параметры большинства из 10 преобразований сильно отличаются и не образуют кластеров в параметрическом пространстве, за исключением трех наборов, отмеченных в последнем столбце таблицы символами (\*). Эти три набора параметров образуют кластер со средними значениями  $\theta = 0.68, s = 2.01, u_0 = 233, v_0 = -41$ . Хотя для корректно сопоставленных точек можно было бы ожидать меньшего различия значений параметров, имеющиеся различия типичны вследствие небольших

### Применение метода кластеризации по расположению для определения преобразования, отображающего модельные признаки на признаки изображения.

$P_i, i = 1, D$  — множество обнаруженных на изображении характерных признаков.

$L_j, j = 1, M$  — множество характерных признаков модели.

**procedure** pose\_clustering( $P, L$ );

{

  for каждая пара признаков точек изображения ( $P_i, P_j$ )

    for каждая пара признаков точек модели ( $L_m, L_n$ ) того же типа

      {

        Вычисление параметров  $\alpha$  для преобразования,  
        отображающего пару точек ( $L_m, L_n$ ) на точки ( $P_i, P_j$ );

        Сохранение  $\alpha$  в параметрическом пространстве;

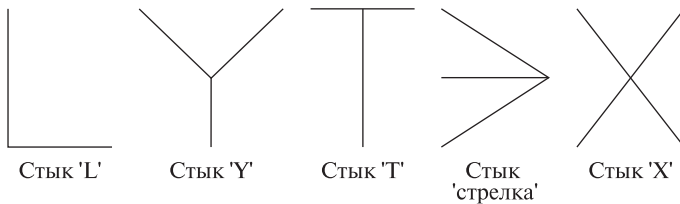
      };

  Поиск в параметрическом пространстве всех кандидатов на кластеры  $\alpha$ ;

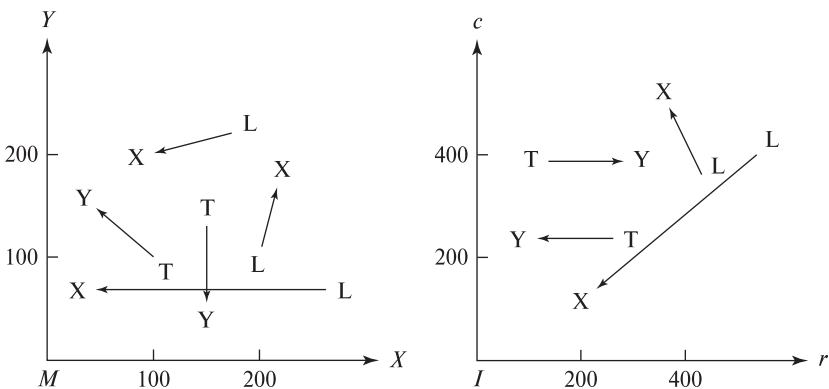
  Обработка каждого крупного кластера посредством отображения всех модельных признаков точек на точки изображения и проверка содержимого изображения в окрестностях этих точек;

**return**(подтвержденный набор параметров  $\{\alpha_k\}$ );

}



**Рис. 11.13.** Разновидности пересечений отрезков, часто используемые в процедурах сопоставления.



**Рис. 11.14.** Пример задачи определения расположения по пяти парам модельных признаков точек и четырем парам признаков точек изображения.

ошибок при определении местоположения признаков точек и нелинейных искажений в процессе формирования изображения.

Если параметры преобразования RST были вычислены с достаточной точностью, то их можно использовать для верификации сопоставленных точек. Эти точки затем можно использовать в качестве управляющих точек для вычисления нелинейного или аффинного преобразования (с большим количеством параметров, чем у преобразования RST), которое будет обеспечивать более высокую точность сопоставления управляющих точек.

При кластеризации по расположению могут использоваться низкоуровневые характерные признаки. Однако при фильтрации признаков по типу точность результатов повышается, а временные затраты уменьшаются.

Кластеризация может быть выполнена с помощью простого алгоритма с временными затратами  $O(n^2)$ . Для каждого текущего набора параметров  $\alpha$  определяется количество других наборов  $\alpha_i$ , которые близки к текущему набору по некоторому критерию расстояния. Расстояния между наборами параметров в кластерном пространстве вычисляются  $n - 1$  раз для каждого из  $n$  имеющихся наборов.

Альтернативный вариант организации вычислений предполагает применение дискретизации. Это хорошо известный традиционный подход, кото-

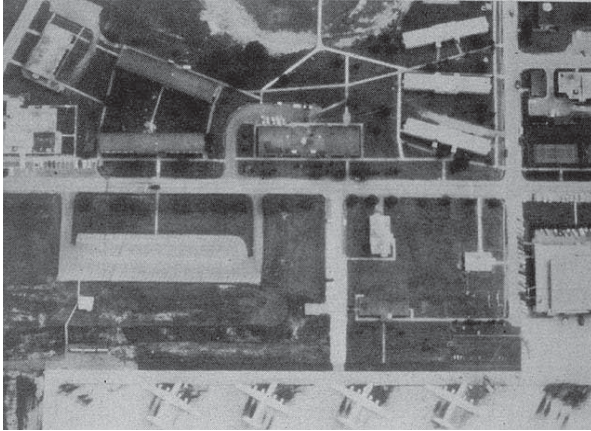
**Таблица 11.3.** Кластерное пространство, построенное по 10 наборам параметров расположения, которые были вычислены по данным на рис. 11.14

Пара модельных точек	Пара точек изображения	$\theta$	$s$	$u_0$	$v_0$
L(170,220),X(100,200)	L(545,400),X(200,120)	0.403	6.10	118	-1240
L(170,220),X(100,200)	L(420,370),X(360,500)	5.14	2.05	-97	514
T(100,100),Y( 40,150)	T(260,240),Y(100,245)	0.663	2.05	225	-48 *
T(100,100),Y( 40,150)	T(140,380),Y(300,380)	3.87	2.05	166	669
L(200,100),X(220,170)	L(545,400),X(200,120)	2.53	6.10	1895	200
L(200,100),X(220,170)	L(420,370),X(360,500)	0.711	1.97	250	-36 *
L(260, 70),X( 40, 70)	L(545,400),X(200,120)	0.682	2.02	226	-41 *
L(260, 70),X( 40, 70)	L(420,370),X(360,500)	5.14	0.651	308	505
T(150,125),Y(150, 50)	T(260,240),Y(100,245)	4.68	2.13	3	568
T(150,125),Y(150, 50)	T(140,380),Y(300,380)	1.57	2.13	407	60

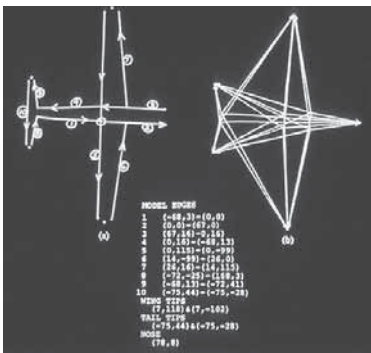
рый уже обсуждался в гл. 10 применительно к преобразованию Хафа. При этом обеспечивается более быстрая обработка, но ее сложнее адаптировать к конкретной задаче. Каждый полученный набор параметров учитывается в дискретном элементе параметрического пространства (в счетчике). После обработки всех наборов в параметрическом пространстве выполняется поиск существенных значений счетчиков. Если несколько похожих  $\alpha_i$  попадают в различные соседние разряды, то кластер может быть потерян.

Примером применения кластеризации служит задача обнаружения моделей самолетов на изображении аэрофотосъемки (рис. 11.15). С помощью методов, описанных в гл. 5 и 10, на изображении выделялись характерные признаки в виде краев и контурных точек с большой кривизной. Подмножества признаков сопоставлялись с моделью, приведенной на рис. 11.15 (b). На рис. 11.15 (c) показаны края, обнаруженные в одном из этих подмножеств, в котором многие признаки удалось совместить с признаками модели с помощью одинаковых параметров преобразования.

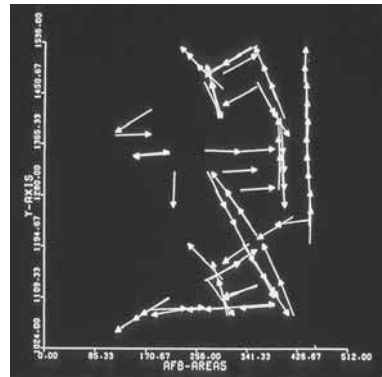
**Геометрическое хэширование** Метод базовых локальных признаков и алгоритм кластеризации по расположению были разработаны для сопоставления одной модели и изображения. В случае нескольких моделей различных объектов в этих методах требуется сопоставлять все модели по очереди. Поэтому рассмотренные методы не слишком удобны для задач с большим количеством различных объектов. Для работы с большими базами данных моделей был разработан метод геометрического хэширования. Этот метод требует большой предварительной обработки и большого объема памяти, но благодаря этому обеспечивается быстрое распознавание и определение параметров расположения объекта.



(a) Исходное изображение аэродрома



(b) Модель объекта



(c) Результаты сопоставления модели

**Рис. 11.15.** Кластеризация по расположению применялась для обнаружения заданных моделей самолетов. (a) Аэрофотосъемка аэродрома. (b) Модель объекта, определенная в терминах реально наблюдаемых краев и фиктивных краев, противоположащих одной угловой точке и одной концевой точке криволинейного сегмента. (c) Окно изображения, содержащее результаты обнаружения фактов успешного сопоставления многих частей модели с признаками изображения при использовании одного и того же преобразования. Воспроизведено с разрешения IEEE.

Предположим, что имеются

1. большая база данных моделей и
2. неизвестный объект, характерные признаки которого были выделены на изображении и для которого известно, что этот объект может быть найден посредством аффинного преобразования одной из моделей.

Требуется найти модель и параметры преобразования, совмещающего модель и объект на изображении.

Будем полагать, что модель  $\mathbf{M}$  задана в форме упорядоченного множества признаков точек. Любое подмножество трех неколлинеарных точек  $E = \{e_{00}, e_{01}, e_{10}\}$  модели  $\mathbf{M}$  определяет аффинный базис, который задает на  $\mathbf{M}$  систему координат (рис. 11.16 (a)). После выбора системы координат любую точку  $x \in \mathbf{M}$  можно представить с помощью аффинных координат  $(\xi, \eta)$ , где

$$x = \xi(e_{10} - e_{00}) + \eta(e_{01} - e_{00}) + e_{00}$$

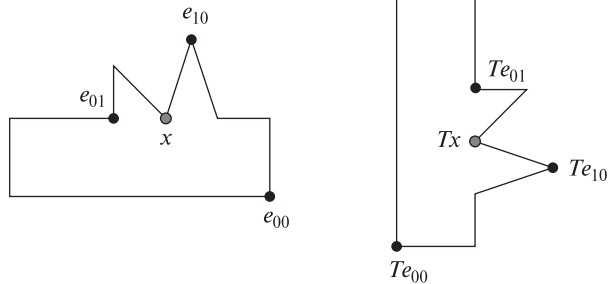
Применяя аффинное преобразование  $T$  к точке  $x$ , получим

$$Tx = \xi(Te_{10} - Te_{00}) + \eta(Te_{01} - Te_{00}) + Te_{00}$$

Следовательно,  $Tx$  имеет те же аффинные координаты  $(\xi, \eta)$  относительно  $(Te_{00}, Te_{01}, Te_{10})$ , что и точка  $x$  относительно  $(e_{00}, e_{01}, e_{10})$ . Этот факт иллюстрируется на рис. 11.16 (b).

*Предварительная обработка.* На этапе предварительной обработки формируется хэш-таблица, содержащая все модели базы данных. Хэш-таблица строится таким образом, что пара аффинных координат  $(\xi, \eta)$  служит индексом элемента хэш-таблицы, в котором хранится список пар модель-базис  $(\mathbf{M}, \mathbf{E})$ . Эти пары представляют такие модели  $\mathbf{M}$ , в которых некоторая модельная точка  $x$  имеет аффинные координаты  $(\xi, \eta)$  относительно базиса  $\mathbf{E}$ . Основные операции предварительной обработки представлены в виде алгоритма 11.3.

*Процедура распознавания объекта.* Хэш-таблица, построенная на этапе предварительной обработки, впоследствии используется в процессе распознавания объектов. В процедуре распознавания также применяется аккумуляторный массив  $\mathbf{A}$ , проиндексированный парами модель-базис. Элементы массива для каждой пары  $(\mathbf{M}, \mathbf{E})$  инициализируются нулевыми значениями и используются для голосования в поддержку гипотезы о том, что существует преобразование  $\mathbf{T}$ , отображающее  $(\mathbf{M}, \mathbf{E})$  на изображение. Вычисление параметров преобразований производится только для таких пар модель-базис, для которых было получено большое количество голосов. В этом заключается одна из операций процедуры верификации, выполняемой после процедуры голосо-



**Рис. 11.16.** Аффинное преобразование точек относительно аффинного базиса.

(a) Исходный объект

(b) Преобразованный объект

### Формирование хэш-таблицы для сопоставления изображения с базой данных моделей методом геометрического хэширования.

**D** — база данных моделей.

**H** — изначально пустая хэш-таблица.

```

procedure GH_Preprocessing(D, H);
{
  for каждая модель M
  {
    Выделение множества признаков точек FM для модели M;
    for каждая неколлинеарная тройка точек E из множества FM
      for каждая другая точка x из FM
        {
          Вычисление  $(\xi, \eta)$  для точки x относительно базиса E;
          Сохранение (M, E) в элементе хэш-таблицы H
            с индексом  $(\xi, \eta)$ ;
        } ;
      } ;
  } ;
}

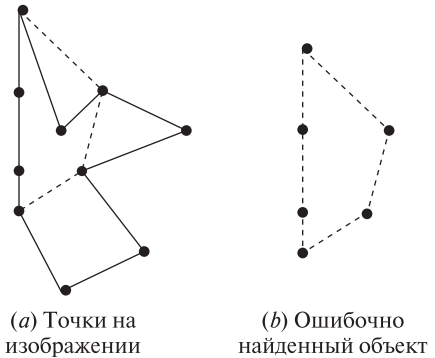
```

**Алгоритм 11.3.** Предварительная обработка в методе геометрического хэширования.

вания. Основные действия по распознаванию и оценке параметров расположения объекта представлены в алгоритме 11.3.

Предположим, что имеется  $s$  моделей, каждая из которых содержит примерно  $n$  точек. Оценка временных затрат на этапе предварительной обработки в таком случае составляет  $O(sn^4)$ . В этой оценке учитывается, что для  $s$  моделей в расчете на каждую модель обрабатывается  $O(n^3)$  троек значений и  $O(n)$  других точек. Количество выполняемых при сопоставлении операций зависит от того, насколько надежно на изображении могут быть найдены признаковые точки, сколько из них закрыты другими объектами, и сколько обнаружено ложных или лишних признаков точек. В наилучшем случае в первую выбранную тройку попадут три признаковые точки, в действительности принадлежащие одной и той же модели. Тогда эта модель получит большое количество голосов, процедура верификации пройдет успешно, и задача будет решена. Для этого наилучшего случая, в предположении, что средний размер списка в хэш-таблице равен небольшому числу и что время хэширования примерно постоянно, временные затраты на этапе сопоставления равны примерно  $O(n)$ . В худшем случае, например, когда модель объекта вообще отсутствует в базе данных, будет проверяться каждая тройка и временные затраты составят  $O(n^4)$ . На практике проверка всех базисов маловероятна, но

**Рис. 11.17.** Алгоритм геометрического хэширования может ошибочно определять присутствие данной модели на изображении. В данном примере 60% признаков точек (слева) привели к подтверждению гипотезы о наличии на изображении объекта (справа), которого в действительности на изображении нет.



также маловероятно и успешное завершение обработки после проверки только одного базиса.

На различных этапах обработки могут возникать ошибки:

1. неточное определение координат признаков точек;
2. пропуск или обнаружение лишних признаков точек;
3. перекрытие нескольких объектов;
4. неустойчивые базисы и
5. ошибочные аффинные преобразования для подмножеств точек.

В частности, алгоритм может вычислить ложное преобразование на основе подмножества точек, которые прошли процедуру верификации точек, но приводят к неверному ответу (рис. 11.17). Кластеризация по расположению и метод базовых локальных признаков также подвержены подобным ошибкам.

## 11.6 РАСПОЗНАВАНИЕ ДВУМЕРНЫХ ОБЪЕКТОВ С ИСПОЛЬЗОВАНИЕМ РЕЛЯЦИОННЫХ МОДЕЛЕЙ

Выше были описаны три метода для сопоставления точек изображения с точками модели: метод базовых локальных признаков, кластеризация по расположению и геометрическое хэширование. В этом разделе в контексте данной главы будут представлены три общих подхода к распознаванию объектов. Во всех трех подходах распознавание рассматривается как отображение структурных компонентов модели на компоненты изображения. При условии маркировки признаков изображения способом, согласованным с маркировкой признаков модели, распознавание эквивалентно отображению достаточного количества признаков некоторой модели на наблюдаемые признаки изображения. Три описываемых подхода к распознаванию объектов различаются способами определения этого отображения.

Применительно к рассмотрению методов сопоставления введем четыре важных понятия — *часть*, *метка*, *назначение* и *отношение*.



**Использование хэш-таблицы для поиска корректной модели и преобразования, отображающего характерные признаки изображения на признаки модели.**

**H** — хэш-таблица, сформированная на этапе предварительной обработки.  
**A** — аккумуляторный массив, проиндексированный парами значений (**M**, **E**).

**I** — анализируемое изображение.

**procedure** GH\_Recognition(**H**, **A**, **I**);

```
{
  Инициализация аккумуляторного массива A нулевыми значениями;
  Выделение характерных признаков на изображении I;
  for каждая тройка точек, образующих базис F
  {
    for каждая другая точка v
    {
      Вычисление  $(\xi, \eta)$  для точки v относительно базиса F;
      Получение списка L пар модель-базис из
        хэш-таблицы H по индексу  $(\xi, \eta)$ ;
      for каждая пара (M, E) из L
         $A[M, E] = A[M, E] + 1$ ;
    };
    Поиск локальных максимумов в аккумуляторном массиве A;
    for каждый максимум (M, E)
    {
      Вычисление T такого, что  $F = TE$ ;
      if достаточное количество точек модели M после преобразования
        обнаружены на изображении then return(T);
    };
  };
}
```

**Алгоритм 11.4.** Распознавание объекта методом геометрического хэширования.

- **Часть** (part) — это объект или структурный компонент сцены, например, сегментированная область, контурный сегмент, отверстие, угол или пятно.
- **Меткой** (label) называется символ, назначенный части с целью ее идентификации и/или в результате распознавания части на некотором уровне обработки.
- **Назначением** (assignment) называется отображение частей на метки. Если  $P_1$  является областью на изображении карты,  $L_1$  — символьная метка озера, а  $L_2$  — символьная метка сельскохозяйственного поля, то назначение может

включать пару  $(P_1, L_2)$  или, возможно,  $(P_1, \{L_1, L_2\})$  для указания на оставшуюся неоднозначность. Пара  $(P_1, NIL)$  означает, что часть  $P_1$  не имеет интерпретации на текущем множестве меток. Интерпретацией сцены называется множество всех пар, образующих назначение.

- **Отношение** (relation) является абстрактным математическим понятием. Отношения обнаруживаются и вычисляются среди объектов сцены и могут храниться для модельных объектов. Например,  $R_4(P_1, P_2)$  может означать, что область  $P_1$  является *смежной* для области  $P_2$ .

С учетом четырех приведенных понятий определим понятие согласованной маркировки.

**87 Определение.** Дано множество частей  $P$ , множество меток этих частей  $L$ , отношение  $R_P$  на множестве  $P$  и второе отношение  $R_L$  на множестве  $L$ . **Согласованной маркировкой** (consistent labeling)  $f$  называется такое назначение меток частям, которое удовлетворяет условию:

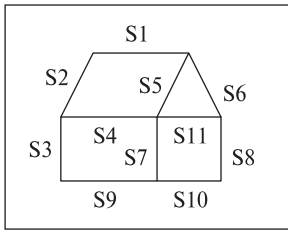
$$\text{Если } (p_i, p_{i'}) \in R_P, \quad \text{то } (f(p_i), f(p_{i'})) \in R_L$$

Например, допустим, что требуется сопоставить два изображения. Для каждого изображения имеется множество выделенных отрезков и задано отношение связности, определяющее пары соединенных отрезков. Обозначим символом  $P$  множество отрезков и  $R_P$  множество пар соединенных отрезков на первом изображении,  $R_P \subseteq P \times P$ . Аналогично, множество  $L$  будет множеством отрезков, а  $R_L$  — множеством пар соединенных отрезков на втором изображении,  $R_L \subseteq L \times L$ . На рис. 11.18 показаны примеры двух изображений и множеств  $P$ ,  $R_P$ ,  $L$  и  $R_L$ . Отметим, что отношения  $R_P$  и  $R_L$  являются симметричными. Если такому отношению принадлежит пара  $(S_i, S_j)$ , то ему также будет принадлежать пара  $(S_j, S_i)$ . В наших примерах перечислены только кортежи  $(S_i, S_j)$  при  $i < j$ , но неявно предполагается и наличие кортежей  $(S_j, S_i)$ .

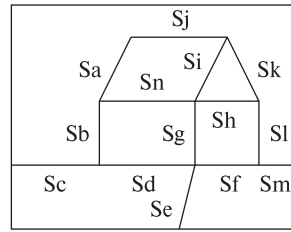
Согласованная маркировка  $f$  для данной задачи задается следующими выражениями:

$$\begin{aligned} f(S1) &= S_j & f(S7) &= S_g \\ f(S2) &= S_a & f(S8) &= S_l \\ f(S3) &= S_b & f(S9) &= S_d \\ f(S4) &= S_n & f(S10) &= S_f \\ f(S5) &= S_i & f(S11) &= S_h \\ f(S6) &= S_k & & \end{aligned} \tag{11.18}$$

В качестве другого примера рассмотрим задачу распознавания объекта, показанного на рис. 11.8. Признаки этого объекта приведены в таблицах, соответствующих рис. 11.8. При сопоставлении будем пользоваться отношением расстояния, которое определено для любых двух точек. Каждая пара отверстий находится в отношении, заданном расстоянием между ними. Расстояние инвариантно относительно поворотов и переносов, но не относительно



Изображение 1



Изображение 2

$$P = \{S1, S2, S3, S4, S5, S6, S7, S8, S9, S10, S11\}.$$

$$L = \{Sa, Sb, Sc, Sd, Se, Sf, Sg, Sh, Si, Sj, Sk, Sl, Sm\}.$$

$$R_P = \{ (S1, S2), (S1, S5), (S1, S6), (S2, S3), (S2, S4), (S3, S4), (S3, S9), (S4, S5), (S4, S7), (S4, S11), (S5, S6), (S5, S7), (S5, S11), (S6, S8), (S6, S11), (S7, S9), (S7, S10), (S7, S11), (S8, S10), (S8, S11), (S9, S10) \}.$$

$$R_L = \{ (Sa, Sb), (Sa, Sj), (Sa, Sn), (Sb, Sc), (Sb, Sd), (Sb, Sn), (Sc, Sd), (Sd, Se), (Sd, Sf), (Sd, Sg), (Se, Sf), (Se, Sg), (Sf, Sg), (Sf, Sl), (Sf, Sm), (Sg, Sh), (Sg, Si), (Sg, Sn), (Sh, Si), (Sh, Sk), (Sh, Sl), (Sh, Sn), (Si, Sj), (Si, Sk), (Si, Sn), (Sj, Sk), (Sk, Sl), (Sl, Sm) \}.$$

**Рис. 11.18.** Пример задачи определения согласованной маркировки.

масштабирования. Будем пользоваться обозначениями  $12(A, B)$  и  $12(B, C)$ , чтобы указать, что модельные точки  $A$  и  $B$  находятся на расстоянии 12 и в аналогичном отношении находятся точки  $B$  и  $C$ . Как видно из таблиц расстояний, отношение  $12(C, D)$  не выполняется. Для учета небольших искажений или ошибок обнаружения признаков можно допустить, что  $12(C, D)$  истинно, даже если расстояние между  $C$  и  $D$  в действительности равно  $12 \pm \Delta$  при некотором малом значении  $\Delta$ .

### Упражнение 11.13. Задача согласованной маркировки

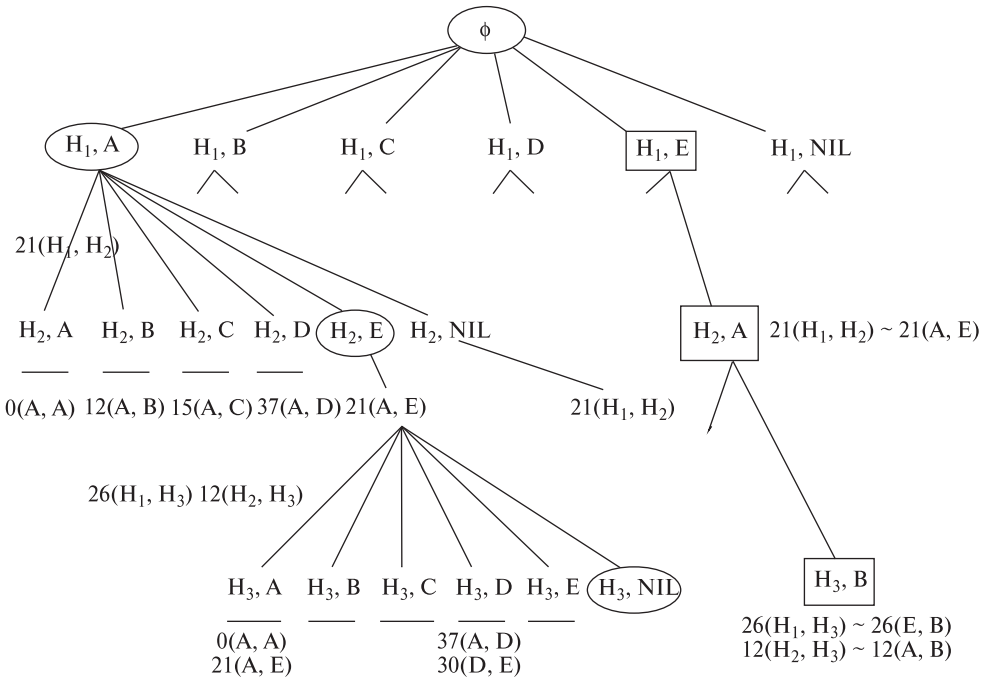
Покажите, что приведенная выше маркировка  $f$  является согласованной. Так как отношения симметричны, то должно выполняться следующее модифицированное ограничение:

$$\text{Если } (p_i, p_{i'}) \in R_P, \text{ то } (f(p_i), f(p_{i'})) \in R_L \text{ или } (f(p_{i'}), f(p_i)) \in R_L.$$

### Дерево интерпретации

**88 Определение.** **Дерево интерпретации** (interpretation tree, ИТ) — это дерево, представляющее все возможные варианты назначения меток частям. Каждый путь в дереве интерпретации завершается потому, что представляет полное согласованное назначение, или потому, что для соответствующего неполного назначения не выполняется некоторое отношение.

На рис. 11.19 показан фрагмент дерева интерпретации для данных изображения на рис. 11.8. У этого дерева три уровня. Каждый уровень соответствует



**Рис. 11.19.** Фрагмент дерева интерпретации для согласованной маркировки частей объекта, показанного на рис. 11.8 (справа).

вариантам назначения меток одному из трех имеющих на изображении отверстий  $H_1, H_2, H_3$ . На первом уровне несогласованностей нет, так как не задано ограничений по расстоянию. Однако пути для большинства возможных меток завершаются после одной проверки расстояния на втором уровне дерева. Например, неполное назначение  $\{(H_1, A), (H_2, A)\}$  является несогласованным, так как отношение  $21(H_1, H_2)$  нарушается отношением  $0(A, A)$ . Для экономии места многие пути на дереве на рисунке не показаны. Путь для меток, обозначенных прямоугольниками, соответствует полному согласованному назначению. Путь для меток, обозначенных эллипсами, также является согласованным, но содержит одну пустую метку NIL и поэтому имеет меньше ограничений для проверки. Это назначение содержит две первые пары полного назначения (обозначенного прямоугольниками) с переставленными метками и одной проверкой расстояния. Многие пути на дереве интерпретации успешно завершаются в силу симметричности отношений. Хотя дерево интерпретации потенциально может содержать экспоненциальное количество путей, большинство путей будут завершаться на третьем уровне из-за ограничивающих отношений. Применение пустой метки NIL позволяет обозначить посторонние признаки или признаки другого объекта сцены.

Дерево интерпретации можно построить с помощью рекурсивной обработки с возвратом, в процессе которой пути формируются методом просмотра дерева «в глубину». На любом шаге процедуры параметр  $f$  содержит согласованное неполное назначение. При инициализации  $f$  присваивается значение NIL. Всякий раз, когда новая маркировка части согласуется с неполным назначением, алгоритм просматривает дерево в глубину, пытаясь назначить непомеченной части другую метку. При обнаружении несогласованности алгоритм возвращается назад и производит альтернативный выбор. В приведенном алгоритме возвращается первый полный путь. Этот путь может содержать метки NIL, если эта метка была явно включена в  $L$ . В качестве усовершенствования алгоритма можно было бы предусмотреть возврат полного пути, содержащего большинство непустых пар, или, возможно, возврат всех полных путей.

Рекурсивный алгоритм поиска на дереве интерпретации определен в общем виде для работы с произвольными  $N$ -арными отношениями  $R_P$  и  $R_L$ , а не только с бинарными отношениями. Отношения  $R_P$  и  $R_L$  могут быть одиночными отношениями, например, как отношение соединения в нашем первом примере, или в качестве  $R_P$  и  $R_L$  могут рассматриваться объединения нескольких различных отношений (например, таких, как соединение, параллельность или расстояние).

**Дискретная релаксация.** В этом методе вместо всех имеющихся ограничений учитываются только локальные. Например, это могут быть все ограничения, соответствующие сопоставленным признакам на одном пути дереве интерпретации. После  $N$  итераций локальные ограничения из окрестности одной части могут быть распространены на другую часть того же объекта, на путь, отстоящий от текущего на  $N$  ребер дерева. Хотя ограничения, используемые на одной итерации, являются более слабыми, чем ограничения при полном поиске на дереве интерпретации, но они могут обрабатываться параллельно. Поэтому процедура поиска оказывается более простой и быстрой.

При инициализации части ей может быть назначена любая метка допустимого типа. Предположим, что части было присвоено множество всех допустимых меток. В методе дискретной релаксации проверяются отношения между заданной частью и всеми остальными. При этом множество допустимых для данной части меток уменьшается. Например, в задаче распознавания символов, если известно, что следующий символ не может быть буквой 'U', можно заключить, что текущий символ не может быть буквой 'Q'. В задаче обработки спутниковых изображений, если известно, что некоторая область изображения не является водной поверхностью, объект внутри нее не может быть кораблем. Метод дискретной релаксации стал популярным благодаря Дэвиду Вальсу (David Waltz), который применил его для ограничения меток, назначаемых отрезкам на чертежах (Фильтрация Вальса обсуждается в книге Уинстона [Winston, 1977].) Д. Вальс использовал последовательный алгоритм. В данном разделе представлен алгоритм, допускающий параллельную обработку.

**Применение дерева интерпретации для поиска отображения характерных признаков модели на характерные признаки изображения. Отображение должно удовлетворять определенным в модели отношениям.**

**P** — множество обнаруженных характерных признаков изображения.

**L** — множество хранимых характерных признаков модели.

**R<sub>P</sub>** — отношения между признаками изображения.

**R<sub>L</sub>** — отношения между признаками модели.

**f** — согласованная маркировка, возвращаемая алгоритмом.

Инициализируется значением NIL.

**procedure** Interpretation\_Tree\_Search(**P**, **L**, **R<sub>P</sub>**, **R<sub>L</sub>**, **f**);

```
{
  p := first(P);
  for каждый l из L
  {
    f' = f ∪ {(p, l)}; /* Добавление к интерпретации пары
      часть-метка */
    ОК = true;
    for каждый N-элементный кортеж (p1, ..., pN) из RP,
      содержащий компонент p и у которого все остальные элементы
      принадлежат области определения (f)
      /* Проверка выполнимости отношений */
      if (f'(p1), ..., f'(pN)) не принадлежат RL then
      {
        ОК := false;
        break;
      }
    if ОК then
    {
      P' = rest(P);
      if isempty(P') then output(f');
      else Interpretation_Tree_Search(P', L, RP, RL, f');
    }
  }
}
```

**Алгоритм 11.5.** Поиск на дереве интерпретации.

Каждой части **P<sub>i</sub>**, согласно ее типу, сначала назначается все множество допустимых меток **L<sub>j</sub>**. Затем проверяется выполнимость всех отношений и обнаруживаются несогласованные метки. Они удаляются из множества допустимых меток. Множества меток различных частей могут обрабатываться параллельно. Если на текущем шаге обработки из каких-либо множеств были

**Таблица 11.4.** Промежуточное состояние в процессе первого прохода маркировки методом дискретной релаксации

	A	B	C	D	E
$H_1$	no $N \ni$ $d(A, N) = 26$	no $N \ni$ $d(B, N) = 21$	no $N \ni$ $d(C, N) = 26$	no $N \ni$ $d(D, N) = 26$	$21(H_1, H_2)$ $A \in L(H_2)$ $26(H_1, H_3)$ $B \in L(H_3)$
$H_2$	$21(H_2, H_1)$ $E \in L(H_1)$ $12(H_2, H_3)$ $B \in L(H_3)$	no $N \ni$ $d(B, N) = 21$	$21(H_2, H_1)$ $D \in L(H_1)$ $12(H_2, H_3)$ $B \in L(H_3)$		
$H_3$	no $N \ni$ $d(A, N) = 26$	$12(H_3, H_2)$ $A \in L(H_2)$ $26(H_3, H_1)$ $E \in L(H_1)$			

удалены некоторые метки, то выполняется еще один шаг обработки. Если же ни одно из множеств не было изменено, то фильтрация прекращается. В результате может не остаться ни одной допустимой интерпретации, или может остаться несколько.

Для пояснения метода рассмотрим пример. С целью упрощения, будем считать, что не было обнаружено лишних признаков, в действительности не относящихся к частям модели. Как и ранее, будем полагать, что некоторые признаки могли быть пропущены.

Сопоставим данные из таблиц 11.1 и 11.2. Фильтрация начинается с того, что все 5 возможных меток назначаются каждому из трех отверстий  $H_1, H_2, H_3$ . Чтобы сделать пример более интересным и приближенным к прикладным задачам, *при сопоставлении по расстоянию будем допускать отклонения  $\pm 1$* . В таблице 11.4 показаны три множества меток в некоторый момент в процессе выполнения первого прохода обработки. В каждой ячейке таблицы указана причина, по которой метка должна быть удалена или может быть оставлена. Метка  $A$  удаляется из множества меток для  $H_1$ , так как отношение  $26(H_1, H_3)$  не выполняется ни для одной метки из множества для  $H_3$ . Метка  $A$  сохраняется для  $H_2$ , так как для выполнимости отношения  $21(H_2, H_1)$  существует метка  $E \in L(H_1)$ , а для выполнимости отношения  $12(H_2, H_3)$  — метка  $B \in L(H_3)$ . Для  $H_1$  сохраняется метка  $C$ , так как  $d(H_2, H_1) = 21 \approx 22 = d(C, D)$ .

Как показано в таблице 11.5, после завершения первого прохода обработки остаются только две допустимых метки для  $H_2$ , одна метка  $E$  для  $H_1$ , и одна метка  $B$  для  $H_3$ . После первого прохода сокращенное множество меток оказывается пригодным для параллельной обработки на втором проходе, на

**Таблица 11.5.** Состояние после завершения первого прохода маркировки методом дискретной релаксации

	A	B	C	D	E
$H_1$	нет	нет	нет	нет	допустимо
$H_2$	допустимо	нет	допустимо	нет	нет
$H_3$	нет	допустимо	нет	нет	нет

котором каждое множество меток подвергается дальнейшей фильтрации в асинхронном параллельном порядке.

### Упражнение 11.14

Приведите подробное обоснование состояний всех меток, удаленных или оставленных после первого прохода обработки (согласно таблице 11.5).

На втором проходе метка  $C$  удаляется из  $L(H_2)$ , так как отношение  $21(H_1, H_2)$  более не удовлетворяется меткой  $D$  в качестве метки для  $H_1$ . После третьего прохода последующие проходы обработки не меняют ни одно множество меток, поэтому фильтрация завершается. В данном случае все множества меток являются одноэлементными и представляют одно назначение и интерпретацию. Высокоуровневая схема алгоритма обработки приведена в виде алгоритма 11.6. Хотя маркировка методом дискретной релаксации является простой и потенциально быстрой процедурой по сравнению с поиском, на дереве интерпретации иногда допускаются существенные неоднозначности в интерпретации. Это объясняется тем, что ограничения применяются только попарно. Релаксационная маркировка может применяться в качестве предварительной обработки перед поиском на дереве интерпретации. Такая предобработка может существенно уменьшить коэффициент ветвления дерева поиска.

**Непрерывная релаксация\*.** В точных процедурах согласованной маркировки (например, при поиске на дереве интерпретации или в методе дискретной релаксации) на любом этапе обработки метка  $l$  для части  $p$  считается либо допустимой, либо нет. При обнаружении несовместимости пары часть-метка  $(p, l)$  с некоторой ранее запомненной парой метка  $l$  помечается как недопустимая для части  $p$ . Характерное для рассмотренных выше алгоритмов применение двухвариантной классификации меток на допустимые и недопустимые позволяет называть эти алгоритмы *дискретными*. В отличие от них, каждой паре часть-метка  $(p, l)$  можно сопоставить вещественное число, представляющее вероятность назначения метки  $l$  для части  $p$ . Алгоритмы, в которых используется этот прием, называются *непрерывными*. В данном разделе будет рассмотрен алгоритм маркировки методом *непрерывной релаксации* для симметричных бинарных отношений.



**Таблица 11.6.** Состояние после завершения второго прохода маркировки методом дискретной релаксации

	A	B	C	D	E
$H_1$	нет	нет	нет	нет	допустимо
$H_2$	допустимо	нет	нет	нет	нет
$H_3$	нет	допустимо	нет	нет	нет

**Таблица 11.7.** Состояние после завершения третьего прохода маркировки методом дискретной релаксации

	A	B	C	D	E
$H_1$	нет	нет	нет	нет	допустимо
$H_2$	допустимо	нет	нет	нет	нет
$H_3$	нет	допустимо	нет	нет	нет

В задаче непрерывной релаксационной маркировки имеется 6-элементный кортеж  $CLRP = (P, L, R_P, R_L, PR, C)$ . Как и ранее,  $P$  представляет собой множество частей,  $L$  — множество меток для этих частей,  $R_P$  — отношения между частями, а  $R_L$  — отношения между метками. Множество  $L$  обычно задается в виде объединения всех частей  $i$  из  $L_i$ , множества допустимых меток для части  $i$ . Предположим, что  $|P| = n$ . Тогда  $PR$  будет множеством из  $n$  функций  $PR = \{pr_1, \dots, pr_n\}$  где  $pr_i(l)$  описывает *априорную вероятность* того, что метка  $l$  является корректной меткой части  $i$ . Множество  $C$  содержит  $n^2$  коэффициентов совместимости  $C = \{c_{ij}, i = 1, \dots, n; j = 1, \dots, n$ . Коэффициент  $c_{ij}$  характеризует влияние части  $j$  на метки части  $i$ . В таком случае, если представить ограничивающие отношения  $R_P$  в виде графа, то коэффициент  $c_{ij}$  можно рассматривать как вес ребра между узлами для частей  $i$  и  $j$ .

Вместо непосредственного использования  $R_P$  и  $R_L$ , скомбинируем их для образования множества  $R$  из  $n^2$  функций  $R = \{r_{ij}, i = 1, \dots, n; j = 1, \dots, n$ . Функция  $r_{ij}(l, l')$  характеризует совместимость метки  $l$  для части  $i$  с меткой  $l'$  для части  $j$ . В дискретном случае функция  $r_{ij}(l, l')$  могла бы принимать значение 1 для обозначения допустимости  $((i, l), (j, l'))$ . Если бы функция совместимости была равна 0, то соответствующая комбинация считалась бы недопустимой. В непрерывном случае,  $r_{ij}(l, l')$  может принимать любое значение в диапазоне от 0 до 1, указывая на степень совместимости отношений между частями  $i$  и  $j$  с отношением между метками  $l$  и  $l'$ . Значения функций совместимости могут вычисляться с использованием свойств элементов  $R_P$  и  $R_L$ . Эти множества могут содержать как простые, так и атрибутивные бинарные отношения. В атрибутивных отношениях паре частей (или паре меток) сопоставляется атрибут, представляющий вероятность выполнимости требуемого отношения на этой паре. Решение задачи непрерывной релакса-

**Удаление несовместимых меток из множеств допустимых меток характерных признаков изображения.**

$P_i, i = 1, D$  — множество обнаруженных характерных признаков изображения.

$S(P_i), i = 1, D$  — множество меток, изначально считающихся допустимыми.

$R$  — отношение, по которому определяется совместимость меток.

```

procedure Relaxation_Labeling( $P, S, R$ );
{
  repeat
    for каждая пара  $(P_i, S(P_i))$ 
    {
      for каждая метка  $L_k \in S(P_i)$ 
        for каждое отношение  $R(P_i, P_j)$  между частями
          изображения
          if  $\exists L_m \in S(P_j)$  с отношением  $R(L_k, L_m)$  в модели
            then сохранить  $L_k$  в  $S(P_i)$ 
            else удалить  $L_k$  из  $S(P_i)$ 
    }
  until не было изменено ни одно множество  $S(P_i)$ 
  return( $S$ );
}

```

**Алгоритм 11.6.** Маркировка методом дискретной релаксации.

ционной маркировки, как и задачи согласованной маркировки, представляет собой такое отображение  $f : P \rightarrow L$ , которое назначает метку каждой части. В отличие от дискретного случая, отсутствует внешнее определение, в котором содержатся условия, накладываемые на отображение  $f$ . Вместо этого отображение  $f$  неявно определяется в процедуре, которая генерирует искомое отображение. Эта процедура называется процедурой *непрерывной релаксации* (continuous relaxation).

На итерациях алгоритмов дискретной релаксации производится удаление возможных меток из множества меток  $L_i$  для части  $i$ , а на итерациях алгоритма непрерывной релаксации выполняется обновление значений вероятности для каждой пары часть-метка. Начальные значения вероятностей задаются множеством  $PR$ , которое содержит функции, определяющие априорные вероятности. Инициализация значений вероятности производится перед нулевой итерацией алгоритма непрерывной релаксации. Следовательно, начальное значение вероятности для каждой части  $i$  и метки  $l$  можно определить следующим образом:

$$pr_i^0(l) = pr_i(l) \quad (11.19)$$

На каждой  $k$ -й итерации новое множество вероятностей  $\{pr_i^k(l)\}$  вычисляется на основе предыдущего множества и информации о совместимости. Для определения  $pr_i^k(l)$  сначала рассмотрим ее часть,  $q_i^k(l)$ , заданную выражением

$$q_i^k(l) = \sum_{\{j|(i,j) \in R_P\}} c_{ij} \left[ \sum_{l' \in L_j} r_{ij}(l, l') pr_j^k(l') \right] \quad (11.20)$$

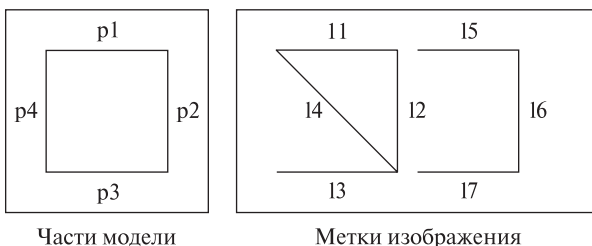
Функция  $q_i^k(l)$  представляет влияние на метку части  $i$  текущих значений вероятности, сопоставленных с метками других частей, которые ограничены частью  $i$ . С использованием этой функции выражение для обновления значений  $pr_i^k$  можно записать в виде

$$pr_i^{k+1}(l) = \frac{pr_i^k(l)(1 + q_i^k(l))}{\sum_{l' \in L_i} pr_i^k(l')(1 + q_i^k(l'))} \quad (11.21)$$

В числителе данного выражения к текущему значению вероятности  $pr_i^k(l)$  добавляется произведение  $pr_i^k(l)q_i^k(l)$  текущей вероятности и коэффициента влияния других частей, связанных отношениями с частью  $i$ . Коэффициент влияния (11.19) зависит от текущих вероятностей собственных возможных меток этих частей. Знаменатель (11.20) предназначен для нормировки выражения посредством вычисления суммы слагаемых для всех возможных меток части  $i$ .

### Упражнение 11.15. Непрерывная релаксация

На рис. 11.20 приведены модель и изображение, содержащие отрезки. Два отрезка называются находящимися в отношении *closadj*, если они имеют общие или близкорасположенные вершины. (а) Сформируйте на множестве частей модели атрибутивное отношение  $R_P$ , определяемое выражением  $R_P = \{(p_i, p_j, d) \mid p_i \text{ closadj } p_j\}$ . На множестве меток изображения определите атрибутивное отношение  $R_L = \{(l_i, l_j) \mid l_i \text{ closadj } l_j\}$ . (б) Выберите коэффициенты совместимости равными  $c_{ij} = 1$  если  $(p_i, p_j) \in R_P$  или в противном случае  $c_{ij} = 0$ . С помощью  $R_P$  и  $R_L$  каким-либо образом определите  $R$ . Пусть  $pr_i(l_j)$  принимает значение 1, если ориентация  $p_i$  совпадает с ориентацией  $l_j$ . Если они перпендикулярны, то  $pr_i(l_j)$  равно 0. Если один отрезок направлен по диагонали, а другой по горизонтали или вертикали, то  $pr_i(l_j)$  принимает значение 0.5. Определите  $p_r$  для частей модели и меток изображения.



**Рис. 11.20.** Модель и изображение для обработки методом непрерывной релаксации.

(в) Выполните несколько итераций непрерывной релаксации для обнаружения вероятной маркировки, назначающей части модели меткам изображения.

**Сопоставление по реляционному расстоянию.** Во многих прикладных задачах не удается получить полностью согласованную маркировку. Ошибки выделения характерных признаков, шум и взаимные перекрытия объектов на изображении могут приводить к обнаружению лишних или к пропуску каких-то частей, и в результате требуемые отношения не удовлетворяются. В таких случаях может применяться метод непрерывной релаксации, но нахождение наилучшего решения не гарантируется. В задачах, в которых важно получить оптимальное решение, можно выполнить поиск наилучшего отображения  $f$  из  $P$  на  $L$  с использованием критерия выполнимости большинства отношений или по критерию минимизации количества пустых меток NIL. В 1981 г. Харалик и Шапиро предложили понятие *реляционного расстояния*. Это понятие позволяет определить наилучшее отображение в общем случае произвольного количества отношений, возможно, различной размерности. Предварительно введем понятие *реляционного описания* изображения или объекта.

**89 Определение. Реляционное описание** (relational description)  $D_P$  представляет собой последовательность отношений  $D_X = \{R_1, \dots, R_I\}$ , где для каждого  $i = 1, \dots, I$  существует положительное целое число  $n_i$  такое, что  $R_i \subseteq P^{n_i}$  для некоторого множества  $P$ . Множество  $P$  содержит части описываемой сущности, а отношения  $R_i$  указывают на наличие различных взаимосвязей между частями.

Реляционное описание является структурой данных, которую можно использовать для представления моделей двумерных фигур, моделей трехмерных объектов, областей изображения и т. п.

Допустим, имеются два реляционных описания  $D_A = \{R_1, \dots, R_I\}$  и  $D_B = \{S_1, \dots, S_I\}$  со множествами частей  $A$  и  $B$ , соответственно. Будем полагать, что  $|A| = |B|$ . Если это условие не выполняется, то для его выполнения добавим к меньшему множеству достаточное количество фиктивных частей. Это условие необходимо для использования реляционного расстояния в качестве метрики.

Обозначим символом  $f$  взаимно однозначное отображение множества  $A$  на  $B$ . Для любого  $R \subseteq A^N$ , где  $N$  положительное целое число, композиция  $R \circ f$  из отношения  $R$  и функции  $f$  определяется выражением

$$R \circ f = \{(b_1, \dots, b_N) \in B^N \mid \text{где существует } (a_1, \dots, a_N) \in R \\ \text{с } f(a_n) = b_n, n = 1, \dots, N\} \quad (11.22)$$

Оператор композиции обозначает операцию поэлементного отображения  $N$ -мерных кортежей из  $R$  на  $N$ -мерные кортежи  $B^N$ .

Функция  $f$  отображает части множества  $A$  на части множества  $B$ . Структурной ошибкой (structural error) функции  $f$  для  $i$ -й пары соответствующих отношений ( $R_i$  и  $S_i$ ) из описаний  $D_A$  и  $D_B$  будем называть величину

$$E_S^i(f) = |R_i \circ f - S_i| + |S_i \circ f^{-1} - R_i| \quad (11.23)$$

Функция структурной ошибки представляет, сколько кортежей из  $R_i$  не отображаются с помощью функции  $f$  на кортежи из  $S_i$  и сколько кортежей из  $S_i$  не отображаются обратной функцией  $f^{-1}$  на кортежи из  $R_i$ . Структурная ошибка определяется только для одной пары соответствующих отношений.

Общая ошибка (total error) преобразования  $f$  по отношению к  $D_A$  и  $D_B$  выражается как сумма структурных ошибок каждой пары соответствующих отношений.

$$E(f) = \sum_{i=1}^I E_S^i(f) \quad (11.24)$$

Общая ошибка (11.23) позволяет количественно охарактеризовать различие между двумя реляционными описаниями  $D_A$  и  $D_B$  по отношению к отображающей функции  $f$ .

Реляционным расстоянием (relational distance)  $GD(D_A, D_B)$  между двумя реляционными описаниями  $D_A$  и  $D_B$  называется величина

$$GD(D_A, D_B) = \min_{\substack{f: A \rightarrow B \\ \text{onto}}} E(f) \quad (11.25)$$

То есть реляционное расстояние равно минимальной полной ошибке, соответствующей некоторому взаимно однозначному отображению  $f$  множества  $A$  на  $B$ . Будем называть отображающую функцию  $f$ , которая минимизирует полную ошибку, наилучшим отображением из  $D_A$  в  $D_B$ . При наличии нескольких наилучших отображений для выбора предпочтительного отображения может использоваться какая-либо дополнительная информация, не предусмотренная в абстрактном реляционном подходе. Несколько наилучших отображений могут получаться при использовании реляционных описаний, обладающих некоторыми симметричными свойствами.

Для иллюстрации понятия реляционного расстояния рассмотрим несколько примеров. На рис. 11.21 показаны два ориентированных графа, содержащих по четыре узла. Наилучшим отображением из  $A = \{1, 2, 3, 4\}$  в  $B = \{a, b, c, d\}$

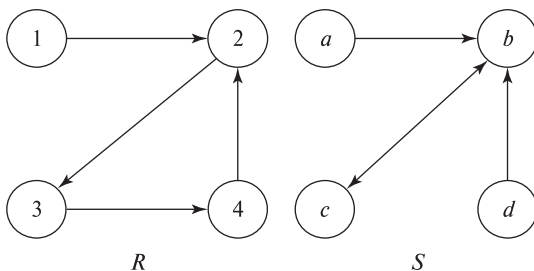


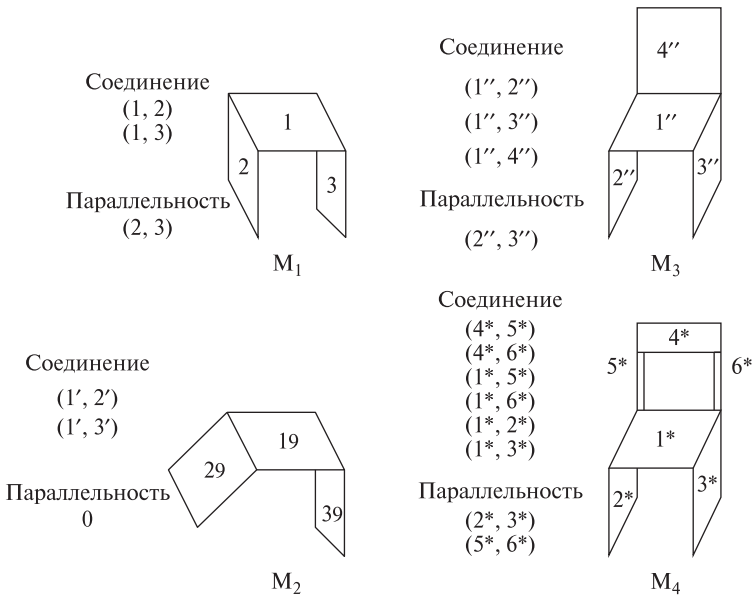
Рис. 11.21. Два ориентированных графа с реляционным расстоянием 3.

будет функция  $\{f(1) = a, f(2) = b, f(3) = c, f(4) = d\}$ . Для этого отображения имеем

$$\begin{aligned}
 |R \circ f - S| &= |\{(1, 2)(2, 3)(3, 4)(4, 2)\} \circ f - \{(a, b)(b, c)(c, b)(d, b)\}| \\
 &= |\{(a, b)(b, c)(c, d)(d, b)\} - \{(a, b)(b, c)(c, b)(d, b)\}| \\
 &= |\{(c, d)\}| \\
 &= 1 \\
 |S \circ f^{-1} - R| &= |\{(a, b)(b, c)(c, b)(d, b)\} \circ f^{-1} - \{(1, 2)(2, 3)(3, 4)(4, 2)\}| \\
 &= |\{(1, 2)(2, 3)(3, 2)(4, 2)\} - \{(1, 2)(2, 3)(3, 4)(4, 2)\}| \\
 &= |\{(3, 2)\}| \\
 &= 1 \\
 E(f) &= |R \circ f - S| + |S \circ f^{-1} - R| \\
 &= 1 + 1 \\
 &= 2
 \end{aligned}$$

Так как  $f$  является наилучшей отображающей функцией, то реляционное расстояние также равно 2.

На рис. 11.22 представлено множество моделей объектов  $M_1, M_2, M_3$  и  $M_4$ , примитивами которых являются области изображения. На рисунке показаны два отношения: отношение соединения и отношение параллельности. Оба отно-



**Рис. 11.22.** Четыре модели объектов. Реляционное расстояние от модели  $M_1$  до  $M_2$  и от  $M_1$  до  $M_3$  равно 1. Реляционное расстояние от модели  $M_3$  до  $M_4$  равно 6.

нения являются бинарными и заданы на множестве примитивов. Рассмотрим две первые модели,  $M_1$  и  $M_2$ . Наилучшее отображение  $f$  отображает примитивы 1 на  $1'$ , 2 на  $2'$ , и 3 на  $3'$ . Отношения соединения при этом отображении являются изоморфными. Отношение параллельности  $(2, 3)$  из модели  $M_1$  не выполняется для примитивов  $2'$  и  $3'$  в модели  $M_2$ . Таким образом, реляционное расстояние между  $M_1$  и  $M_2$  равно 1.

Теперь рассмотрим модели  $M_1$  и  $M_3$ . Наилучшее отображение отображает 1 в  $1''$ , 2 в  $2''$ , 3 в  $3''$ , и фиктивный примитив в  $4''$ . При этом отображении изоморфными являются отношения параллельности. В модели  $M_3$  соединений на одно больше, чем в  $M_2$ . Реляционное расстояние снова будет равно 1.

В заключение рассмотрим модели  $M_3$  и  $M_4$ . Наилучшее отображение отображает  $1''$  на  $1^*$ ,  $2''$  на  $2^*$ ,  $3''$  на  $3^*$ ,  $4''$  на  $4^*$ ,  $5_d$  на  $5^*$ , и  $6_d$  на  $6^*$ . ( $5_d$  и  $6_d$  являются фиктивными примитивами.) Для этого отображения имеем

$$\begin{aligned}
 |R_1 \circ f - S_1| &= |\{(1'', 2'')(1'', 3'')(1'', 4'')\} \circ f \\
 &\quad - \{(4^*, 5^*)(4^*, 6^*)(1^*, 5^*)(1^*, 6^*)(1^*, 2^*)(1^*, 3^*)\}| \\
 &= |\{(1^*, 2^*)(1^*, 3^*)(1^*, 4^*)\} \\
 &\quad - \{(4^*, 5^*)(4^*, 6^*)(1^*, 5^*)(1^*, 6^*)(1^*, 2^*)(1^*, 3^*)\}| \\
 &= |\{(1^*, 4^*)\}| \\
 &= 1 \\
 |S_1 \circ f^{-1} - R_1| &= |\{(4^*, 5^*)(4^*, 6^*)(1^*, 5^*)(1^*, 6^*)(1^*, 2^*)(1^*, 3^*)\} \circ f^{-1} \\
 &\quad - \{(1'', 2'')(1'', 3'')(1'', 4'')\}| \\
 &= |\{(4'', 5_d)(4'', 6_d)(1'', 5_d)(1'', 6_d)(1'', 2'')(1'', 3'')\} \\
 &\quad - \{(1'', 2'')(1'', 3'')(1'', 4'')\}| \\
 &= |\{(4'', 5_d)(4'', 6_d)(1'', 5_d)(1'', 6_d)\}| \\
 &= 4 \\
 |R_2 \circ f - S_2| &= |\{(2'', 3'')\} \circ f - \{(2^*, 3^*)(5^*, 6^*)\}| \\
 &= |\{(2^*, 3^*)\} - \{(2^*, 3^*)(5^*, 6^*)\}| \\
 &= |\emptyset| \\
 &= 0 \\
 |S_2 \circ f^{-1} - R_2| &= |\{(2^*, 3^*)(5^*, 6^*)\} \circ f^{-1} - \{(2'', 3'')\}| \\
 &= |\{(2'', 3'')(5_d, 6_d)\} - \{(2'', 3'')\}| \\
 &= |\{(5_d, 6_d)\}| \\
 &= 1 \\
 E_S^1(f) &= 1 + 4 = 5 \\
 E_S^2(f) &= 0 + 1 = 1 \\
 E(f) &= 6
 \end{aligned}$$

---

**Упражнение 11.16.** Дерево поиска с использованием реляционного расстояния

Измените алгоритм поиска на дереве интерпретации для нахождения реляционного расстояния между двумя структурными описаниями. В процессе вычисления реляционного расстояния предусмотрите определение наилучшего отображения.

---

**Упражнение 11.17.** Одностороннее реляционное расстояние

В определении реляционного расстояния в уравнении 11.24 используется двухсторонняя ошибка отображения, что может быть полезно при сравнении двух независимых объектов. При сопоставлении модели и изображения часто требуется использовать одностороннюю ошибку отображения, проверяя, сколько отношений модели имеется на изображении, но не наоборот. Предложите определение одностороннего реляционного расстояния, которое можно использовать при сопоставлении модели и изображения.

---

**Упражнение 11.18.** Учет пустых меток при вычислении реляционного расстояния

Определение реляционного расстояния в уравнении 11.24 не рассчитано на явное использование пустых меток NIL. Если часть  $j$  имеет пустую метку NIL, то любое отношение  $(i, j)$  будет приводить к ошибке, так как  $(f(i), NIL)$  не существует. Предложите модифицированное определение реляционного расстояния, которое однократно учитывает пустые метки в качестве ошибок и не учитывает их повторно для отсутствующих отношений, вызванных наличием пустых меток NIL.

---

**Упражнение 11.19.** Атрибутивное реляционное расстояние

Определение в виде уравнения (11.24) также не рассчитано на использование атрибутивных отношений, в которых для каждого кортежа, кроме последовательности частей, хранится один или более атрибутов отношения. Например, отношение соединения для отрезков может иметь атрибут, в котором хранится значение угла между соединенными отрезками. Формально, атрибутивное  $n$ -арное отношение  $R$  на множестве частей  $P$  и множество атрибутов  $A$  является множеством  $R \subseteq P_n \times A_m$  для некоторого неотрицательного целого числа  $m$ , которое задает количество атрибутов в отношении. Определите модифицированное реляционное расстояние для атрибутивных отношений.

**Реляционная индексация.** Поиск на дереве интерпретации иногда даже после предварительной релаксационной фильтрации выполняется слишком медленно (например, при сравнении изображения с большой базой данных моделей). Для структурных описаний, заданных в терминах маркированных



отношений, реляционное расстояние можно приближенно вычислить с помощью простой схемы с голосованием.

В качестве наглядного пояснения рассмотрим изображение, которые содержит две концентрические окружности и два прямых угла, соединенных ребром. Требуется быстро найти все модели, содержащие такие структуры, и затем выполнить более тщательное сопоставление с этими моделями. Для быстрого доступа к моделям по заданной неполной графовой структуре можно предусмотреть индексацию базы данных моделей. Тогда по индексу, построенному для двух концентрических окружностей, будут получены все модели, содержащие такие взаимосвязанные признаки. Каждой из этих моделей отдадим один голос. Затем найдем все модели, содержащие соединенные ребром прямые углы. Все модели, которые были получены повторно, теперь будут иметь два голоса. Операции индексированного доступа могут выполняться весьма быстро за счет того, что индекс формируется на этапе предварительной обработки до выполнения распознавания. При построении индекса производится выделение значимых бинарных отношений из каждой модели и сохранение информации о них в индексной таблице соответствий.

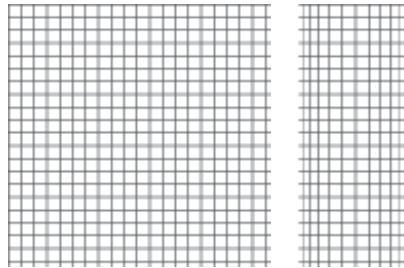
Допустим, имеется база данных  $DB = \{M_1, M_2, \dots, M_T\}$ , содержащая  $T$  моделей объектов. Каждая модель объекта  $M_t$  состоит из множества атрибутивных частей  $P_t$  и множества маркированных отношений  $R_t$ . Для простоты будем полагать, что каждая часть имеет одну метку, а не вектор атрибутов, и что отношение является бинарным отношением, также с единственной меткой, назначенной каждому кортежу. В данном случае модель представляется в виде множества *2-графов*, содержащих по два узла и два ориентированных ребра. Каждый узел представляет часть, а каждое ребро — ориентированное бинарное отношение. В узле хранится значение, являющееся *меткой* части (а не уникальным идентификатором части). Аналогично, для ребра хранится значение, являющееся меткой отношения. Например, один узел мог бы представлять эллипс, а другой — пару параллельных отрезков. Ребро между узлом параллельных отрезков и узлом эллипса могло бы представлять отношение «содержать внутри», а ребро в обратном направлении — отношение «находиться внутри».

Реляционная индексация требует выполнения предварительной обработки, в процессе которой формируется большая хэш-таблица. Хэш-таблица индексируется с использованием представления 2-графов в виде строк. После построения хэш-таблицы по любому 2-графу можно найти и получить список всех моделей, содержащих этот 2-граф. В нашем примере, могли бы быть найдены все модели, содержащие эллипс между двумя параллельными отрезками. В процессе распознавания объекта на изображении выделяются характерные признаки и вычисляются все 2-графы, представленные на изображении. Аккумуляторный массив, содержащий по одному элементу для каждой модели базы данных, сначала заполняется нулевыми значениями. Затем каждый 2-граф изображения используется в качестве индекса для доступа к хэш-таблице и получения списка соответствующих моделей. Для каждой из этих моделей увеличивается значение в аккумуляторном массиве. В дискретной версии

алгоритма добавляется один голос; в вероятностном алгоритме могло бы добавляться значение вероятности. После использования для голосования всех 2-графов, в качестве кандидатов для верификации выбираются модели с наибольшим количеством голосов.

## 11.7 НЕЛИНЕЙНЫЕ МЕТОДЫ ДЕФОРМАЦИИ ИЗОБРАЖЕНИЙ

Кроме линейных преобразований, большое значение имеют нелинейные функции для описания деформации изображений. Например, они могут применяться для устранения нелинейных искажений изображения, в частности, радиальной дисторсии широкоугольного объектива. Или, изображение может потребоваться подвергнуть некоторой деформации в художественных целях. На рис. 11.23 показан пример нелинейной деформации, отображающей регулярную сетку на цилиндр. Полученный результат напоминает наблюдение с некоторого расстояния цилиндра, завернутого в лист бумаги с изображением. Применение этой цилиндрической деформации к изображению двадцатидолларовой банкноты демонстрируется на рис. 11.24.



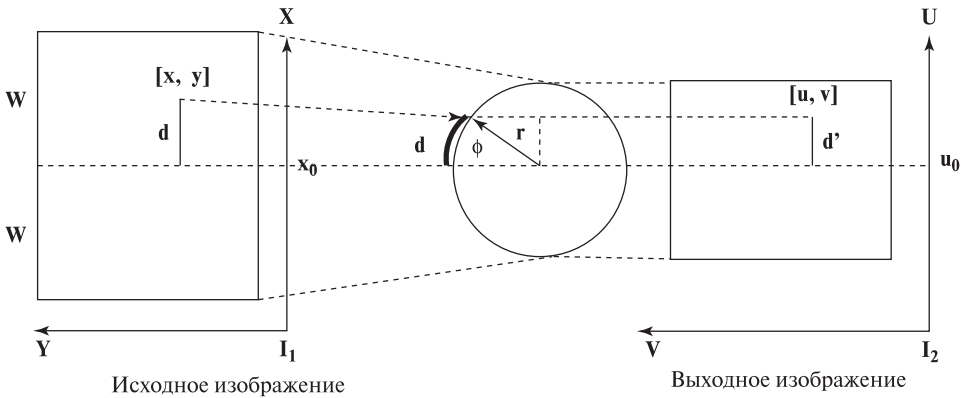
**Рис. 11.23.** (Слева) регулярная сетка линий. (Справа) сетка после цилиндрической деформации.

Для описания цилиндрической деформации выберем на изображении прямую, пересекающуюся с осью цилиндра, и затем запишем выражение, представляющее «сжатие» пикселей относительно оси цилиндра на выходном изображении. Результаты применения цилиндрической деформации с двумя различными параметрами показаны на рис. 11.24. Для получения центрального изображения был выбран цилиндр большего радиуса, чем для правого изображения.

Выполнение цилиндрической деформации поясняется на рис. 11.25. Выберем ось деформации (она задается значением  $x_0$ ) и ширину  $W$ . Значение  $W$  соответствует четверти длины окружности основания цилиндра. Любой отрезок длины  $d$  на исходном изображении «наворачивается на цилиндр» и затем проектируется на выходное изображение. Значение  $d$  соответствует длине  $x - x_0$ , где  $x_0$  является координатой оси цилиндра по оси  $X$ . Координата  $y$  точки исходного изображения при деформации сохраняется, так что имеем  $v = y$ . Рис. 11.25 позволяет получить следующие выражения. Во-первых,  $W = (\pi/2)r$  так как  $W$  соответствует одной четверти окружности основания цилиндра. Запишем пропорцию  $d/W = \phi/(\pi/2)$  и синус угла  $\sin \phi = d'/r$ . Комбинируя



**Рис. 11.24.** (Слева) изображение центральной части банкноты достоинством 20\$. (В центре) портрет президента Джексона после цилиндрической деформации с длиной окружности основания цилиндра 640 пикселей. (Справа) тот же самый портрет после деформации с длиной окружности основания цилиндра 400 пикселей.



**Рис. 11.25.** Выходное изображение в правой части было получено *наворачиванием* исходного изображения (слева) на цилиндр (в центре). Отрезок  $d$  исходного изображения преобразуется в отрезок  $d'$  на выходном изображении.

эти выражения, получим  $d = x - x_0 = (2W/\pi)\arcsin((\pi/2W)(u - u_0))$ . Конечно,  $d' = u - u_0 = u - x_0$ .

Полученная формула позволяет вычислить координаты пиксела исходного изображения  $[x, y]$ , если известны координаты пиксела выходного изображения  $[u, v]$  и параметры деформации  $x_0$  и  $W$ . Почему преобразование выполняется в таком направлении? Если брать каждый пиксел исходного изображения и преобразовывать его в пиксел выходного, то не гарантируется, что в результате будут заполнены все пикселы выходного изображения. Выходное цифровое изображение удобнее получать посредством процедуры однократного вычисления значений всех пикселов выходного изображения по данным пикселов исходного изображения. Эта процедура представлена в алгоритме 11.7. Кроме того, этот подход применим в случаях, когда выходное изображение содержит больше или меньше пикселов, чем исходное. Основной принцип генерации вы-

**Выполнение цилиндрической деформации.** ${}^1\mathbf{I}[\mathbf{x}, \mathbf{y}]$  – исходное изображение. $\mathbf{x}_0$  – параметр, определяющий ось цилиндра. $\mathbf{W}$  – ширина. ${}^2\mathbf{I}[\mathbf{u}, \mathbf{v}]$  – выходное изображение.

```

procedure Cylindrical_Warp( ${}^1\mathbf{I}[\mathbf{x}, \mathbf{y}]$ )
{
   $\mathbf{r} = 2\mathbf{W}/\pi$ ;
  for  $\mathbf{u}:=0, \mathbf{Nrows}-1$ 
    for  $\mathbf{v}:=0, \mathbf{Ncols}-1$ 
      {
         ${}^2\mathbf{I}[\mathbf{u}, \mathbf{v}] = 0$ ; // Используется в качестве фонового значения
        if ( $|\mathbf{u} - \mathbf{u}_0| \leq \mathbf{r}$ )
          {
             $\mathbf{x} = \mathbf{x}_0 + \mathbf{r} \arcsin((\mathbf{u} - \mathbf{x}_0)/\mathbf{r})$ ;
             $\mathbf{y} = \mathbf{v}$ ;
             ${}^2\mathbf{I}[\mathbf{u}, \mathbf{v}] = {}^1\mathbf{I}[\text{round}(\mathbf{x}), \text{round}(\mathbf{y})]$ ;
          }
      }
  return( ${}^2\mathbf{I}[\mathbf{u}, \mathbf{v}]$ );
}

```

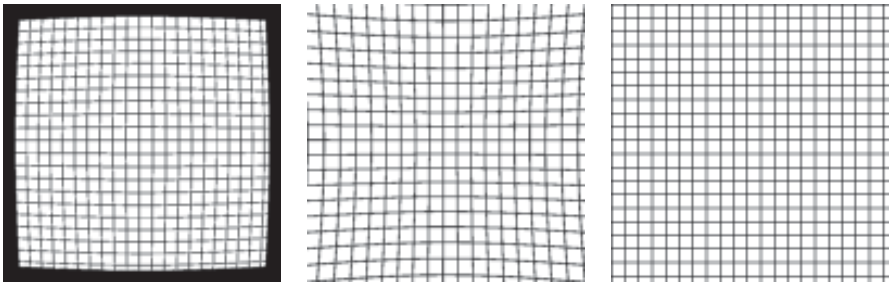
**Алгоритм 11.7.** Цилиндрическая деформация фрагмента изображения.

ходного изображения заключается в использовании *обратного преобразования* для выборки значений из исходного изображения.

**Упражнение 11.20**

(а) Определите преобразование, которое сначала отображает круговую область исходного изображения на полусферу, а затем проецирует полусферу на выходное изображение. Круговая область на исходном изображении задается центром  $(x_c, y_c)$  и радиусом  $r_0$ . (б) Разработайте компьютерную программу для выполнения этого преобразования.

**Устранение радиальной дисторсии.** Радиальная дисторсия есть у большинства объективов. Она может быть незаметна для человека, но иногда приводит к большим ошибкам фотометрических измерений. В таких случаях требуется устранить влияние радиальной дисторсии. Оно заключается в том, что положение видимой на изображении точки искажается пропорционально ее расстоянию от оптической оси. На рис. 11.26 показаны две распространенные разновидности радиальной дисторсии, а также восстановленное изображение.



**Рис. 11.26.** Две разновидности радиальной дисторсии. (*Слева*) бочкообразные искажения. (*В центре*) подушкообразные искажения. Дисторсия может быть устранена с помощью нелинейного преобразования изображения. (*Справа*) изображение после устранения дисторсии.

Если предположить, что оптическая ось проходит вблизи центра изображения, то для компенсации дисторсии можно сместить все точки изображения либо ближе, либо дальше от центра. Величина смещения пропорциональна квадрату расстояния точки от центра изображения. Это преобразование является нелинейным, так как величина смещения непостоянна. Иногда для коррекции радиальной дисторсии используется выражение более высокой степени с четными показателями, которое приведено в виде уравнений (11.26). Точка  $[x_c, y_c]$  является центром изображения, через который, как предполагается, проходит оптическая ось. Для вычисления величины радиальной дисторсии используются первые две четные степени радиального расстояния. Наилучшие значения констант  $c_2$  и  $c_4$  можно определить по результатам эмпирического изучения радиальных смещений известных управляющих точек или с помощью формальной процедуры калибровки с использованием метода наименьших квадратов.

$$\begin{aligned}
 R &= \sqrt{((x - x_c)^2 + (y - y_c)^2)} \\
 D_r &= (c_2 R^2 + c_4 R^4) \\
 x &= x_c + (x - x_c) D_r \\
 y &= y_c + (y - y_c) D_r
 \end{aligned}
 \tag{11.26}$$

**Полиномиальные отображения.** Для устранения многих незначительных глобальных искажений изображений могут применяться полиномиальные отображения, содержащие слагаемые с двумя переменными не выше второй степени. Эти полиномы имеют вид (11.27). Для адаптации этих выражений к различным геометрическим факторам должны быть определены 12 коэффициентов. Для оценки этих коэффициентов необходимо знать координаты минимум шести управляющих точек до и после отображения. (Каждая управляющая точка дает два уравнения.) Однако на практике обычно используется большее количество точек. Отметим, что если в уравнениях (11.27) оставить только три первых слагаемых, то будет получено аффинное

преобразование.

$$\begin{aligned}u &= a_{00} + a_{10}x + a_{01}y + a_{11}xy + a_{20}x^2 + a_{02}y^2 \\v &= b_{00} + b_{10}x + b_{01}y + b_{11}xy + b_{20}x^2 + b_{02}y^2\end{aligned}\quad (11.27)$$

### Упражнение 11.21

Покажите, что уравнение радиальной дисторсии (11.26) при  $c_4 = 0$  можно точно представить в виде полиномиального отображения вида (11.27).

## 11.8 ИТОГИ

В данной главе рассматривались различные понятия и методы, относящиеся к задачам двумерного сопоставления. Главной темой было выполнение двумерного сопоставления с использованием преобразований. Преобразования могут применяться в качестве простых операций обработки изображений для обнаружения или выборки области изображения, для совмещения двух изображений в одной координатной системе, для добавления или устранения искажений двумерных изображений. В главе были рассмотрены алгебраические выражения этих преобразований и связанные с ними методы и приложения. Этот материал будет расширен в гл. 13 применительно к сопоставлению трехмерных моделей и сцен.

Вторая важная тема этой главы заключается в интерпретации двумерных изображений на основе соответствий с двумерными моделями. Общий принцип заключается в *распознавании посредством совмещения*. При этом изображение интерпретируется на основе обнаружения модели и RST-преобразования такого, которое отображает известные структурные компоненты модели на компоненты изображения. Были представлены несколько алгоритмических подходов, в том числе кластеризация по расположению, поиск на дереве интерпретации и метод локальных базовых признаков.

Также были рассмотрены метод дискретной релаксации и реляционное сопоставление. Эти два метода могут применяться в широком круге задач, несмотря на то, что они рассматривались в контексте ограничений в виде геометрических отношений. Реляционное сопоставление потенциально более надежно, чем жесткое совмещение, если отношения сами являются более устойчивыми по сравнению с теми, которые зависят от метрических свойств.

Дефекты изображений, вызванные дисторсиями объектива, наклоном луча зрения, эффектами дискретизации и т. п., могут приводить к нарушению метрических отношений на изображении. Топологические отношения, например, связность, смежность и инцидентность, обычно более устойчивы к таким ошибкам. Успешное сопоставление с использованием топологических отношений на изображении и/или между частями модели может применяться для последующего обнаружения большого количества сопоставленных точек. Эти точки могут затем использоваться для поиска отображающей функции с большим количеством параметров, которые позволяют учесть метрические искажения.

Методы, рассмотренные в данной главе, могут непосредственно применяться во многих практических задачах. В гл. 14 эти методы расширяются на случай трехмерного пространства.

## 11.9 ЛИТЕРАТУРА

В статье [Van Wie, Stein, 1977] обсуждается система для автоматического совмещения спутниковых изображений с картами. Сначала преобразование вычисляется приближенно с учетом времени съема изображения. Приближенное преобразование используется для уточненного поиска управляющих точек с использованием шаблонов. Управляющие точки затем применяются для получения уточненного преобразования.

В книге [Wolberg, 1990] описываются деформации изображений, в том числе подробно рассматриваются вопросы выборки исходного изображения и устранение ступенчатости с помощью сглаживания. Обсуждение двумерного сопоставления методом кластеризации по расположению соответствует работе [Stockman и др., 1982], в которой был также описан пример обнаружения самолетов. Более общее рассмотрение, включающее трехмерный случай, дано в работе [Stockman, 1987]. В статье [Grimson, Lozano-Perez, 1984] продемонстрировано применение ограничений по расстоянию при сопоставлении модельных точек с наблюдаемыми точками данных. Аппроксимация методом наименьших квадратов рассматривается во многих книгах и является самостоятельной глубокой темой. Метод наименьших квадратов часто используется для оценки параметров преобразований при наличии шума с использованием большого количества управляющих точек. В книге [Wolberg, 1990] обсуждается несколько применений метода наименьших квадратов в контексте деформаций изображений. В книге [Daniel, Wood, 1971] рассматривается общая задача аппроксимации.

В некоторых задачах оказывается невозможно найти геометрическое преобразование, глобально применимое на всем изображении. В таких случаях изображение может быть разделено на несколько областей, для каждой из которых определяются собственные управляющие точки и вычисляются отдельные преобразования. Преобразования для соседних областей должны гладко согласовываться вдоль границы между областями. В статье [Goshtasby, 1988] представлен способ выполнения этого требования.

Теоретическое рассмотрение и алгоритмы для задач согласованной маркировки можно найти в статьях [Nalalick, Shapiro, 1979, 1980]. Методы дискретной и непрерывной релаксации были введены в статье [Rosenfeld, Hummel, Zucker, 1976]. Непрерывная релаксация была дополнительно проанализирована в статье [Hummel, Zucker, 1983]. Методы для сопоставления с использованием структурных описаний рассмотрены в работе [Shapiro, Nalalick, 1981]; описание реляционной индексации с использованием 2-графов можно найти в статье [Costa, Shapiro, 1995]. Использование инвариантных атрибутов структурных частей для индексированного доступа к моделям рассмотрено в статье [Chen, Stockman, 1996].

1. Chen, J. L., and G. Stockman. 1996. Indexing to 3D model aspects using 2D contour features. *Proc. Int. Conf. Comput. Vision and Pattern Recog. (CVPR)*, San Francisco, CA (June 18–20), expanded paper to appear in the journal *CVIU*.
2. Clowes, M. 1971. On seeing things. *Artificial Intelligence*, v. 2:79–116.
3. Costa, M. S., and L. G. Shapiro. 1995. Scene analysis using appearance-based models and relational indexing. *IEEE Symposium on Comput. Vision* (Nov. 1995), 103–108.
4. Daniel, C., and F. Wood. 1971. *Fitting Equations to Data*. John Wiley & Sons, Inc., New York.
5. Goshtasby, A. 1988. Image registration by local approximation methods. *Image and Vision Computing*, v. 6(4):255–261.
6. Grimson, W., and T. Lozano-Perez. 1984. Model-based recognition and localization from sparse range or tactile data. *Int. J. Robotics Research*, v. 3(3):3–35.
7. Haralick, R., and L. Shapiro. 1979. The consistent labeling problem I. *IEEE Trans.*, v. PAMI-1:173–184.
8. Haralick, R., and L. Shapiro. 1980. The consistent labeling problem II. *IEEE Trans.*, v. PAMI-2:193–203.
9. Hummel, R., and S. Zucker. 1983. On the foundations of relaxation labeling processes. *IEEE Trans.*, v. PAMI-5:267–287.
10. Lamden, Y., and H. Wolfson. 1988. Geometric hashing: a general and efficient model-based recognition scheme. *Proc. 2nd Int. Conf. Comput. Vision*, Tarpon Springs, FL (Nov. 1988), 238–249.
11. Rogers, D., and J. Adams. 1990. *Mathematical Elements for Computer Graphics*, 2nd ed. McGraw-Hill, New York.
12. Rosenfeld, A., R. Hummel, and S. Zucker. 1976. Scene labeling by relaxation operators. *IEEE Trans. Systems, Man, and Cybern.*, v. SMC-6:420–453.
13. Shapiro, L. G., and R. M. Haralick. 1981. Structural descriptions and inexact matching. *IEEE Trans. Pattern Recog. and Machine Intelligence*, v. PAMI-3(5):504–519.
14. Stockman, G., S. Kopstein, and S. Benett. 1982. Matching images to models for registration and object detection via clustering. *IEEE Trans. PAMI*, v. PAMI-4(3): 229–241.
15. Stockman, G. 1987. Object recognition and localization via pose clustering. *Comput. Vision, Graphics and Image Proc.*, v. 40:361–387.
16. Van Wie, P., and M. Stein. 1977. A LANDSAT digital image rectification system. *IEEE Trans. Geosci. Electron.*, v. GE-15 (July 1977).
17. Winston, P. 1977. *Artificial Intelligence*. Addison-Wesley.
18. Wolberg, G. 1990. *Digital Image Warping*. IEEE Computer Society Press, Los Alamitos, CA.



# Восприятие трехмерных сцен по двумерным изображениям

В этой главе рассматриваются вопросы, связанные с интерпретацией двумерных изображений на основе структуры трехмерных сцен. Люди обладают исключительными способностями в области зрительного восприятия и анализа трехмерных сцен. Людям для этого не требуется никаких усилий, и они обычно не задумываются о механизмах зрительного восприятия. Перед изложением последующего материала сделаем три существенных замечания. Во-первых, хотя при обсуждении восприятия трехмерных сцен существенное внимание уделяется причинам тех или иных явлений, люди способны сразу воспринимать структуру сцен без сознательного анализа. Многие аспекты человеческого зрения в настоящее время еще недостаточно хорошо изучены. Во-вторых, хотя удастся построить хорошие модели для ряда отдельных зрительных операций, очевидно, что в процессе интерпретации сложных сцен в зрении человека одновременно участвуют несколько различных механизмов. В-третьих, объяснение зрительного поведения человека не относится к теме данной книги. Вместо этого в ней рассматриваются способы решения конкретных задач в различных прикладных областях. В таких задачах обычно удается выделить относительно небольшой набор воспринимаемых признаков.

Материал в начале данной главы в основном носит описательный характер. В первом разделе рассматриваются *внутренние изображения*, которые являются промежуточным двумерным представлением для хранения существенных локальных свойств трехмерных сцен. Затем описываются свойства, связанные с текстурой, движением и формой, которые позволяют оценивать свойства трехмерной сцены по ее двумерному изображению. Основное внимание в данной главе уделяется классификации источников информации, а не их математическому моделированию. Тем не менее в последних разделах описываются несколько математических моделей. Среди них модель формирования изображений на основе перспективной проекции, модель стереоскопической системы для вычисления дальности, модель тонкой линзы для оценки разре-

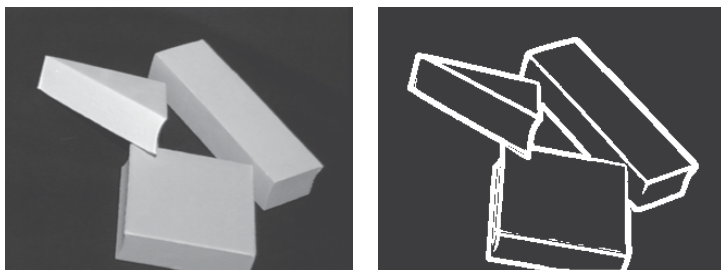
пения и резкости изображений. Еще несколько математических моделей будут рассмотрены в гл. 13.

## 12.1 ВНУТРЕННИЕ ИЗОБРАЖЕНИЯ

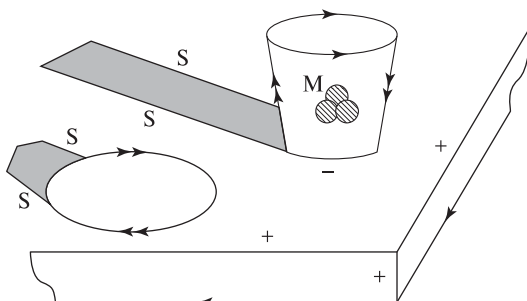
Трехмерную сцену можно попытаться представить в виде композиции элементов поверхностей объектов, которые освещаются источниками света и проектируются на области двумерного изображения. Границы между различными элементами трехмерных поверхностей и между участками различной освещенности образуют на двумерном изображении контрастные края, или *контуры*. Для простых сцен (например, рис. 12.1 и 12.2) можно построить описание, в котором будут представлены все поверхностные элементы и свойства освещения. Некоторые ученые полагают, что основная цель низкоуровневой обработки в зрительной системе человека состоит в формировании подобного представления сцены, которое затем используется в процессе дальнейшей обработки. Это интересный вопрос для дальнейших исследований, но применительно к нашим задачам точный ответ на него не обязателен. Независимо от того, используется или нет какое-либо промежуточное представление сцен в зрительной системе человека, мы будем пользоваться таким представлением сцен и описанием изображений для решения задач методами компьютерного зрения.

На рис. 12.2 схематично изображена сцена, которая содержит пустой тонкостенный стакан и яйцо, расположенные на поверхности стола около его угла. С выбранной точки наблюдения образы стакана и яйца закрывают образ поверхности стола. Стрелки вдоль краев областей обозначают перекрытие поверхностных элементов. Направление стрелки используется для указания закрывающей поверхности. По принятому на рис. 12.2 соглашению, при следовании в направлении стрелки закрывающая поверхность располагается справа. Одиночная стрелка ( $>$ ) обозначает *лезвие* (например, лезвие ножа). При приближении к краю лезвия ориентация закрывающего поверхностного элемента меняется мало. Ориентация закрываемой поверхности после пересечения края лезвия не зависит от ориентации закрывающей поверхности. Все контуры объектов на рис. 12.1 образуют лезвия. На рис. 12.2 лезвия сформированы на нижнем краю стола, где образы краев стола в форме узких плоских фрагментов закрывают неизвестный фон. Верхний край (тонкостенного) бумажного стакана также образует лезвие, так как поверхность стакана закрывает фон и до пересечения границы образа имеет согласованную ориентацию. Более интересным случаем лезвия является верхний контур передней поверхности стакана, которая закрывает его внутреннюю часть.

*Лимбы* ( $\gg$ ) формируются образами гладких трехмерных объектов. Например, лимб может соответствовать образу человеческого тела. По мере приближения на двумерном изображении к границе лимба ориентация соответствующего трехмерного поверхностного элемента изменяется и нормаль к поверхности на границе лимба оказывается перпендикулярна лучу зрения.



**Рис. 12.1.** (Слева) полутоновое изображение сцены с тремя деталями. (Справа) результаты применения детектора краев Превитт с окном  $5 \times 5$ . (С любезного разрешения Deborah Trytten.)



**Рис. 12.2.** Двумерное изображение с маркированными контурами. С помощью меток обозначена связь двумерных контрастных признаков со свойствами трехмерной сцены (например, с ориентацией и освещенностью поверхностных элементов). Метки + и – обозначают сгибы поверхностей. Одиночная стрелка (>) отмечает лезвие, образованное поверхностью справа от стрелки, а двойная стрелка – плавный лимб справа от стрелки. Границы теневых областей отмечены меткой S, а границы отражающих областей – меткой M.

Трехмерная поверхность является *самоперекрывающейся*, это значит, что ее ориентация продолжает непрерывно изменяться после перехода прослеживаемых трехмерных поверхностных элементов на невидимую на двумерном изображении заднюю часть объекта. Лезвие обозначает действительно существующий трехмерный край, а лимб – нет. Граница образа яйца на изображении целиком является границей лимба, а у стакана есть две отдельные границы лимба. Как известно художникам, затенение объекта становится темнее по мере отклонения лимба от направления освещения. Лезвия и лимбы часто называются *разрывными краями*. Для этих краев между закрывающей и закрываемой поверхностью имеется разрыв значений дальности (расстояния). На приведенном ниже рис. 12.10 представлена гораздо более сложная сцена, содержащая много краев тех же типов, что и на рис. 12.2. Например, фонарный

столб и фонарь имеют лимбовые края, а самый правый край здания в левой части рисунка образует лезвие.

---

### Упражнение 12.1

Поставьте на стол перед собой чашку и посмотрите на нее, закрыв один глаз. Для обозначения нормали к поверхности прикоснитесь к чашке карандашом. Убедитесь, что карандаш перпендикулярен лучу зрения.

*Сгибы* образуются в областях резких изменений ориентации поверхности или в местах соединения двух различных поверхностей. На рис. 12.2 сгибы присутствуют на краю стола и в месте соединения стакана и стола. Поверхность на краю стола является выпуклой (метка '+'), а поверхность в месте соединения стакана и стола — вогнутой, что обозначено меткой '-'. В системе машинного зрения, обрабатывающей изображения методом «снизу-вверх», вероятнее всего отсутствовали бы сведения о том, что сцена содержит стакан и стол. Аналогично и человек, глядя на изображение, не сможет определить, приклеен ли стакан к столу или не сделаны ли стакан и стол из цельного куска дерева. Но подобные варианты интерпретации методом «сверху-вниз» существенно расходятся с повседневным опытом. На двумерных полутоновых изображениях сгибы часто (но не всегда) проявляются в виде значительных изменений интенсивности, так как соединенные поверхности относительно источников света обычно располагаются по-разному.

---

### Упражнение 12.2

Краевое изображение треугольной детали на рис. 12.1 содержит шесть контурных отрезков. Какие метки соответствуют этим отрезкам?

---

### Упражнение 12.3

Рассмотрим изображение трех деталей на рис. 1.7 из гл. 1. (Большинство контуров на этом рисунке выделены белым цветом.) Схематично изобразите все контуры этих деталей и присвойте им метки. Достаточно ли рассмотренных типов меток для интерпретации всех контурных сегментов? Понадобились ли метки всех этих типов?

Образование на изображениях контуров еще двух типов в основном объясняется свойствами освещения, а не формой трехмерных поверхностей. Метки типа *знак* ('M') обозначают изменение альбедо поверхностей. Например, метка 'M' на рис. 12.2 обозначает эмблему на стакане, которая выглядит как темный символ на светлой поверхности. Границы освещенных ('I') и теневых ('S') областей формируются вследствие изменения потока световой энергии, падающей на поверхность объекта. Например, количество падающей энергии

может уменьшаться в случае попадания поверхности в область тени других объектов.

В заключение раздела приведем определения, относящиеся к описанию структуры поверхностей на изображении. Важно отметить, что это описание представляет структуру трехмерной сцены на основе особенностей некоторого двумерного изображения этой сцены. Перечисленные в определениях структурные компоненты трехмерных сцен обычно (но не всегда) приводят к формированию на полутоновых изображениях достаточно легко обнаружимых контуров.

**90 Определение. Сгиб (crease)** — это область резкого изменения ориентации поверхности или соединения двух различных поверхностей. При переходе сгиба поверхность остается непрерывной, а нормаль к ней имеет разрыв. Сгиб виден из некоторой окрестности точек наблюдения.

**91 Определение. Лезвие (blade)** соответствует случаю, когда одна непрерывная поверхность закрывает другую поверхность. На границе поверхности угол между нормалью к поверхности и направлением наблюдения меняется гладко и непрерывно. На изображении контур лезвия имеет форму гладкой кривой.

**92 Определение. Лимб (limb)** формируется на изображении, когда одна непрерывная поверхность на своем фоне закрывает другую поверхность. Нормаль к закрывающей поверхности меняется гладко и непрерывно, а на границе лимба становится перпендикулярна к направлению наблюдения. Таким образом, поверхность на изображении закрывает сама себя. Граница лимба на изображении является гладкой кривой.

**93 Определение. Знаком (mark)** называется область, образовавшаяся вследствие изменения отражающих свойств поверхности. Например, из-за рисунка на поверхности или при соединении фрагментов из различных материалов.

**94 Определение. Граница освещенности (illumination boundary)** формируется из-за резкого изменения освещенности поверхности. Это может объясняться изменением свойств источников света или затенением от других объектов.

**95 Определение. Разрывный край (jump edge)** — это лимб или лезвие, для которых характерно скачкообразное изменение глубины по разные стороны края (контура) между поверхностью закрывающего объекта и фоновой закрываемой поверхностью.

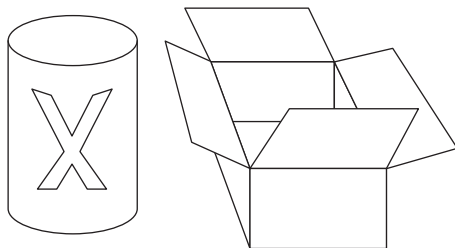
**Упражнение 12.4.** Маркировка отрезков на изображении куба

Нарисуйте куб в стандартном положении, чтобы на рисунке было видно 3 грани, 9 ребер и 7 углов. (а) Предполагая, что куб «висит в воздухе», присвойте каждому из 9 отрезков метку из множества  $\{+, -, >, \gg\}$ . Метка должна давать корректную трехмерную интерпретацию причины формирования отрезка на изображении. (б) Повторите задание (а) в предположении, что куб лежит на плоской поверхности стола. (в) Повторите задание (а) в предположении, что куб является корпусом термодатчика на стене комнаты.

**Упражнение 12.5.** Маркировка изображений повседневных объектов

Назначьте метки отрезкам на рис. 12.3. На нем изображены закрытая банка с содой марки X и открытая пустая коробочка, лежащие на столе.

**Рис. 12.3.** (Слева) закрытая банка соды марки X. На синюю банку нанесен большой оранжевый символ 'X'. (Справа) пустая коробочка с четырьмя открытыми верхними створками. Частично видна нижняя сторона коробочки, не закрытая боковыми стенками.



В гл. 5 рассматривались методы обнаружения контрастных признаков на полутоновых изображениях. Способы описания и прослеживания контуров обсуждались в гл. 10. К сожалению, некоторые пространственные явления приводят к формированию одинаковых признаков на двумерном изображении. Допустим, на двумерном полутоновом изображении был выделен контур. Как решить, чему он соответствует — реальному объекту или тени от какого-либо объекта? Рассмотрим, например, изображение рощи деревьев в солнечный день (или рис. 5.26 из гл. 5, на котором тени ног верблюда демонстрируют аналогичный эффект). Может оказаться, что тени деревьев на траве ('S') будут надежнее выделяться детектором краев, чем границы лимбов ( $\gg$ ) от стволов деревьев. Каким образом при интерпретации изображения отличить образ тени от образа дерева?

**Упражнение 12.6**

С учетом материала из гл. 5 объясните, почему тень от ствола дерева может обнаруживаться надежнее по сравнению с образом ствола дерева.

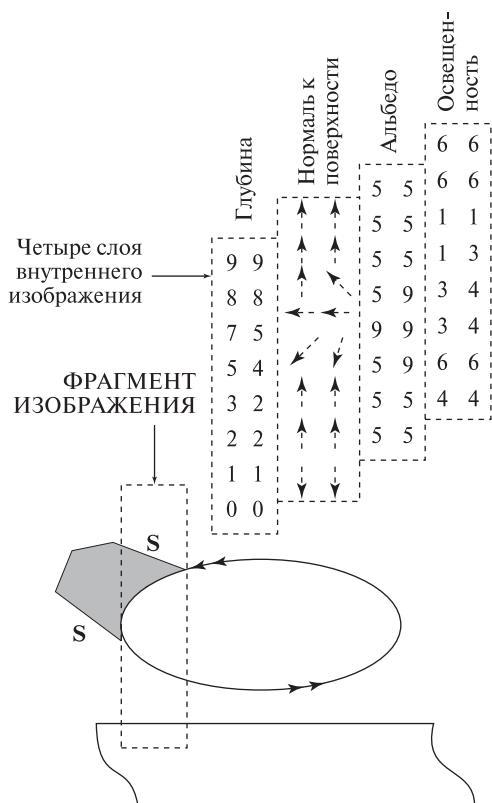
Ряд исследователей предложили использовать в системах машинного зрения *внутренние изображения* (intrinsic image). Например, каждый пиксел

внутреннего изображения может содержать четыре значения. Они представляют внутренние свойства поверхностного элемента сцены, соответствующего данному пикселу:

- **дальность** (глубина);
- **ориентация** (направление нормали);
- **освещенность**;
- **альbedo** или другая характеристика отражающих свойств.

Люди способны достоверно оценивать подобные характеристики сцены в каждом пикселе изображения с учетом окружающего контекста. Автоматическое построение внутренних изображений остается предметом исследований, хотя и не таких интенсивных, как раньше. Для решения многих задач анализа изображений внутренние изображения не требуются. Некоторые методы, которые могут оказаться полезными для построения полных или частичных внутренних изображений, будут рассмотрены в гл. 13.

На рис. 12.4 приведен пример внутреннего изображения, соответствующего фрагменту изображения на рис. 12.2. Показана информация из небольшой по-



**Рис. 12.4.** Внутреннее изображение, соответствующее небольшой полоске на краю яйца на рис. 12.2. Каждый пиксел содержит четыре значения, представляющих глубину поверхности, ее ориентацию, освещенность и альbedo. Более подробные пояснения приведены в тексте.

лосы внутреннего изображения в области конца яйца. В направлении поперек стола значения глубины изменяются постепенно, а на краю стола быстрее. Глубина имеет разрыв там, где поверхность яйца закрывает стол. Ориентация (нормаль) поверхности стола одинакова во всех точках наверху стола и резко меняется на его краю. Ориентация поверхности яйца гладко меняется от точки к точке. Значения альбеда показывают, что стол сделан из более темного (5) материала, чем яйцо (9). Пиксели стола, расположенные в тени, по сравнению с пикселями вне теневой области имеют меньшие значения освещенности (1). Аналогично, пиксели поверхности яйца, соответствующие поверхностным элементам, которые расположены под малым углом к направлению освещения, выглядят темнее (3) пикселей элементов, лежащих прямо перед источником света. Предполагается, что источник света располагается в правом верхнем углу сцены.

---

### Упражнение 12.7. Маркировка отрезков на изображении естественной сцены

Обратитесь к рис. 12.10. Схематично изобразите несколько основных контуров, видимых на этом изображении города Квебек. Назначьте им метки из множества  $\{I/S, M, +, -, >, \gg\}$ .

## 12.2 МАРКИРОВКА КОНТУРНЫХ ИЗОБРАЖЕНИЙ ОБЪЕКТОВ С ПЛОСКИМИ ГРАНЯМИ

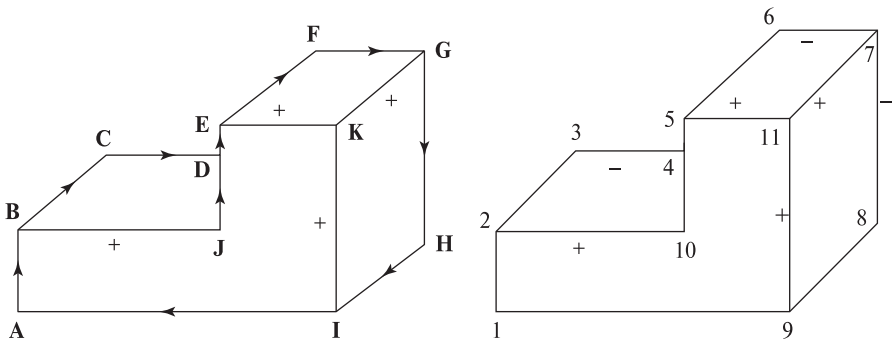
Структура контуров на изображении сильно зависит от структуры трехмерных объектов. В данном разделе эта связь будет продемонстрирована на примере сцен с сильными ограничениями на объекты и условия наблюдения. **Будем полагать, что все углы рассматриваемых трехмерных объектов являются трехгранными, а все грани — плоскими. Углы образуются на пересечении в точности трех граней.** Примером подобного объекта является деталь на рис. 12.5. Для описания трехмерных структурных компонент будем пользоваться терминами *грани*, *сгибы* и *углы*, а для описания образов этих компонент на двумерных изображениях — терминами *области*, *ребра* и *стыки*. Предположим, что имеется контурное двумерное изображение трехмерных объектов, которое состоит из областей, ребер и стыков. Кроме того, будем считать, что при небольших изменениях точки наблюдения топологические свойства изображения сильно не меняются. То есть при этом на изображении не исчезают и не появляются новые грани, ребра или стыки. Такое расположение наблюдаемого объекта часто называется *стандартным положением* (general position).

Множество рассматриваемых трехмерных объектов сильно ограничено и сцены с этими объектами весьма далеки от реальных сцен. Однако методы, разработанные в данном контексте, оказались полезными при решении задач в

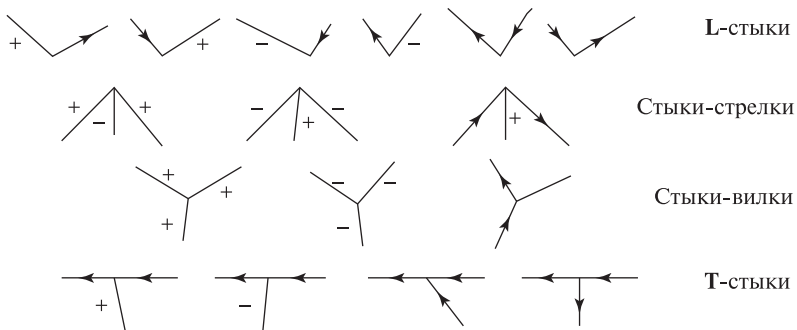


различных прикладных областях. Поэтому набор алгоритмов для сопоставления и интерпретации, рассматривавшихся в гл. 11, будет дополнен алгоритмами для обработки изображений объектов с плоскими гранями. Задачи анализа сцен из объектов с плоскими гранями имеют большое историческое значение и повышают наглядность рассмотрения более современных методов анализа изображений.

В предыдущем разделе обсуждалась маркировка ребер на изображении с помощью меток  $\{+, -, >\}$  для обозначения сгибов или лезвий согласно нашей интерпретации структуры трехмерной сцены. Лимбы на изображениях объектов с плоскими гранями не встречаются. Около тридцати лет назад было обнаружено, что возможные комбинации меток ребер, образующих стыки, сильно ограничены. Всего возможны 16 таких комбинаций. Они приведены



**Рис. 12.5.** Две различные интерпретации одного и того же контурного изображения. (Слева) деталь «висит в пространстве». (Справа) деталь приклеена к стене сзади. На правом рисунке пропущены метки лезвий. Они совпадают с метками левого рисунка.



**Рис. 12.6.** На изображениях объектов с трехгранными углами возможны только 16 топологически различных стыков отрезков. (Все углы в пространстве образованы пересечением трех плоскостей. Объект наблюдается в стандартном положении.) Стыки делятся на четыре типа (сверху-вниз): **L-стыки**, **стрелки**, **вилки** и **T-стыки**.

на рис. 12.6. На рис. 12.5 показано, как эти конфигурации стыков встречаются на двух различных трехмерных интерпретациях одного и того же двумерного контурного изображения.

С учетом количества соединяющихся ребер и значений углов между ними выделяются четыре типа стыков. По очевидным причинам они называются *L-стыками*, *стрелками*, *вилками* и *T-стыками*. В строках на рис. 12.6 показаны примеры стыков различных типов. На рис. 12.5 есть стыки всех четырех типов. Стык с меткой **J** является экземпляром самого левого **L**-стыка в верхней строке на рис. 12.6, а стык **C** — экземпляром второго справа **L**-стыка. Стык **G** является экземпляром крайней справа стрелки из второй строки рис. 12.6. На рис. 12.5 есть только один **T**-стык, обозначенный меткой **D**. На рис. 12.6 обратите внимание, что закрывающее ребро (крест) **T**-стыка не накладывает ограничений на закрываемое ребро. Для него остаются возможными все четыре варианта. На изображении левой детали на рис. 12.5 есть четыре стрелки (**B**, **E**, **G**, **I**). Все они имеют одинаковую (выпуклую) структуру. Однако деталь на рис. 12.5 (справа) содержит один другой (вогнутый) стрелочный стык (**7**). Он указывает на вогнутость в области соединения детали и стены.

Очевидно, что 16 различных стыков можно извлечь из двумерных проекций трехмерных объектов с плоскими гранями. Сложнее доказать, что не существует других стыков (хотя такое доказательство было получено). На данном этапе читатель может непосредственной проверкой убедиться, что других стыков найти не удастся (см. упражнения 12.8 и 12.9).

### Упражнение 12.8

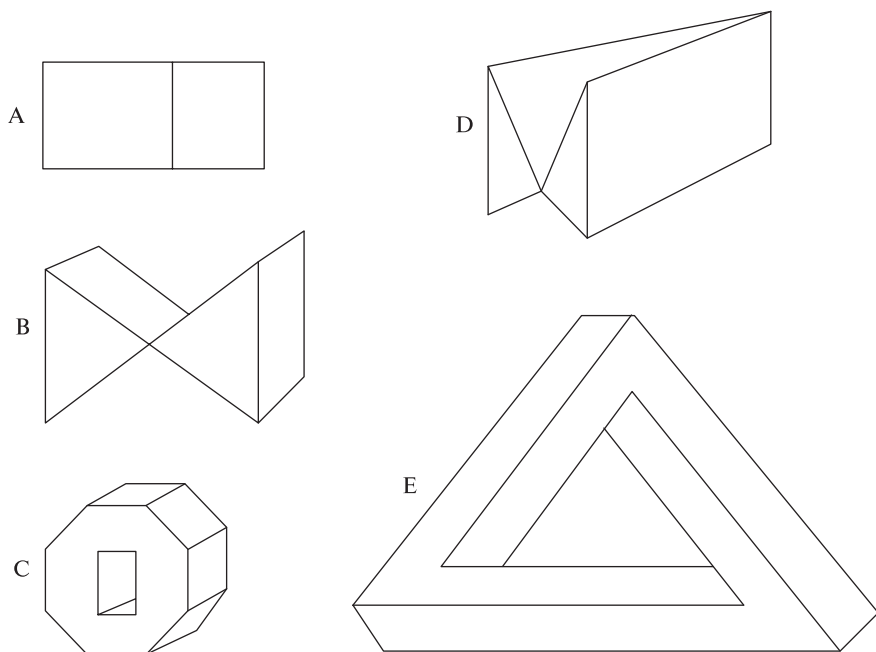
Промаркируйте ребра на рис. 12.1 (справа) в соответствии с собственной зрительной интерпретацией сцены на левом рисунке.

### Упражнение 12.9

Попробуйте назначить метки сгибов и лезвий всем ребрам всех деталей на рис. 12.7. Каждый стык должен совпадать с одним из стыков каталога на рис. 12.6. (а) Для каких контурных изображений удастся построить согласованную маркировку? (б) Какие изображения выглядят похожими на изображения реальных объектов, но не поддаются маркировке? Почему маркировка заканчивается неудачей? (в) Какие контурные изображения выглядят похожими на изображения невозможных объектов? Можно ли построить для них согласованную маркировку?

### Упражнение 12.10

Схематично изобразите и промаркируйте контурное изображение реальной сцены, которая содержит как минимум по два различных экземпляра стыков



**Рис. 12.7.** Контурные изображения, имеющие или не имеющие трехмерную интерпретацию на ограниченном множестве объектов с плоскими гранями.

каждого из четырех типов. При построении изображения собственной сцены можете использовать любые структурные компоненты из рисунков, приведенных в данном разделе.

Для автоматической маркировки контурных изображений могут применяться два алгоритмических подхода, которые рассматривались в гл. 11. Это последовательный перебор с возвратом и параллельная релаксационная маркировка.

Сначала формализуем решаемую задачу. **Дано двумерное контурное изображение, содержащее множество ребер  $P_i$ . Требуется назначить каждому ребру метку  $L_j$ , соответствующую признаку трехмерной сцены, который привел к формированию ребра на изображении. При этом комбинации меток на стыках ребер должны принадлежать каталогу допустимых стыков.**

В контексте гл. 11 ребра соответствуют наблюдаемым объектам, а метки — модельным объектам. Символы  $P$  и  $L$  также были использованы для единообразия с гл. 11, к которой можно обратиться для изучения деталей алгоритмов.

В приведенных далее алгоритмах показаны только наиболее общие операции обработки. В отсутствие какой-либо дополнительной информации оба алгоритма часто генерируют несколько интерпретаций. Один из распростра-

**Назначение согласованной интерпретации всем ребрам графа, представляющего сцену.**

Входные данные: граф, представляющий ребра объекта  $\mathbf{E}$  и стыки  $\mathbf{V}$ .

Выходные данные: отображение множества ребер  $\mathbf{E}$  на множество меток  $\mathbf{L} = \{+, -, >, <\}$ .

- Произвольным образом упорядочим множество ребер:  
 $\mathbf{E} = \{P_1, P_2, \dots, P_n\}$ .
- На очередном  $i$ -м шаге обработки попытаемся маркировать ребро  $P_i$  с помощью следующей непроверенной метки из множества меток  $\mathbf{L} = \{+, -, >, <\}$ .
- Проверка согласованности новой метки со всеми остальными ребрами, которые смежны с текущим ребром  $P_i$  посредством стыка из  $\mathbf{V}$ .
- Если новая назначенная метка приводит к образованию стыка, отсутствующего в каталоге, то выполняется возврат. Иначе пытаемся перейти на следующий шаг обработки.

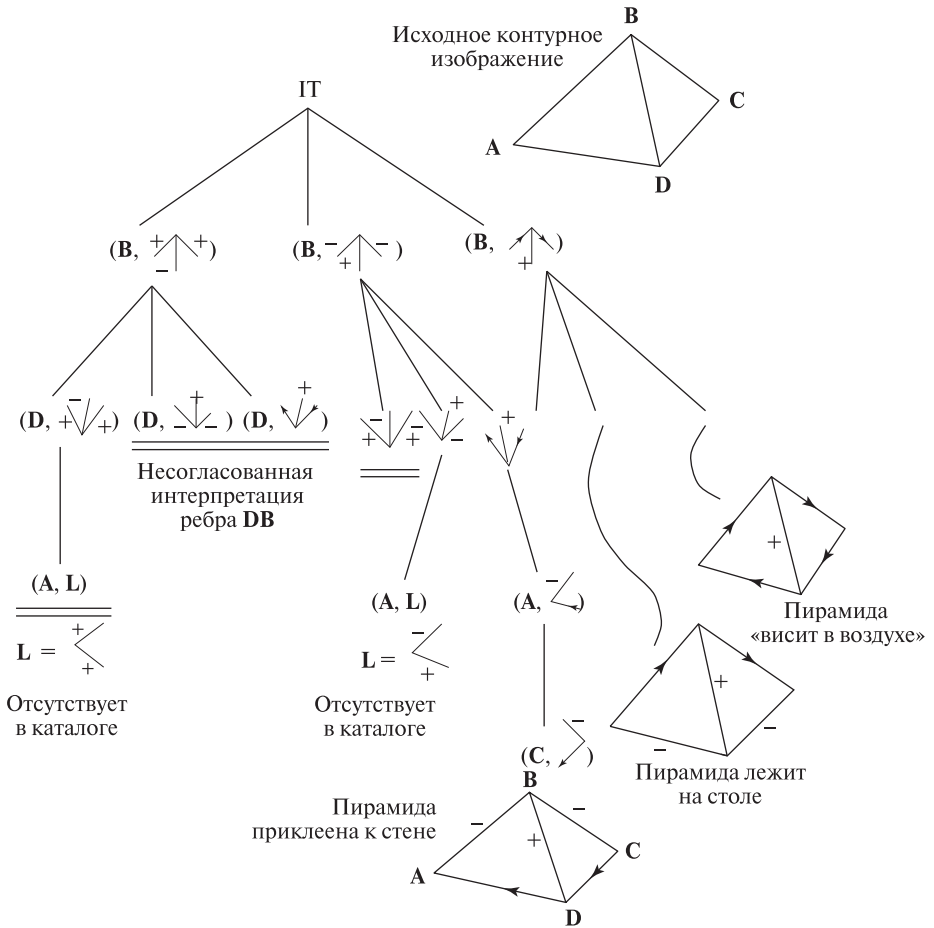
**Алгоритм 12.1.** Маркировка ребер объекта методом перебора с возвратом.

нанных приемов обработки заключается в маркировке всех ребер на выпуклой оболочке контурного изображения в качестве лезвийных ребер ( $' > '$ ). При следовании вдоль этих ориентированных ребер выпуклая оболочка оказывается с правой стороны.

Когда возможно, ребрам сначала назначаются метки с наибольшими ограничениями. Соответствие ребер определенным пространственным признакам, например, сгибу, может устанавливаться на основе некоторой дополнительной информации (например, полученной с помощью стереоскопической системы). Определение типа каждого стыка согласно углам и количеству соединенных ребер может выполняться на этапе предварительной обработки. В других вариантах подобной обработки меткам стыка назначаются варианты интерпретации из каталога и затем исключаются те назначения, которые оказываются несовместимы с интерпретациями соседних стыков. На рис. 12.8 показано дерево интерпретации для контурного изображения четырехгранной пирамиды. Пространство поиска довольно мало. Это объясняется наличием сильных ограничений на множестве объектов с плоскими гранями и трехгранными углами.

**Упражнение 12.11**

Завершите дерево интерпретации, показанное на рис. 12.8. Для этого укажите все ребра и узлы, пропущенные в правой части дерева.



**Рис. 12.8.** Дерево интерпретации для контурного изображения пирамиды, показанного в правой верхней части рисунка. На каждом уровне дерева один из четырех стыков маркируется с использованием одной из 16 меток стыков из каталога на рис. 12.6. На первом уровне дерева выбирается интерпретация стыка В. На последующих уровнях дерева назначаются интерпретации стыкам D, А и С. Три полных пути на дереве дают три полных интерпретации, приведенных в правой нижней части рисунка.

**Упражнение 12.12**

Постройте пятиуровневое дерево интерпретации для назначения согласованных меток всем ребрам пирамиды, показанной на рис. 12.8. Во-первых, сформулируйте задачу с использованием системы обозначений из задачи согласованной маркировки из гл. 11. На основе 5 наблюдаемых ребер и 4 допустимых

**Назначение согласованных интерпретаций всем ребрам графа сцены.**

Входные данные: граф, представляющий ребра объекта  $\mathbf{E}$  и стыки  $\mathbf{V}$ .

Выходные данные: отображение множества ребер  $\mathbf{E}$  на *подмножества* множества меток  $L = \{+, -, >, <\}$ .

- На этапе инициализации каждому ребру  $P_i$  назначаются все метки  $\{+, -, >, <\}$ .
- На каждом шаге выполняется фильтрация меток каждого ребра посредством следующей обработки всех остальных ребер:
  - Если метка  $L_j$  не образует допустимого стыка совместно с допустимыми метками ребер, соединенных с ребром  $P_i$ , то метку  $L_j$  надо исключить из множества меток  $P_i$ .
- Итерации прекращаются, если на очередном шаге фильтрации ни одно множество меток не уменьшилось.

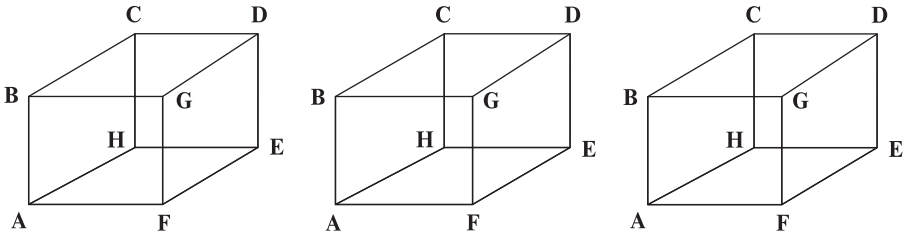
**Алгоритм 12.2.** Маркировка ребер объектов с плоскими гранями методом дискретной релаксации.

меток ребер определите множества  $P, L, R_P, R_L$ . Во-вторых, изобразите дерево интерпретации. Содержит ли оно три полных пути, соответствующих трем полным путям на рис. 12.8?

**Релаксационная маркировка контурного изображения.** Для ограничения возможных интерпретаций сегментов контурного изображения может применяться алгоритм дискретной релаксации, описанный в гл. 11. Далее будет рассмотрена процедура назначения меток ребрам на контурном изображении. Похожая процедура может быть построена для назначения меток стыкам.

Существует много разновидностей алгоритмов релаксационной маркировки. Алгоритм 12.2 является одним из простейших представителей этого семейства. Основные операции этого алгоритма могут выполняться в любом порядке или даже параллельно в пределах каждого шага обработки.

В результате исследований было выяснено, как при обработке изображений сцен с объектами с плоскими гранями можно учитывать ограничения на основе данных об интенсивности изображения и как работать с изображениями различного разрешения. Область анализа сцен с объектами с плоскими гранями оказалась очень интересной и в ней были получены существенные результаты. Однако в минимальном варианте, рассматривавшемся в данном разделе, приемы обработки сцен объектов с плоскими гранями будут неприменимы для большинства реальных сцен. Это объясняется тем, что (а) большинство трех-



**Рис. 12.9.** (См. Упражнение 12.13.) *Куб Некера* имеет несколько интерпретаций. Это можно заметить при внимательном наблюдении любой из этих фигур. Например, в различных вариантах интерпретации центрального рисунка два стыка-вилки по-разному выбираются в качестве образов переднего или заднего угла.

мерных объектов не удовлетворяют сделанным предположениям и (б) доступные двумерные изображения обычно весьма далеки от требуемых контурных изображений. Методы анализа сцен с объектами с плоскими гранями были усовершенствованы по ряду направлений. Например, каталоги меток и стыков дополнялись для учета образов объектов с криволинейными поверхностями и для локализации ошибок на контурных изображениях. Работы по данной теме перечислены в списке литературы в конце главы.

### Упражнение 12.13. Демонстрация эффекта Некера

Это упражнение напоминает предыдущее упражнение 12.12, в котором рассматривалась маркировка ребер изображения пирамиды в стандартном положении. На рис. 12.9 показаны каркасные изображения куба. Грани отсутствуют, поэтому перекрытий нет и все 12 стибов видны как ребра на контурном изображении. (а) Посмотрите на крайнее слева изображение. Формирует ли ваша зрительная система трехмерную интерпретацию этого контурного изображения? Меняется ли эта интерпретация при пристальном наблюдении в течении нескольких минут? (б) Промаркируйте центральное изображение таким образом, чтобы стык G был образом переднего угла. Удалите стык H и три входящих в него ребра так, чтобы на рисунке был изображен сплошной куб. (в) Повторите задание (б) так, чтобы на этот раз стык H был образом переднего угла и удалите ребра, входящие в стык G. *Трехмерный каркасный объект HE является корректным объектом на множестве объектов с плоскими гранями. Однако для интерпретации малых окрестностей любого угла куба можно пользоваться приведенным выше каталогом стыков.*

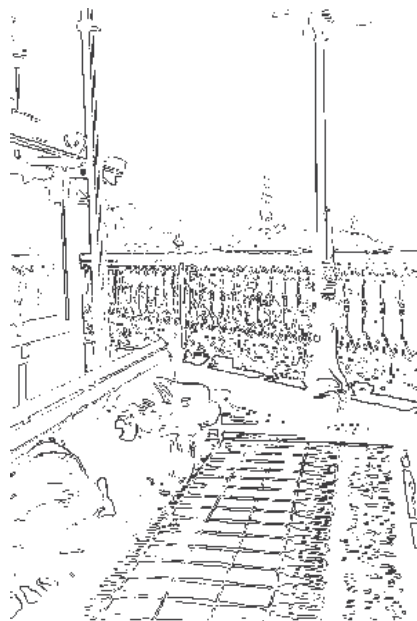
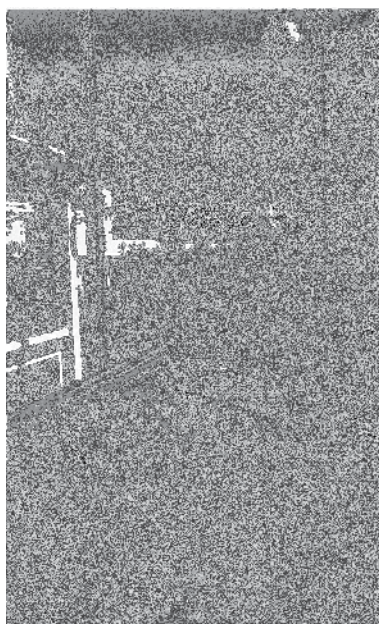
### Упражнение 12.14

Примените метод поиска на дереве интерпретации из гл. 11 для контурных изображений на рис. 12.7. Найдите какие-нибудь корректные маркировки, соответствующие полным путям на дереве интерпретации.

### 12.3 ТРЕХМЕРНЫЕ ПРИЗНАКИ НА ДВУМЕРНЫХ ИЗОБРАЖЕНИЯХ

Изображения часто являются двумерными проекциями трехмерных сцен. Однако каждый, кто знаком с живописью или кино, знает, что двумерное изображение способно вызывать ощущение восприятия трехмерного пространства. Известно много признаков, которые можно добавить на двумерное изображение для обеспечения его трехмерной интерпретации.

На изображении на рис. 12.10 имеется несколько различных дальностных признаков. Образы двух спящих людей загораживают образ скамьи, которая загораживает столб, а он в свою очередь перекрывает витиеватую ограду. Ограда закрывает образы деревьев, которые закрывают здания со шпильями, а они перекрывают образ неба. Солнце расположено далеко справа. На это указывает тень от правого фонарного столба, а также то, что правая сторона этого столба ярче левой. Невидимая на изображении правая часть ограды отбрасывает узорчатую тень на пол и формирует ложное впечатление мозаичного патио. Текстура на полу обозначает плоскую поверхность, а сжатие текстуры означает постепенное удаление этой поверхности от наблюдателя. Человек легко может определить ориентацию стены здания слева на основе оценки ориентации краев в этой области изображения. Образ ограды кони-



**Рис. 12.10.** Вид города Квебек с холма над рекой Святого Лаврентия. (Слева) изображение, содержащее большое количество дальностных признаков. (Справа) результаты применения детектора краев Робертса с последующей бинаризацией для выборки пикселей со значениями не менее 90% от максимального значения.



чески сужается в направлении справа-налево и дает значимый признак того, что пространственное расстояние до ограды увеличивается. Аналогично, доски скамьи конически сходятся в направлении слева-направо. Образы фонарного столба и людей намного больше образов шпильей. Это означает, что шпильей зданий расположены гораздо дальше от наблюдателя.

**96 Определение.** Промежуточное расположение (interposition) — это дальностный признак, который формируется, когда один объект закрывает другой перед наблюдателем. Этот признак означает, что закрывающий объект расположен к наблюдателю ближе закрываемого объекта.

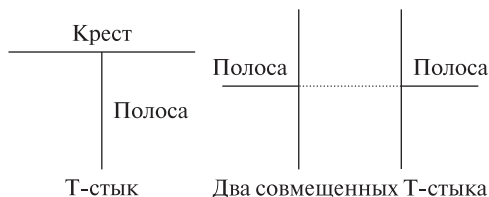
### Упражнение 12.15

Примените процедуру релаксационной маркировки из гл. 11 к контурным изображениям на рис. 12.7. Если множество меток какого-либо ребра становится пустым, то это означает отсутствие согласованной интерпретации. Если какие-либо ребра после завершения маркировки имеют больше одной метки, значит, были получены неоднозначные интерпретации, которые могут генерироваться данным алгоритмом. В подобном случае нанесите несколько полученных меток на контурное изображение и проверьте, какие метки действительно допустимы.

### Упражнение 12.16

Найдите все Т-стыки ребер коробки на рис. 12.3. Соответствуют ли все стыки действительным перекрытиям поверхностей?

Как было продемонстрировано выше, промежуточное расположение существенно упрощает интерпретацию изображения на рис. 12.10. Очевидно, что скамья расположена ближе к наблюдателю, чем закрываемый ею фонарь, а фонарный столб ближе, чем ограда. Распознавание отдельных объектов может помочь при использовании этого дальностного признака, однако это



**Рис. 12.11.** Т-стыки обозначают перекрытие одного объекта другим. Край креста соответствует закрывающему объекту, а края полосы — закрываемому объекту. По сравнению с одним стыком два совмещенных Т-стыка обеспечивают более значимый признак наличия перекрытия объектов.

не обязательно. Т-стыки на контурах изображения обеспечивают сильные локальные ограничения. Рассмотрим рис. 12.11. Обратите внимание, что на краевом изображении на рис. 12.10 (справа) край здания выглядит как полоса с Т-стыком в верхней части скамьи, а ограда образует полосу с Т-стыком с правым краем фонарного столба. Сопоставление пары Т-стыков дает даже более значимый признак, так как свидетельствует о наличии непрерывного объекта позади другого объекта. Краевое изображение на рис. 12.10 достаточно сложное, как обычно и бывает в случае естественных сцен. В следующем упражнении рассматривается более простой случай. **Промежуточное расположение распознаваемых объектов или поверхностей может использоваться для вычисления относительной глубины этих объектов.**

**97 Определение.** Обрато пропорциональная зависимость дальности объекта и размеров его образа на изображении дает признак **перспективного масштабирования** (perspective scaling). Термин *масштабирование* используется для сравнения размеров объектов, которые заданы в плоскости, параллельной плоскости изображения.

После распознавания шпилей зданий на рис. 12.10 можно заключить, что они довольно далеки от наблюдателя, так как их размеры на изображении малы. При просмотре изображения справа-налево расстояние до вертикальных элементов ограды увеличивается, и поэтому размеры их образов уменьшаются. Аналогично, если посмотреть с высокого здания вниз, то высоту можно оценить на основе видимых размеров людей и автомобилей на улице внизу. **Размеры распознанного на изображении объекта могут использоваться для вычисления пространственной глубины этого объекта.**

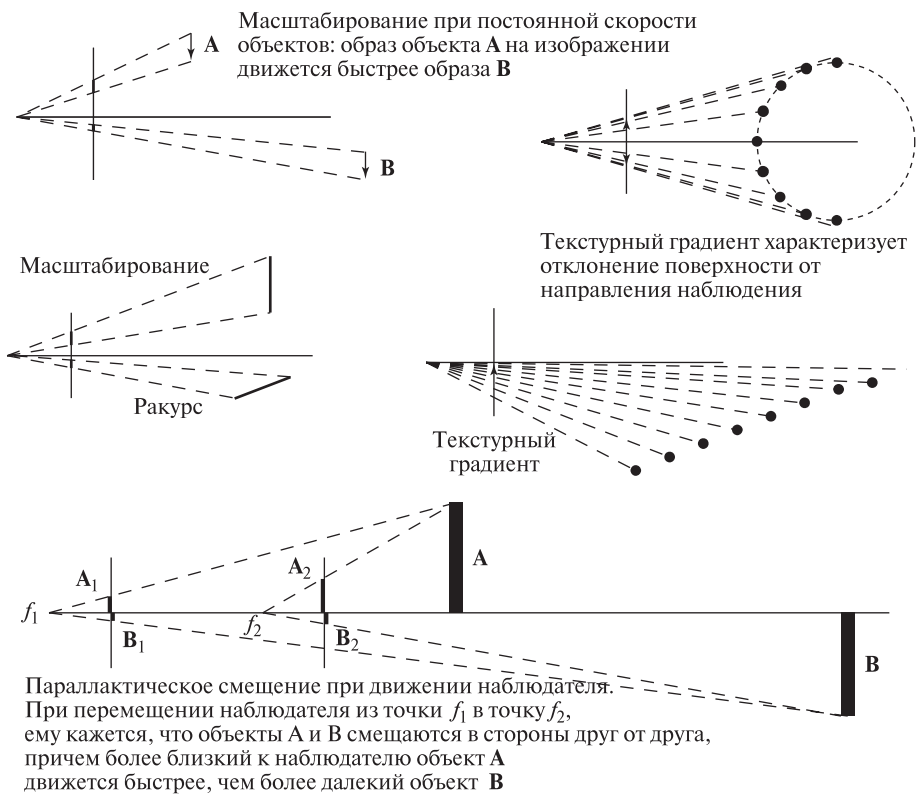
**98 Определение.** **Эффект ракурса** (foreshortening) на изображении объекта возникает при наблюдении объекта под острым углом к его оси симметрии. Ракурс является еще одним существенным признаком, связывающим двумерный образ со структурой трехмерного объекта.

Размеры образов скамьи и людей на рис. 12.10 меньше, чем было бы в случае расположения скамьи параллельно плоскости изображения. Аналогично, образы вертикальных элементов ограды располагаются плотнее по мере удаления от наблюдателя. Этот эффект ракурса отсутствовал бы при направлении наблюдения, перпендикулярном плоскости ограды.

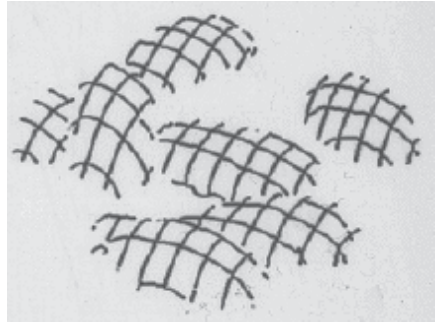
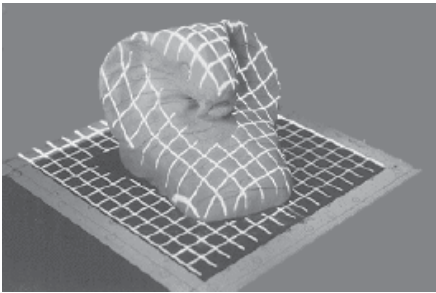
Текстурный градиент возникает похожим образом. Элементы текстуры подвергаются масштабированию и ракурсу, и результирующие изменения текстуры дают наблюдателю информацию о расстоянии и ориентации текстурированной поверхности. Этот эффект легко заметить, если посмотреть на кирпичное здание, мозаичный пол или железнодорожные пути, на кукурузное поле или толпу на стадионе. Текстурный градиент демонстрируется на рис. 12.12. Возникновение текстурного градиента на изображении поясняется на схеме на рис. 12.13. По направлению к центру изображения тексели, или образы точек, сдвигаются все ближе друг к другу, что соответствует увеличению глубины



**Рис. 12.12.** Изображение кукурузного поля содержит несколько текстур (растения и грядки растений) и текстурные градиенты. В направлении снизу-вверх на изображении текстуры становятся более плотными, так как на каждый участок изображения попадает все больше образов листьев кукурузы. (С любезного разрешения John Gerrish.)



**Рис. 12.13.** Схемы возникновения эффектов масштабирования, ракурса, текстурного градиента и параллакса при движении наблюдателя. На каждой схеме передняя плоскость изображения обозначена вертикальным отрезком, а объекты расположены справа от нее.



**Рис. 12.14.** Применение структурированного освещения в виде шаблона световых полос для формирования текстуры, выявляющей форму трехмерной поверхности. (Слева) сетка световых полос спроектирована на гладкую скульптуру, расположенную на плоском основании. (Справа) изображение световых полос на объектах после удаления плоского фона. Что это за объекты? (С любезного разрешения Gongzhu Hu.)

соответствующих пространственных объектов. На рис. 12.14 показано формирование текстуры при освещении объектов сцены с использованием шаблона в виде регулярной сетки световых полос. Это искусственное (структурированное) освещение не только дает больше информации о форме поверхностей при восприятии человеком, но также может применяться для автоматического вычисления нормалей к поверхностным элементам или даже их глубины. Эти вопросы рассматриваются в следующей главе. В целом, изменение текстуры на изображении может использоваться для вычисления ориентации трехмерной поверхности, содержащей эту текстуру.

**99 Определение. Текстурным градиентом** (texture gradient) называется изменение текстуры на изображении (измеряемое или воспринимаемое человеком) вдоль некоторого направления на изображении. Этот признак часто соответствует изменению глубины или ориентации трехмерных поверхностей, соответствующих данной текстуре на изображении.

Если нормаль к текстурированной поверхности не совпадает с направлением наблюдения, то эти поверхности приводят к формированию текстурного градиента на изображении. Обратное утверждение не всегда верно. Этот эффект часто используется художниками для создания иллюзии трехмерных поверхностей на плоском холсте.

**100 Определение. Параллактическое смещение** дает движущемуся наблюдателю информацию о глубине объектов (даже для неподвижных объектов, так как они кажутся движущимися относительно друг друга). Образы более близких объектов движутся быстрее, чем образы более далеких.

Параллактическое смещение при движении наблюдателя объясняется перемещением наблюдателя. Похожий эффект возникает в случае неподвижного наблюдателя и движущихся объектов. Он поясняется на рис. 12.13 наряду с несколькими другими дальностными признаками изображений, получаемых на основе перспективной проекции. Например, когда мы идем по улице (предположим, закрыв один глаз), то образы боковых объектов, мимо которых мы проходим (скамейки, дверные проемы и т. п.), движутся поперек сетчатки намного быстрее, чем образы таких же объектов, расположенных в нескольких десятках метров впереди. При управлении машиной встречные далекие машины формируют устойчивые образы, но на близком расстоянии их образы проходят поперек боковых окон очень быстро. Аналогично, образы автомобилей, проезжающих мимо стоящего человека, кажутся ему намного более быстро движущимися, чем образы далеких автомобилей. Математические выражения перспективной проекции позволяют описать связь параллактического смещения с масштабированием и ракурсом.

В данном разделе были рассмотрены не все известные дальностные признаки, которые могут встретиться на отдельных двумерных изображениях. Например, цвет удаленных объектов может казаться более синим, чем цвет таких же близких объектов. Или, образы далеких объектов могут терять резкость из-за рассеивающих явлений в атмосфере между объектами и наблюдателем. В гл. 13 описывается выделение дальностных признаков на основе фокусировки объектива. Кроме того, в данном разделе не учитывались ряд допущений о реальном пространстве, например, мы не предполагали наличие опорной плоскости (земной поверхности) или силы тяжести, выделяющей особое вертикальное направление. Это существенные характеристики среды, в которой формировалась зрительная система человека.

---

### Упражнение 12.17

Закройте один глаз и посмотрите на карандаш. Держите карандаш параллельно прямой между вашими глазами и переместите его от лица на расстояние вытянутой руки. Заметно ли изменение размеров образа из-за масштабирования, ракурса или совместного влияния этих эффектов? Теперь возьмите карандаш в центре тяжести и сохраняйте постоянное расстояние между глазом и центром карандаша. Поверните карандаш вокруг центра и пронаблюдайте изменение его образа. Заметны ли изменения из-за масштабирования или ракурса? Запишите приближенную тригонометрическую формулу для вычисления размера образа карандаша в зависимости от угла поворота.

---

### Упражнение 12.18

Расположите палец вертикально и близко перед своим носом. По очереди открывайте глаза на одну-две секунды. Пронаблюдайте кажущееся движение

вашего пальца (который в действительности не должен двигаться). Переместите палец чуть дальше и повторите опыт. Переместите палец на длину вытянутой руки и повторите опыт еще раз. (Можно совмещать кончик пальца с дверной ручкой или другим более далеким объектом.) Опишите величину кажущегося смещения в зависимости от расстояния между пальцем и носом.

## 12.4 ДРУГИЕ СПОСОБЫ ОПРЕДЕЛЕНИЯ ПРОСТРАНСТВЕННЫХ СВОЙСТВ ОБЪЕКТОВ

В гл. 10 обсуждалось несколько способов группировки характерных признаков изображения в более крупные структуры, например, с учетом связности объектов. Эти способы могут оказаться полезными для построения трехмерной интерпретации изображения. Но они также способны ввести в заблуждение, так как в некоторых случаях приводят к некорректным вариантам интерпретации. В данном разделе кратко рассматривается еще ряд явлений, которые могут использоваться для интерпретации структуры трехмерных сцен по характерным признакам двумерных изображений.

### 12.4.1 Определение формы объектов по одному признаку

В 1980-х годах было выполнено большое количество работ в области построения вычислительных моделей для определения формы поверхности на основе различных признаков изображения. Исследователи обычно изучали возможности использования некоторого избранного признака изображения, а не их комбинации. Несколько таких математических моделей рассматриваются в гл. 13. Явления, лежащие в их основе, будут более подробно описаны далее. В этом разделе перечисляются особенности изображений, которые часто используются в качестве трехмерных дальностных признаков.

**Определение формы по освещенности.** При изучении художественной графики способы закраски и затенения объектов рассматриваются в качестве важного приема для передачи пространственных свойств объектов на двумерном изображении. На картинах сцен, содержащих гладкие объекты (например, яблоко), на образах этих объектов часто изображаются блики. Эти блики располагаются в окрестности точек поверхности, в которых луч, падающий от источника света, и луч, отраженный по направлению к наблюдателю, образуют равные углы с нормалью к поверхности. Поверхностные элементы гладких объектов становятся все более темными по мере увеличения угла между направлением освещения и нормалью к поверхности. Плоские поверхности на изображении обычно выглядят однородными. Интенсивность их образов на изображении пропорциональна углу между нормалью к плоскости и направлением освещения.

Было разработано несколько методов определения формы поверхностей с использованием математических формул для вычисления нормали к поверхно-

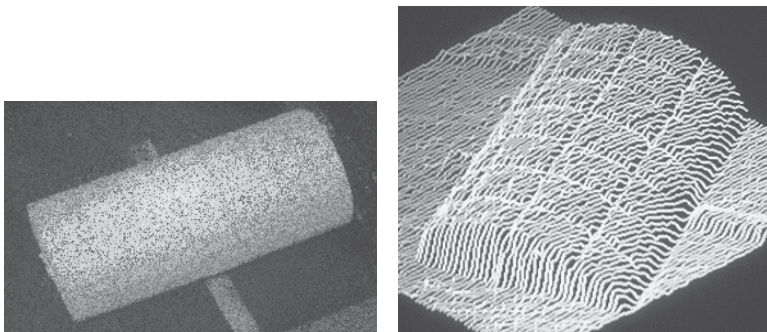


сти по значениям интенсивности пикселей изображения. Однако большинство этих методов требуют калибровки для оценки параметров модели, описывающей связь освещенности и ориентации поверхности. Некоторые методы предполагают применение нескольких камер. В общем случае в моделях для определения формы по данным освещенности должны учитываться следующие величины: количество падающей и отраженной световой энергии, направление падающего и отраженного света, ориентация и коэффициент отражения поверхностного элемента. Необходимость учета большого количества параметров приводит к тому, что методы вычисления формы по данным освещенности оказываются работоспособны только в контролируемой среде с управляемым освещением.

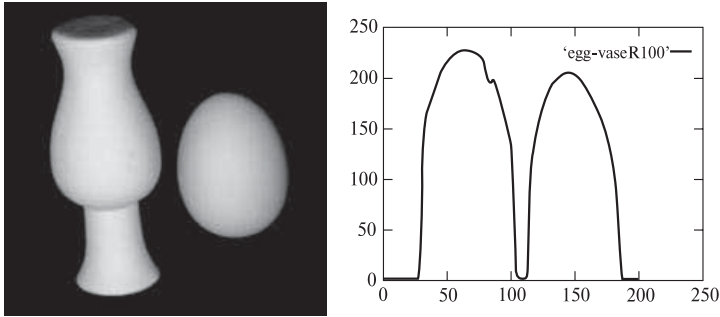
На рис. 12.15 представлен цилиндр с нанесенной сеткой. Этот цилиндр освещен одним источником света. На рис. 12.16 показано изображение матовых объектов. Данные освещенности, представленные в виде значений интенсивности пикселей данного изображения, позволяют оценить форму поверхностей.

**Определение формы по текстурной информации.** Если предполагается, что наблюдаемая однотипная текстура расположена на некоторой трехмерной поверхности, то для вычисления ориентации этой поверхности может использоваться текстурный градиент на двумерном изображении. Понятие текстурного градиента уже описывалось выше. На рис. 12.18 показана однородная текстура трехмерной поверхности, при наблюдении которой на двумерном изображении формируется текстурный градиент. Для описания взаимного расположения поверхности и наблюдателя используются две угловые величины — *наклон* и *скос*.

**101 Определение. Наклон (tilt)** плоской поверхности определяется как направление проекции нормали к поверхности на изображении. На рис. 12.18



**Рис. 12.15.** (Слева) изображение искусственно освещенного цилиндра, представляющего собой банку, которая обернута в клетчатую бумагу. (Справа) трехмерный график функции интенсивности, соответствующий точке наблюдения, которая была немного смещена относительно точки наблюдения изображения слева. Обратите внимание, что форма цилиндра хорошо передается данными интенсивности.



**Рис. 12.16.** Диффузное отражение от гладких матовых объектов — вазы и яйца. Справа приведен график интенсивностей пикселей вдоль выделенной строки. Форма кривой интенсивности связана с формой объекта. (С любезного разрешения Deborah Trytten.)

угол наклона измеряется относительно направления строк изображения (ось X). **Скос** (slant) поверхности — это угол между нормалью к поверхности и лучом зрения (также см. рис. 12.18).

Предположим, что прямо стоящий человек смотрит вперед на пшеничное поле, которое кажется ему плоским. При вертикальном положении головы наклон поля будет равен  $90^\circ$ . Если человек смотрит далеко вперед, то скос близок к  $90^\circ$ . Если человек смотрит прямо под ноги, то скос близок к 0. При повороте головы на  $45^\circ$  влево наклон поля будет равен  $45^\circ$ . Поворот головы на  $45^\circ$  вправо привел бы к наклону поверхности поля, равному  $135^\circ$ .

На изображении на рис. 12.19 двумя основными плоскими поверхностями являются тротуар и стена террасы. Наклон тротуара равен  $90^\circ$ , а скос равен примерно  $75^\circ$ . (Тротуар поднимается вверх под углом около  $15^\circ$ .) Стена террасы имеет наклон около  $170^\circ$  и скос примерно  $70^\circ$ .

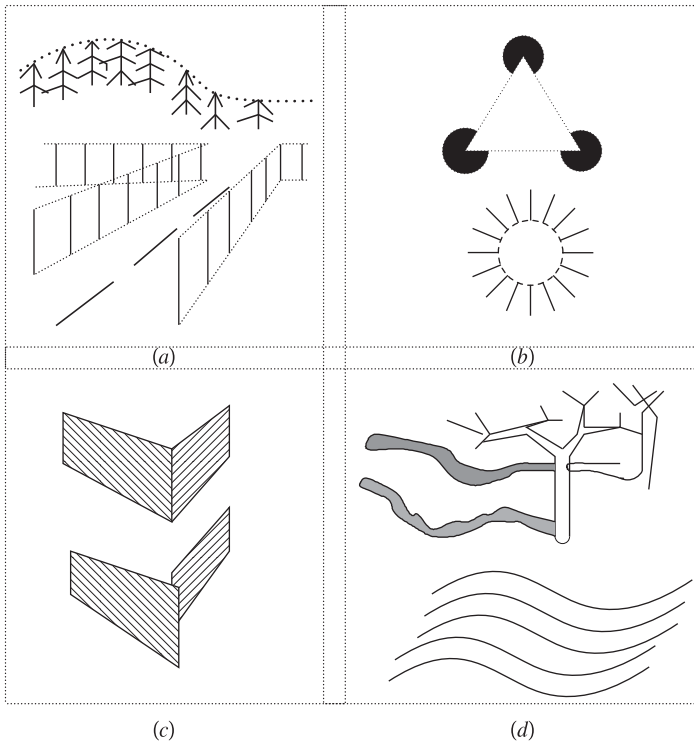
Понятия наклона и скоса применимы к любым поверхностям, а не только к тем, которые расположены близко от земной поверхности. Например, это внешние и внутренние стены зданий, стороны коробок и вагонов. Наклон и скос могут также использоваться в качестве характеристик криволинейных поверхностных элементов. Но у криволинейных поверхностей нормаль к поверхности не имеет постоянного значения, поэтому усложняется вычисление текстурного градиента на изображении.

### Упражнение 12.19

- Определите наклон и скос каждой из четырех граней объекта на рис. 12.5.
- Повторите задание для граней объектов на рис. 12.1.

**Определение формы по контурам.** Люди способны оценивать форму трехмерных объектов на основе формы контуров образов этих объектов на



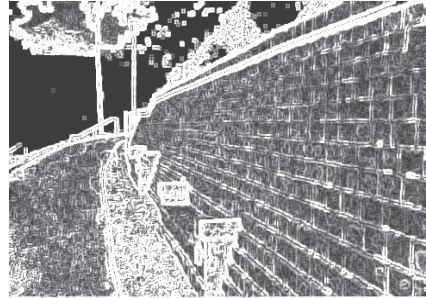
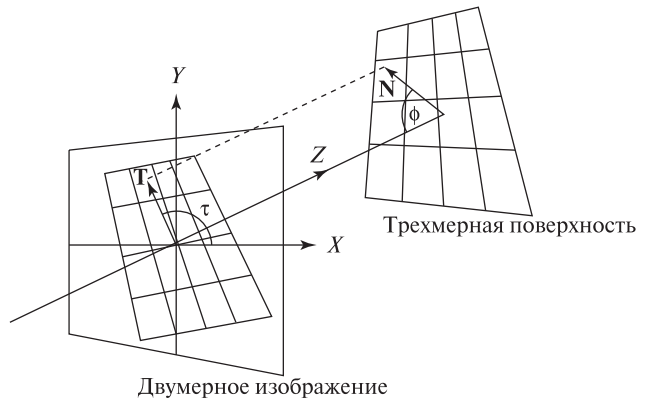


**Рис. 12.17.** Ряд дальностных признаков для оценки пространственных свойств по характерным признакам двумерного изображения. (a) Виртуальные прямые и кривые, которые могут образовываться группами похожих признаков. (b) Виртуальные контуры могут создавать у людей иллюзию восприятия объектов, которые кажутся более яркими, чем окружающий фон. (c) Совмещенность двумерных образов обычно, но не всегда, указывает на совмещенность объектов в трехмерном пространстве. (d) Форма кривых на двумерном изображении упрощает восприятие формы трехмерной поверхности.

двумерном изображении. Например, в качестве трехмерной интерпретации видимого на изображении эллипса человек легко укажет диск или сферу. В случае однородной текстуры или затенения человек, скорее всего, выберет диск. При согласованном изменении текстуры или затенения образа по направлению к контуру, человек, вероятно, предпочтет интерпретацию сферы. Но эскизы и другие контурные рисунки часто не содержат затенения или текстур. Однако люди способны определять пространственную форму объектов даже по контурам их образов.

Были разработаны методы для вычисления нормалей к поверхности в точках внутри области, ограниченной гладкой кривой. Рассмотрим простой случай кругового контура. Предположение о гладкости означает, что в трехмерном

**Рис. 12.18.** Наклон и скос поверхности определяются на основе ориентации нормали к поверхности  $\mathbf{N}$  относительно системы координат наблюдателя. Наклон  $\tau$  — это направление проекции  $\mathbf{N}$  на изображение (проекция обозначена вектором  $\mathbf{T}$ ). Скос  $\phi$  — это угол между  $\mathbf{N}$  и лучом зрения.



**Рис. 12.19.** (Слева) изображение с различными текстурами. (Справа) результаты применения детектора краев Превит с маской  $5 \times 5$ . Тротуар имеет наклон  $90$  градусов и скос примерно  $75$  градусов. Наклон кирпичной стены равен примерно  $170$  градусов, а скос — примерно  $70$  градусов.

в пространстве нормали к поверхности на лимбе объекта перпендикулярны и к лучу зрения, и к элементам на круговом поперечном сечении, образ которого виден на изображении. Таким образом, в контурных точках изображения можно уникальным образом определить направление нормали. В концевых точках диаметров кругового контура нормали направлены противоположно. Вдоль диаметра направление нормали можно интерполировать, обеспечивая корректное направление нормали в средней точке диаметра. Конечно, для определения корректности направления необходимо дополнительное ограничение. Например, оно могло бы учитывать различие эллиптической и сферической поверхностей или то, что наблюдатель смотрит на сферическую оболочку изнутри. Подобные предположения позволяют однозначно определить поверхность. В качестве ограничений при интерполяции нормали к поверхности может применяться информация об освещенности. Эти данные могут помочь

различить яйцо и мяч, но не позволят отличить внешнюю часть мяча от внутренней.

### Упражнение 12.20

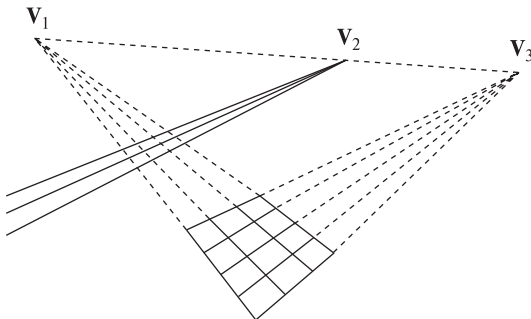
Найдите карикатуру или комикс, содержащий фигуры человека или животного с непрерывным контуром. (а) Имеются ли на изображении какие-либо тени, закраска или текстуры, помогающие восприятию трехмерной формы объекта? Если нет, добавьте несколько таких признаков самостоятельно, предполагая, что источник света расположен спереди в верхнем правом углу сцены. (б) Скопируйте контур объекта на чистый лист бумаги. Для представления формы объекта укажите направление нормалей примерно в 20 точках на его контуре. Эти значения должны соответствовать возможному содержанию внутреннего изображения объекта.

#### 12.4.2 Точки схода

При перспективной проекции происходит интересное искажение направления параллельных прямых. Этот факт используется художниками на протяжении столетий. На рис. 12.20 демонстрируются два хорошо известных явления. Во-первых, трехмерная прямая, наклоненная к оптической оси, будет казаться исчезающей в некоторой точке двумерного изображения. Эта точка называется *точкой схода*. Во-вторых, как показано на рис. 12.20, пучок параллельных прямых имеет общую точку схода. Эти явления легко описать с помощью математической модели перспективной проекции.

Точки схода различных пучков прямых, параллельных одной и той же плоскости, лежат на *линии горизонта*. На рис. 12.20 точки  $V_1$  и  $V_3$  определяют горизонт земной поверхности, в качестве которой рассматривается наклонная плоскость. Обратите внимание, что другой пучок из трех параллельных прямых (может быть, изображающих дорогу) сходится в точке  $V_2$ , которая тоже принадлежит линии горизонта.

Учет свойств перспективной проекции позволяет оценить параметры модели камеры по изображениям, полученным с помощью некалиброванной



**Рис. 12.20.** При перспективной проекции прямые, расположенные под углом к оптической оси, кажутся исчезающими в некоторой точке двумерного изображения. Параллельные прямые кажутся пересекающимися в *точке схода*.

камеры. В последнее время на основе свойств перспективной проекции были разработаны системы для построения трехмерных моделей сцен на основе видеосъемок, выполненных с различных точек наблюдения.

### 12.4.3 Определение глубины с помощью фокусировки

Для вычисления глубины поверхностного элемента, изображаемого в некотором пикселе, может применяться одна изолированная камера. Используемый прием напоминает действие человеческого глаза. Для фокусирования на объект интереса мускулы глаза изменяют форму хрусталика, т. е. его фокусное расстояние. Выполняя фокусировку на объект или на его края, можно оценить глубину этого объекта. На основе этого принципа был разработан ряд устройств, в том числе устройства с автоматическим управлением фокусировкой объектива камеры.

Для краткости изложения будем предполагать, что фокусное расстояние объектива может меняться в некотором диапазоне с известным шагом. Для всех возможных значений фокусного расстояния  $f$  формируются изображения. К ним применяется детектор краев. Для каждого пиксела запоминается такое значение  $f$ , которое соответствует самому резкому краю. Затем это значение используется для определения глубины точки трехмерной поверхности, изображаемой в данном пикселе. Далеко не все точки изображения соответствуют контрастным окрестностям трехмерных поверхностей, и в этих точках не будет получено значений резкости края, пригодных для вычислений глубины.

Короткофокусные объективы (например, с  $f < 8$  мм) часто характеризуются как объективы с хорошей *глубиной поля зрения*. Это означает, что объект остается в фокусе в широком диапазоне расстояний от камеры. Поэтому для точного определения глубины объектов с помощью фокусировки короткофокусные объективы подходят хуже, чем длиннофокусные. В разделе 12.7 этот вопрос будет рассмотрен на основе известного в физике уравнения тонкой линзы.

**Учет теневых областей.** Выше отмечалось, что для формирования признаков на однородных поверхностях может использоваться структурированное освещение. Похожим образом могут использоваться и теневые области. Например, люди и системы машинного зрения могут определять наличие поверхности и оценивать ее форму с помощью какого-либо шаблонного образа на этой поверхности. Рассмотрим правый нижний фрагмент на рис. 12.17. Шаблон, образованный кривыми внизу, позволяет предположить в качестве трехмерной интерпретации холмистую поверхность. Похожая ситуация показана на рис. 12.14, где шаблон спроектированных световых полос выявляет эллипсоидальную форму некоторых картофелин. Лыжники могут учитывать тени от деревьев на заснеженной местности для оценки угла горного склона, так как потеря равновесия возможна при ошибке даже в 15 см.

#### 12.4.4 Признаки, связанные с движением

Параллактическое смещение уже обсуждалось ранее. Когда движущийся зрительный датчик следит за объектом в трехмерном пространстве, то при приближении датчика к объекту на двумерном изображении точки этого объекта кажутся «расходящимися в стороны». (Если бы объект удалялся от датчика, то точки казались бы «сходящимися».)

Точка на изображении, которая кажется центром расхождения точек объекта, называется *фокусом расширения*. При приближении камеры к неподвижному объекту эта точка соответствует направлению движения камеры. Аналогичный эффект наблюдается при приближении объекта к неподвижной камере. В гл. 9 эти явления рассматривались применительно к вычислению оптического потока. Известны количественные методы описания взаимосвязи потока изображения с расстоянием и скоростью объектов или наблюдателя.

#### 12.4.5 Контуры и виртуальные прямые

Как показано на рис. 12.17, контуры и кривые могут быть *виртуальными* (воображаемыми). На левом верхнем фрагменте виртуальные кривые на изображении определяются верхними концами столбов изгороди, верхушками деревьев и линиями разметки на дороге. Справа вверху на рис. 12.17 приведены два известных примера, демонстрирующие некоторые особенности человеческого зрительного восприятия. При рассмотрении этого рисунка люди часто считают, что яркая треугольная область закрывает темные окружности, а яркая круговая область закрывает множество лучей, расходящихся из центральной точки. Известна гипотеза, согласно которой в случае, если зрительная система человека воспринимает (некорректно) наличие перекрывающихся друг друга объектов, то при этом отбрасывается интерпретация, по которой отражающие свойства объекта и фона считаются совпадающими. Эта особенность восприятия настолько сильна, что на верхнем правом фрагменте рис. 12.17 человек легко видит яркую треугольную и круговую область даже при отсутствии пунктирных виртуальных кривых. Системы компьютерного зрения не подвержены подобным иллюзиям, когда центральная область воспринимается более яркой, чем фон, так как компьютер непосредственно обрабатывает измеренные значения интенсивности пикселей.

**102 Определение. Виртуальные прямые** или кривые образуются в процессе зрительного восприятия человеком вследствие бессознательной группировки похожих точек или объектов, расположенных вдоль воображаемой прямой или кривой на изображении.

---

#### Упражнение 12.21

Аккуратно нарисуйте на двух белых карточках иллюзорные фигуры, показанные в верхнем правом фрагменте на рис. 12.17. Покажите каждую фигуру пяти

разным людям и выясните, считают ли они центральную область более яркой, чем фон. Не спрашивайте их об этом напрямую. Вместо этого задайте им общие вопросы относительно наблюдаемых фигур. Например, спросите: *Какие объекты вы видите? Пожалуйста, опишите их форму и цвет.* Проанализируйте полученные результаты.

### 12.4.6 Совмещенность

Люди стремятся отбрасывать интерпретации, которые подразумевают *случайные совмещения* объектов в пространстве или объектов и наблюдателя. Вместо этого обычно предполагается, что совмещенность образов на двумерном изображении связана с некоторыми пространственными свойствами объектов. Например, в верхней части рис. 12.17(с) человек увидит две совмещенные четырехугольные области и скорее всего будет считать, что две (прямоугольные) трехмерные грани соединяются в пространстве и образуют ребро на изображении. Это ребро из-за эффекта ракурса воспринимается как сгиб. Внизу на рис. 12.17 (с) показан вид этих же граней после небольшого изменения точки наблюдения. Стыки-вилки и стыки-стрелки из верхней части рисунка преобразуются в Т-стыки, и теперь эти образы воспринимаются как две перекрывающиеся поверхности.

Восприятия виртуальных кривых и совмещенных объектов являются двумя разными проявлениями одних и тех же физиологических особенностей зрительной системы человека. Все четыре фрагмента рис. 12.17 можно объяснить на основе одного и того же принципа. В выпшедшей в 1983 г. монографии Ирвинга Рока (Irving Rock) высказывается предположение о том, что зрительная система человека при интерпретации изображения выбирает простейшую гипотезу. Это предположение объясняет многие эксперименты и помогает установить причины принятия зрительных решений. Однако также известны ряд экспериментальных фактов, которые противоречат данному предположению.

Ниже приведены несколько эвристических правил, которые были предложены для интерпретации изображений. Ни одно из этих правил не обеспечивает корректных интерпретаций во всех случаях: для каждого из них легко найти опровергающие примеры. В описании правил термин *край* используется не только для обозначения признаков двумерных изображений, но и для обозначения сгиба, знака или области тени в трехмерном пространстве.

- Прямому краю на изображении часто соответствует прямой край в трехмерном пространстве.
- Ребрам, образующим стык на двумерном изображении, часто соответствуют края, образующие угол в пространстве.
- Похожие образы, лежащие вдоль кривой на изображении, часто соответствуют похожим объектам, расположенным вдоль кривой в пространстве.
- Полигональная область на изображении часто соответствует полигональной грани в пространстве.

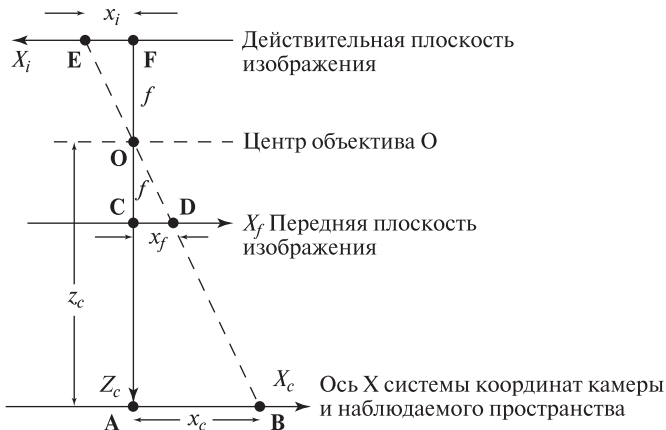
- Гладкий криволинейный контур на двумерном изображении часто соответствует гладкому объекту в пространстве.
- Симметричные области на изображении часто соответствуют симметричным объектам в пространстве.

## 12.5 МОДЕЛЬ ФОРМИРОВАНИЯ ИЗОБРАЖЕНИЙ НА ОСНОВЕ ПЕРСПЕКТИВНОЙ ПРОЕКЦИИ

В этом разделе обсуждается математическая модель формирования изображений на основе перспективной проекции. Эту модель особенно легко построить для описания взаимосвязи точек в системе координат камеры  $\mathbf{C}$  с точками в действительной системе координат изображения  $\mathbf{R}$ .

Начнем рассмотрение с простого одномерного случая, представленного на рис. 12.21. Тем не менее эта одномерная модель применима к ряду реальных задач, например, для описания получения изображений плоской земной поверхности с самолета с помощью камеры, оптическая ось которой направлена вертикально вниз. Наблюдаемая точка  $\mathbf{B}$  проектируется в точку изображения  $\mathbf{E}$ . Начало системы координат камеры расположено в точке  $\mathbf{O}$ . Расстояние между точками  $\mathbf{O}$  и  $\mathbf{B}$ , измеренное вдоль оптической оси, равно расстоянию  $z_c$  между точками  $\mathbf{O}$  и  $\mathbf{A}$ . Образ точки  $\mathbf{B}$  смещен от центра изображения на расстояние  $x_i$  вдоль оси  $X_i$ . Фокусное расстояние объектива равно  $f$ . Учитывая подобие треугольников, запишем уравнение 12.1. Из него следует, что действительная координата точки на двумерном изображении (размер объекта) равна пространственной координате точки (размеру одномерного объекта), взятой в масштабе, равном отношению фокусного расстояния к расстоянию до объекта.

$$x_i/f = x_c/z_c \quad \text{или} \quad x_i = (f/z_c) x_c \quad (12.1)$$



**Рис. 12.21.** Простая модель перспективной проекции с указанием действительной и передней плоскости изображения. Пространственный размер объекта  $x_c$  связан с размером его образа  $x_i$  посредством соотношений подобных треугольников:  $x_i/f = x_c/z_c$ .

Если вычислить координаты образов всех точек, лежащих на одинаковом расстоянии от камеры, то будет получено изображение, представляющее собой уменьшенную копию двумерного плоского объекта. Эта модель применима к таким прикладным задачам, как анализ микроскопических изображений, аэрофотосъемок, или обработка изображений отсканированных документов.

Процесс формирования изображения удобнее рассматривать на *передней плоскости изображения*, а не на действительной плоскости изображения, которая по отношению к наблюдаемому объекту расположена позади объектива. Изображение на действительной плоскости перевернуто относительно наблюдаемого объекта, а на передней плоскости имеет ту же ориентацию, что и объект. Передняя плоскость изображения — это воображаемая плоскость, расположенная на расстоянии  $f$  от оптического центра в направлении наблюдаемой сцены. Образы объектов на передней и действительной плоскостях изображения имеют одинаковые размеры и отличаются только направлением. На рис. 12.21 точки **C** и **D** на передней плоскости изображения соответствуют точкам **F** и **E** на действительной плоскости изображения. Уравнение перспективной проекции справедливо для точек на обеих плоскостях изображений. Начиная с этого момента и далее, построение изображений будет рассматриваться на передней плоскости.

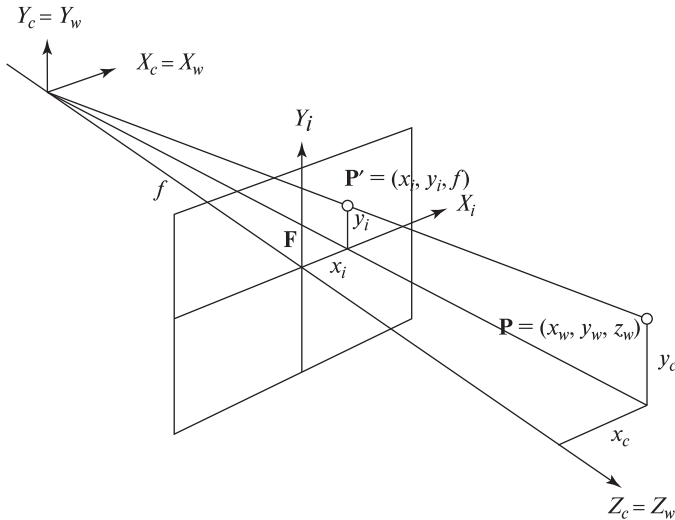
Схема перспективной проекции трехмерной сцены на двумерную плоскость приведена на рис. 12.22. Соответствующая математическая модель представлена в виде уравнений 12.2. Эти уравнения для координатных направлений  $x$  и  $y$  были получены с использованием тех же соотношений подобных треугольников, что и в одномерном случае. Проекционное отображение трехмерного пространства на плоскость не является однозначным. Все трехмерные точки, лежащие вдоль одного проектирующего луча, будут проектироваться в одну и ту же точку двумерного изображения. Поэтому на двумерном изображении теряется существенный объем пространственной информации.

$$\begin{aligned}x_i/f &= x_c/z_c \quad \text{или} \quad x_i = (f/z_c) x_c \\y_i/f &= y_c/z_c \quad \text{или} \quad y_i = (f/z_c) y_c\end{aligned}\tag{12.2}$$

Математическая модель в виде уравнений 12.2 может использоваться в компьютерных алгоритмах для определения множества всех трехмерных точек на проектирующем луче, проходящем через точку изображения  $(x_i, y_i)$ . Эти уравнения имеют фундаментальное значение для всех трехмерных задач, рассматриваемых в данной книге.

В заключение данного раздела еще раз отметим, что простые уравнения (12.2) описывают взаимосвязь точек в трехмерной системе координат камеры с точками на двумерной действительной системе координат изображения. Для записи уравнений перспективной проекции в системе координат объекта или в мировой системе координат требуется применять преобразования систем координат, которые будут рассматриваться в гл. 13.





**Рис. 12.22.** Модель перспективной проекции для получения двумерного изображения.

В случае, если с помощью камеры наблюдается плоский объект, расположенный на постоянном расстоянии  $z_c = c_1$ , то изображение будет выглядеть как копия этой плоскости в уменьшенном масштабе. Положив  $c_2 = f/c_1$ , из уравнений (12.2) получим простые соотношения  $x_i = c_2 x_c$  и  $y_i = c_2 y_c$ . Следовательно, при такой организации сцены можно работать непосредственно в системе координат изображения, считая изображение масштабированной копией плоского пространственного объекта.

---

**Упражнение 12.22.** Свойства изображения  
с однородным масштабированием

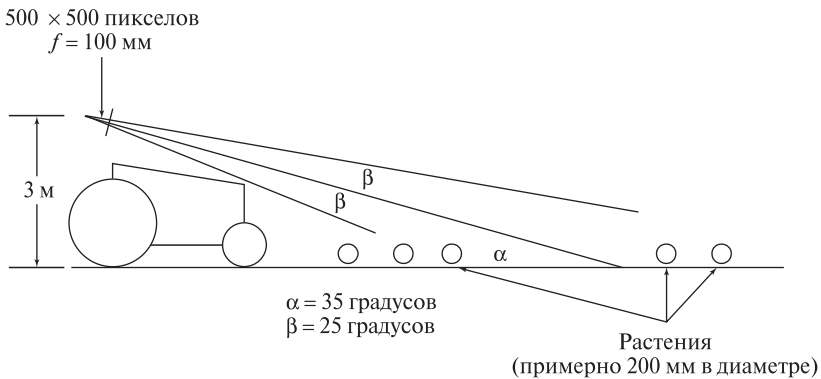
Предположим, что камера направлена вертикально вниз для наблюдения стола так, что плоскость изображения параллельна верхней поверхности стола (напоминает фотоувеличитель). Схема расположения камеры и объекта соответствует рис. 12.21. Покажите, что длина образа гвоздя размером 2,5 см постоянна и не зависит от положения гвоздя на столе в пределах поля зрения камеры.

---

**Упражнение 12.23.** Зрительная система  
для автоматического управления трактором

Допустим, что в автоматической системе для рулевого управления сельскохозяйственным трактором, используемым для прополки сорняков и внесения

удобрений, используется информация от зрительной системы. Камера направлена вперед по ходу движения трактора (рис. 12.23). Фокусное расстояние объектива равно 100 мм. Камера расположена на высоте 3000 мм над землей. Угловое поле зрения составляет 50 градусов, а угол между оптической осью и земной поверхностью равен 35 градусам.

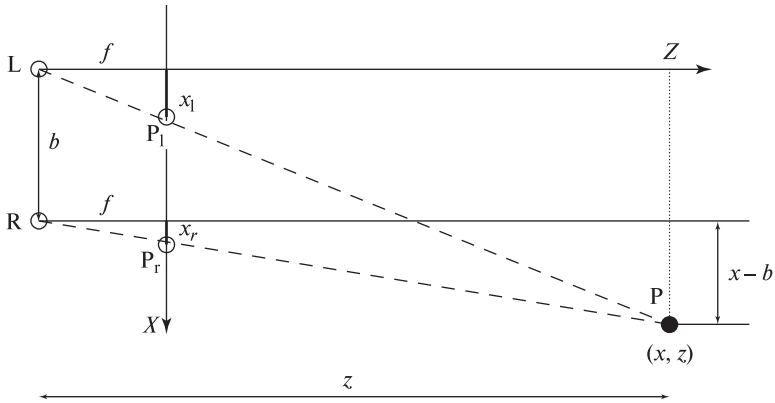


**Рис. 12.23.** Схема зрительной системы для управления сельскохозяйственным трактором. См. Упражнение 12.23.

(а) Чему равна глубина поля зрения вдоль земной поверхности? (б) Предположим, что растения посажены с промежутками 500 мм. Какие промежутки будут на изображении между растениями, попадающими на края поля зрения? (Промежутки на изображении различны для близких и далеких растений.) (в) Допустим, что растения похожи на сферы диаметром 200 мм, а размеры изображения равны 500 × 500 пикселей. Оцените пиксельный диаметр образов растений на изображении. (Диаметр будет иметь различные значения для близких и далеких растений.) (г) Будут ли образы соседних растений перекрываться на изображении или между ними останутся промежутки?

## 12.6 ОПРЕДЕЛЕНИЕ ГЛУБИНЫ С ПОМОЩЬЮ СТЕРЕОСКОПИЧЕСКОЙ СИСТЕМЫ

Принцип определения пространственного расположения точек с помощью стереоскопической системы можно объяснить с помощью несложных геометрических построений и алгебраических выражений. Схема стереоскопической зрительной системы показана на рис. 12.24. Будем полагать, что две камеры установлены так, что их оси  $X$  коллинеарны, а оси  $Y$  и  $Z$  параллельны. Ось  $Y$  перпендикулярна плоскости рисунка. Начало координат — или центр проектирования — правой камеры смещен относительно центра левой камеры вдоль оси  $X$  на величину  $b$ , которая называется *базой* (baseline) стереоскопической системы. При наблюдении некоторой точки объекта  $P$  на левом изображении



**Рис. 12.24.** Геометрическая модель простой стереоскопической системы. В качестве системы координат стереосистемы выбрана система координат левой камеры **L**. База стереосистемы  $b$  известна. Все координаты измеряются относительно **L**, за исключением  $x_r$ , которая измеряется относительно системы координат правой камеры **R**.

формируется точка **P**<sub>1</sub>, а на правом — точка **P**<sub>r</sub>. Из геометрических построений очевидно, что точка **P** лежит на пересечении лучей **LP**<sub>1</sub> и **RP**<sub>r</sub>.

Учитывая подобие треугольников на рис. 12.24, получим уравнения (12.3):

$$z/f = x/x_l \quad (12.3)$$

$$z/f = (x - b)/x_r$$

$$z/f = y/y_l = y/y_r$$

Отметим, что по построению координаты точек изображений  $y_l$  и  $y_r$  могут считаться одинаковыми. После нескольких подстановок и преобразований выразим из уравнений (12.3) две неизвестные координаты  $x$  и  $z$  точки **P**.

$$z = fb/(x_l - x_r) = fb/d \quad (12.4)$$

$$x = x_l z/f = b + x_r z/f$$

$$y = y_l z/f = y_r z/f$$

В выражение глубины точки **P** в уравнения (12.4) была введена величина *рассогласования*  $d$ . Она равна разности координат  $x_l$  и  $x_r$  точек на левом и правом изображениях. Решение уравнений (12.4) дает все три координаты, полностью определяющие положение точки **P** в трехмерном пространстве. Из уравнений (12.4) очевидно, что расстояние до точки **P** увеличивается при уменьшении рассогласования и уменьшается при увеличении рассогласования. Расстояние стремится к бесконечности при приближении рассогласования к нулю. По

построению простой стереоскопической системы на рис. 12.24 рассогласование двух координат  $y$  точек на двух изображениях отсутствует.

**103 Определение.** Рассогласование (disparity) характеризует различие местоположения образов одной и той же трехмерной точки при перспективном проектировании на изображениях двух различных камер.

На рис. 12.24 показана одиночная точка  $P$ , расположенная в трехмерном пространстве так, что не возникает проблемы обнаружения соответствующих точек изображений  $P_1$  и  $P_r$ . При наблюдении реальных трехмерных сцен поиск соответствующих точек часто оказывается весьма сложной задачей, так как среди большого количества точек поверхностей может быть неясно, какие точки левого и правого изображений являются образами одной и той же точки трехмерной поверхности.

Предположим, что имеется стереопара изображений кукурузного поля, похожих на рис. 12.12. На таких изображениях вдоль строк изображений могло бы оказаться много похожих краевых точек. Камеры в стереоскопических системах часто очень точно выравниваются так, чтобы ограничить область поиска соответствующих точек двух изображений в пределах строк с одинаковыми индексами.

Хотя для стереосистем известно и используется много различных ограничений, все еще остаются нерешенные проблемы. Одна из проблем возникает в случае, когда точка  $P$  видна только на одном изображении. В примере с изображениями кукурузного поля насыщенная текстура приводила к получению очень большого количества признаков точек. Противоположная по смыслу проблема часто возникает при наличии слишком малого количества признаков точек. Эта проблема характерна для сцен с гладкими объектами без текстуры (например, таких, как мраморная статуя или покрытый снегом холм). В промышленных условиях с помощью специального освещения иногда удается привнести в сцену искусственные признаки точек (см. рис. 12.14). Более подробно эти вопросы будут рассматриваться ниже.

Несмотря на перечисленные трудности, исследования в области стереоскопических систем привели к разработке нескольких коммерчески доступных стереосистем. В некоторых из них используется больше двух камер. Ряд систем способны генерировать дальнометрические изображения с частотой, близкой к кадровой частоте видеокамер. В гл. 16 обсуждается использование стереоскопической системы в задаче идентификации пользователей банкомата.

---

### Упражнение 12.24

Выполните следующий эксперимент, демонстрирующий свойства стереоскопического зрения. (а) Прямо перед собой на расстоянии примерно 30 см от носа расположите книгу. Направление наблюдения должно быть перпендикулярно обложке книги. Посмотрите на книгу каждым глазом по очереди в течение двух секунд. Заметно ли рассогласование признаков точек (например, букв

названия) на левом и правом изображениях? (б) Повторите эксперимент, удерживая книгу на расстоянии вытянутой руки. Рассогласования увеличиваются или уменьшаются? (в) Существенно поверните книгу. Можете ли вы найти такую ориентацию книги, чтобы переплет был виден только правым глазом?

---

**Упражнение 12.25.** Ошибка вычислений глубины в стереоскопической системе

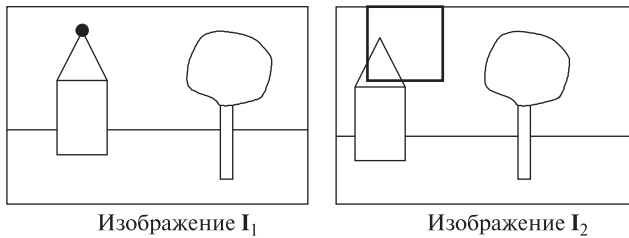
Предположим, что с помощью стереосистемы с базой  $b = 10$  см и камерами с фокусными расстояниями  $f = 2$  см наблюдается точка  $\mathbf{P} = (10 \text{ см}, 1000 \text{ см})$ . Точка  $\mathbf{P}$  лежит на оптической оси правой камеры (см. рис. 12.24). Допустим, что вследствие различных ошибок координата на изображении  $x_l$  на 1 процент меньше истинного значения, а координата на правом изображении  $x_r$  определена точно. Какая ошибка глубины (в сантиметрах) будет получена при вычислениях с использованием уравнений (12.4)?

**Стереои изображения.** Стереои изображения применяются в компьютерных графических системах для представления трехмерных сцен пользователю в интерактивном режиме. Эта задача представления трехмерных сцен в компьютерной графике является обратной к задаче компьютерного зрения: известны все точки трехмерной поверхности  $(x, y, z)$  и графическая система должна сформировать левое и правое изображения. Переписывая уравнения (12.4), получим уравнения (12.5), которые дают формулы для вычисления координат образов  $(x_l, y_l)$  и  $(x_r, y_r)$  для заданной точки объекта  $(x, y, z)$  при известных значениях базы  $b$  и фокусного расстояния  $f$ . Таким образом, на основе компьютерной модели объекта графическая система генерирует два отдельных изображения. Эти изображения представляются пользователю одним из двух способов: (а) с помощью специального шлема со встроенными дисплеями одно изображение демонстрируется левому глазу, а другое — правому; или (б) на экране монитора два изображения показываются по очереди с использованием дополнительных цветов, а пользователь наблюдает эти изображения через специальные очки с различными цветными фильтрами на каждом глазу. Известен недорогой третий метод представления статических стереоскопических изображений. Люди способны воспринимать стереопару изображений, которые напечатаны рядом друг с другом на листе обыкновенной бумаги. (Например, пристально посмотрите на стереопару на рис. 12.25. Эти изображения заимствованы из работы [Tanimoto, 1998], приведенной в списке литературы.)

$$x_l = xf/z \tag{12.5}$$

$$x_r = f(x - b)/z$$

$$y_l = y_r = yf/z$$



**Рис. 12.25.** Кросс-корреляционный метод для обнаружения соответствующих признаков на изображениях стереопары.

В гл. 15 более подробно обсуждаются способы использования стереоизображений в системах виртуальной реальности, которые формируют у пользователя реалистичное представление о трехмерных сценах. Например, они могут применяться рентгенологами для изучения структуры трехмерных объемных данных, полученных на ЯМР-томографе.

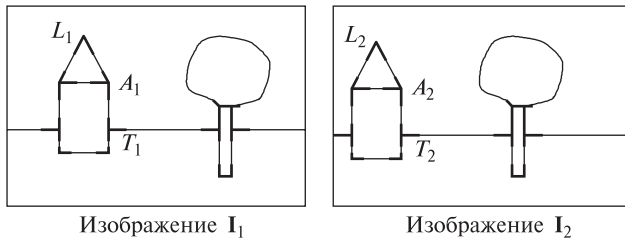
### 12.6.1 Обнаружение соответствующих признаков

Наиболее сложные операции обработки в стереоскопических зрительных системах связаны не с вычислениями глубины, а с определением соответствующих признаков, которые необходимы для этих вычислений. Некорректно установленные соответствия приводят к ошибкам в значениях глубины. Ошибки могут приводить как к незначительным отклонениям от истинного значения, так и к полностью неверным результатам. В данном разделе обсуждаются основные методы обнаружения соответствующих признаков и некоторые применяемые при этом ограничения.

**Кросс-корреляция.** Один из самых старых методов обнаружения соответствия между пикселями двух изображений основан на использовании оператора кросс-корреляции. Этот оператор был описан в гл. 5. В кросс-корреляционном методе предполагается, что для заданной точки  $P_1$  изображения  $I_1$  (на первом изображении стереопары) на изображении  $I_2$  (на втором изображении стереопары) должна существовать такая область, в которой может быть найдена точка  $P_2$ , соответствующая точке  $P_1$ .

Размер области поиска соответствующей точки определяется на основе информации о расположении камеры. Эта информация используется при получении изображений. В задачах промышленного компьютерного зрения информацию о расположении камеры часто можно извлечь из параметров камеры, которые определяются на этапе калибровки (см. гл. 13). В задачах удаленного зондирования и других задачах обработки изображений земной поверхности размер области поиска можно оценить на основе эталонных изображений.

В любом случае, для пиксела  $P_1$  изображения  $I_1$  выполняется поиск в пределах ограниченной области  $I_2$ . В процессе поиска к окрестностям пикселей



**Рис. 12.26.** Обнаружение соответствующих признаков на изображениях стереопары методом символического сопоставления. В качестве кандидатов на соответствующие признаки выбраны L-стыки и стыки-стрелки (A). T-стыки обычно игнорируются, так как они часто образуются вследствие перекрытия поверхностей, а не из-за пространственных свойств объектов.

$P_1$  и  $P_2$  применяется оператор кросс-корреляции. Пиксел изображения  $I_2$ , для которого достигается максимальный отклик оператора кросс-корреляции, считается наилучшим вариантом сопоставления для  $P_1$  и используется для вычисления глубины соответствующей пространственной точки.

Кросс-корреляционный метод был весьма успешно использован для обнаружения соответствующих признаков на спутниковых и авиационных изображениях. Этот метод поясняется на рис. 12.25. Черным кружком на изображении  $I_1$  обозначена точка, для которой требуется найти соответствующую точку. Поиск этой точки выполняется в пределах квадратной области изображения  $I_2$ .

**Символическое сопоставление и реляционные ограничения.** Второй общепринятый метод обнаружения соответствий предполагает поиск характерного признака на одном изображении, который сопоставляется с характерным признаком на другом изображении. В качестве характерных признаков часто выбираются стыки, отрезки и области.

Сопоставление данного типа можно описать в терминах задачи согласованной маркировки, которая описывалась в гл. 11. В качестве множества частей  $P$  рассматривается множество признаков первого изображения  $I_1$ , а в качестве множества меток  $L$  — множество признаков второго изображения  $I_2$ . При использовании признаков нескольких типов необходимо следить, чтобы часть и ее метка были однотипными. (Отметим, что T-стыки используются редко, так как они обычно возникают при перекрытии ребер, а не из-за пространственных особенностей трехмерных объектов.) Пространственные отношения  $R_P$ , которые выполняются на множестве  $P$ , должны совпадать с пространственными отношениями  $R_L$ , выполняемыми на множестве  $L$ .

Допустим, что в качестве сопоставляемых признаков выбраны стыки (см. рис. 12.26). Соответствующие стыки должны быть однотипными (например, L-стык должен отображаться на другой L-стык). Если два стыка соединены отрезком на первом изображении (например, L и A), то соответствующие стыки на втором изображении также должны соединяться отрезком.

Если сопоставляемые признаки являются отрезками, то при сопоставлении могут использоваться отношения параллельности и коллинеарности. При сопоставлении областей может применяться отношение смежности.

При сопоставлении признаков возможны проблемы, которые возникают и в других методах сопоставления стереоизображений. Не каждый признак удастся обнаружить на двух изображениях. На одном из них некоторые признаки могут отсутствовать из-за смещения точки наблюдения. Ряд признаков видны на одном изображении, но перекрываются какими-либо поверхностями на другом. Некоторые признаки могут быть обнаружены неверно или пропущены. Также могут быть обнаружены посторонние признаки.

В процедурах символического сопоставления обычно не выполняется поиск наилучшей согласованной маркировки, а вместо этого применяется какая-либо приближенная версия сопоставления. В таком случае целесообразно выполнять поиск отображения по критерию минимизации ошибки или применять непрерывную релаксацию для получения приближенного ответа.

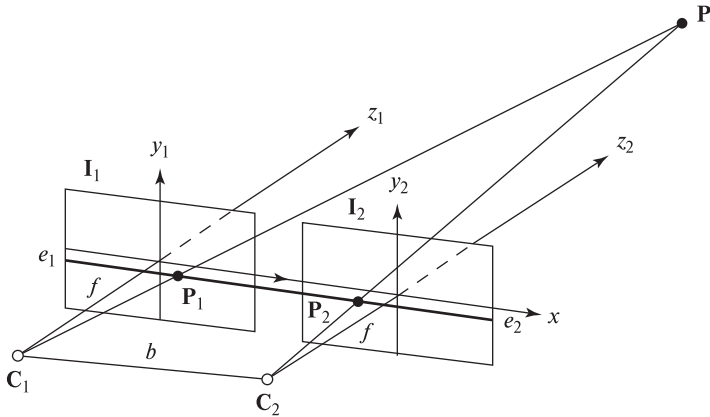
После того как для признаков первого и второго изображения было определено сопоставляющее отображение, обработка не заканчивается. Соответствие стыков позволяет получить разреженную дальнометрическую карту, в которой значения глубины определены только для небольшого множества точек. Сопоставление отрезков может привести к установлению соответствия между их концевыми или средними точками. При сопоставлении областей требуются дополнительные усилия для выбора внутри областей точек, которые будут считаться соответствующими. Полученные разреженные карты глубины далее могут быть дополнены посредством линейной интерполяции между известными значениями.

При использовании метода символического сопоставления ошибки могут возникать на многих этапах обработки. Вероятно, поэтому на практике продолжает широко применяться кросс-корреляционный метод, особенно применительно к изображениям естественных сцен.

**Эпиполярное ограничение.** Сопоставление стереоскопических изображений может быть упрощено, если известна относительная ориентация камер. В таком случае двумерное пространство поиска точки изображения, которая соответствует заданной точке другого изображения, сокращается до одномерного пространства. Для этого учитываются свойства *эпиполярной конфигурации* (epipolar geometry) пары изображений. На рис. 12.27 показана эпиполярная конфигурация для простого случая, когда две одинаковые параллельные плоскости изображений параллельны базе стереосистемы. Тогда известно, что для заданной точки  $P_1 = (x_1, y_1)$  изображения  $I_1$  соответствующая точка  $P_2 = (x_2, y_2)$  на изображении  $I_2$  лежит на той же строке изображения, т. е.  $y_1 = y_2$ . Будем называть такую конфигурацию *нормальной парой изображений* (normal image pair).

Геометрические свойства нормальной пары изображений особенно просты, но не всегда удается расположить камеры таким образом. Рассогласования





**Рис. 12.27.** Эпиполярная конфигурация нормальной пары изображений. Точка трехмерного пространства  $P$  проецируется в точку  $P_1$  на изображении  $I_1$  и в точку  $P_2$  на изображении  $I_2$ . Плоскости этих изображений параллельны базе стереосистемы. Оптические оси перпендикулярны базе и параллельны друг другу.

точек на нормальной паре могут быть недостаточно большими для точного вычисления значений глубины. Конфигурация стереосистемы общего вида допускает произвольное расположение и ориентацию камер. В поле зрения каждой камеры должна попадать значительная часть наблюдаемой сцены. Эпиполярная конфигурация стереоскопической системы общего вида показана на рис. 12.28.

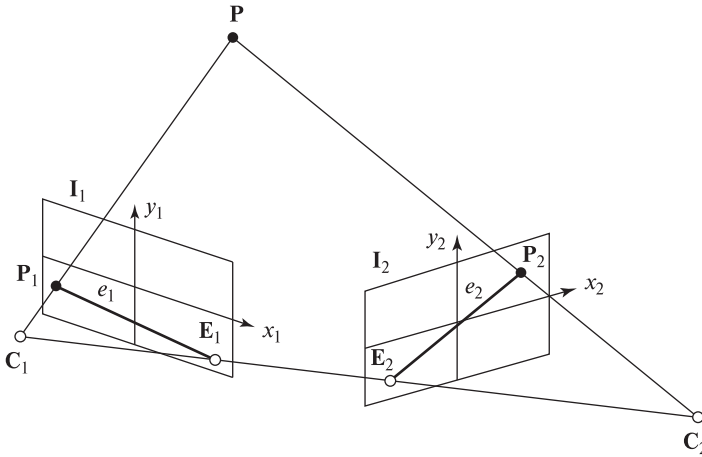
**104 Определение.** Эпиполярной плоскостью (epipolar plane) называется плоскость, которой принадлежит трехмерная точка  $P$ , оптические центры двух камер  $C_1$  и  $C_2$ , и две точки изображений  $P_1$  и  $P_2$ , являющиеся проекционными образами точки  $P$ .

**105 Определение.** Эпиполярными прямыми (epipolar line) называются две прямые  $e_1$  и  $e_2$ , которые являются прямыми пересечения эпиполярной плоскости с двумя плоскостями изображений  $I_1$  и  $I_2$ .

Если на изображении  $I_1$  задана точка  $P_1$ , принадлежащая эпиполярной прямой  $e_1$ , и известно относительное расположение камер (см. гл. 13), то на изображении  $I_2$  можно найти соответствующую эпиполярную прямую  $e_2$ , на которой должна лежать соответствующая точка  $P_2$ . Если некоторая другая точка  $P'_1$  изображения  $I_1$  не принадлежит эпиполярной плоскости точки  $P_1$ , то эта точка  $P'_1$  должна лежать на эпиполярной прямой, отличной от  $e_1$ .

**106 Определение.** Эпиполюм (epipole) изображения стереопары называется точка, в которой пересекаются все эпиполярные прямые этого изображения.

На рис. 12.28 эпиполюми изображений  $I_1$  и  $I_2$  являются точки  $E_1$  и  $E_2$ , соответственно.



**Рис. 12.28.** Эпиполярная конфигурация пары изображений общего вида. Точка трехмерного пространства  $P$  проецируется в точку  $P_1$  на изображении  $I_1$  и в точку  $P_2$  на изображении  $I_2$ . Плоскости изображений не параллельны. Точка  $P_1$  изображения  $I_1$  лежит на эпиполярной прямой  $e_1$ . Прямая  $e_2$  — это соответствующая эпиполярная прямая, которой принадлежит точка  $P_2$  изображения  $I_2$ . Точка  $E_1$  является эпиполем изображения  $I_1$ , а точка  $E_2$  — эпиполем изображения  $I_2$ .

**Ограничение упорядочения.** Предположим, что задана пара точек сцены и их соответствующие проекции на каждом изображении стереопары. Ограничение упорядочения утверждает, что если эти точки лежат на непрерывной трехмерной поверхности, то их образы должны быть упорядочены одинаковым образом вдоль эпиполярных прямых на каждом изображении стереопары. Это ограничение по сравнению с эпиполярным ограничением имеет более эвристический характер, так как на этапе сопоставления точно неизвестно, являются ли две точки изображения образами точек одной и той же трехмерной поверхности. Следовательно, ограничение упорядочения может применяться при поиске потенциальных соответствующих точек, но без подтверждающих проверок может приводить к ошибкам сопоставления.

**Связь ошибок определения глубины с зоной обзора.** В процессе разработки стереоскопической системы приходится искать компромисс между размером зоны обзора и точностью вычислений глубины. На малой базе небольшие ошибки определения местоположения точек изображений  $P_1$  и  $P_2$  будут приводить к большим ошибкам вычисления глубины трехмерной точки  $P$  (это видно из вышеприведенных рисунков). Увеличение базы позволяет повысить точность. Однако при удалении камер друг от друга возрастает вероятность сбоя обнаружения соответствующих точек из-за эффектов перекрытия образов объектов на изображениях. В ряде работ предлагается выбирать между оптическими осями угол  $\pi/4$ , при котором достигается разумный компромисс между точностью вычислений глубины и размером зоны обзора.

## 12.7 ФОРМУЛА ТОНКОЙ ЛИНЗЫ\*

Построение изображения, формируемого собирающей тонкой линзой, показано на рис. 12.29. Луч, проходящий через точку объекта  $\mathbf{P}$  параллельно оптической оси, после преломления линзой проходит через фокальную точку  $\mathbf{F}_i$  и точку изображения  $\mathbf{p}'$ . Другие лучи, проходящие через точку  $\mathbf{P}$ , после преломления тонкой линзой также будут проходить через  $\mathbf{p}'$ . Луч, проходящий через оптический центр  $\mathbf{O}$ , не преломляется. Луч, проходящий через точку изображения  $\mathbf{p}'$  параллельно оптической оси, проходит через линзу и вторую фокальную точку  $\mathbf{F}_j$ .

Формулу тонкой линзы можно записать на основе чертежа на рис. 12.29. Так как расстояние  $X$  равно расстоянию между точками  $\mathbf{R}$  и  $\mathbf{O}$ , то на основе подобия треугольников  $\mathbf{ROF}_i$  и  $\mathbf{Sp}'\mathbf{F}_i$  получим следующее равенство.

$$\frac{X}{f} = \frac{x'}{z'} \quad (12.6)$$

С учетом подобия треугольников  $\mathbf{POT}$  и  $\mathbf{p}'\mathbf{OS}$  получим еще одно равенство.

$$\frac{X}{f + Z} = \frac{x'}{f + z'} \quad (12.7)$$

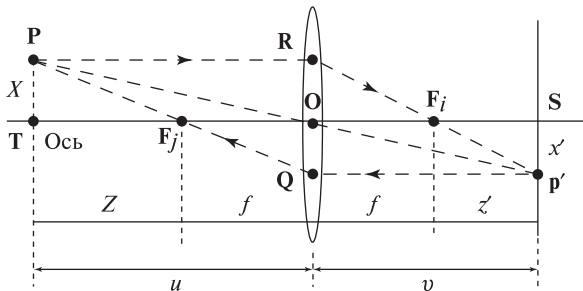
Подстановка  $X$  из выражения (12.6) в (12.7) дает

$$f^2 = Zz' \quad (12.8)$$

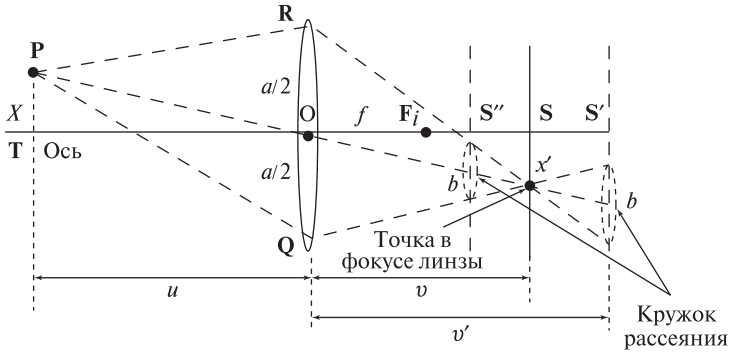
После подстановки  $u - f$  для  $Z$  и  $v - f$  для  $z'$  получим

$$uv = f(u + v) \quad (12.9)$$

Разделим обе части полученного выражения на  $(uvf)$ . В результате получим формулу тонкой линзы в наиболее известном виде. Эта формула описывает



**Рис. 12.29.** Построение изображения, формируемого собирающей тонкой линзой. Луч, проходящий через точку объекта  $\mathbf{P}$  параллельно оптической оси, после преломления линзой проходит через фокальную точку  $\mathbf{F}_i$  и точку изображения  $\mathbf{p}'$ . Луч, проходящий через точку изображения  $\mathbf{p}'$  параллельно оптической оси, проходит через линзу и фокальную точку  $\mathbf{F}_j$ .



**Рис. 12.30.** Точка **P** окажется вне фокуса линзы, если глубина точки или положение плоскости изображения будут нарушать равенство в формуле тонкой линзы. Если **S** является местоположением плоскости изображения, когда **P** находится в фокусе, то **P** окажется вне фокуса после смещения плоскости изображения в положение **S'** или **S''**. При этом образ **P** будет выглядеть как диск диаметром **b**.

взаимосвязь фокусного расстояния с расстоянием объекта  $u$  от центра линзы и расстоянием изображения  $v$  от центра линзы.

$$\frac{1}{f} = \frac{1}{u} + \frac{1}{v} \quad (12.10)$$

**Фокус и глубина поля зрения.** Предположим, что точка **P** расположена точно в фокусе (рис. 12.29). Если сместить плоскость изображения, как показано на рис. 12.30, то точка **P** окажется вне фокуса линзы. Равенство (12.10), выполнявшееся для значения  $v$ , теперь нарушается при новом значении расстояния  $v'$ . Аналогично, если плоскость изображения останется неподвижной, но сместится точка **P**, то изменится расстояние  $u$ , и равенство в формуле тонкой линзы снова не будет выполняться. В любом случае, вместо четкой точки на плоскости изображения образ пространственной точки будет размываться и выглядеть как диск диаметром  $b$  («кружок рассеяния»). Установим связь диаметра кружка рассеяния с разрешением камеры и глубиной ее поля зрения.

Будем считать допустимым значение  $b$  порядка одного пиксела. С учетом этого предположения, вычислим ближайшее и наиболее далекое расстояние точки **P** от линзы, в диапазоне между которыми диаметр кружка рассеяния для этой точки не превышает заданного предельного значения. При фиксированном расстоянии  $u$  в качестве глубины наблюдения объекта соответствующее идеальное расстояние между линзой и плоскостью изображения  $v$  определяется по формуле тонкой линзы (12.10). Будем считать известными апертуру линзы  $a$  и фокусное расстояние  $f$ . Схема для пояснения дальнейшего изложения приведена на рис. 12.30. Для расстояний  $u$  и  $v$  точка **P** располагается в фокусе

линзы идеальным образом. Выясним пределы изменения  $u$ , в которых диаметр кружка рассеяния не превышает предельного значения  $b$ .

Для предельных значений  $v'$  из подобия треугольников на рис. 12.30 запишем выражения

$$\begin{aligned} v' &= \frac{a+b}{a}v \quad \text{при } v' > v \\ v' &= \frac{a-b}{a}v \quad \text{при } v' < v \end{aligned} \quad (12.11)$$

Согласно формуле тонкой линзы, при  $v' > v$  расстояние  $u'$  по сравнению  $u$  будет соответствовать более близкому от камеры положению, а при  $v' < v$  — более далекому. Вычислим ближайшую точку  $u_n$ , для которой достигается диаметр кружка рассеяния  $b$  (см. рис. 12.30). Для этого несколько раз применим формулу тонкой линзы для описания взаимосвязи параметров  $u, v, f$  и уравнение (12.11) при  $v' > v$ .

$$\begin{aligned} u_n &= \frac{fv'}{v' - f} = \frac{f \frac{(a+b)v}{a}}{\frac{(a+b)v}{a} - f} \\ &= \frac{f \frac{(a+b)}{a} \frac{uf}{(u-f)}}{\frac{(a+b)}{a} \frac{uf}{(u-f)} - f} \\ &= \frac{uf(a+b)}{af + bu} = \frac{u(a+b)}{a + \frac{bu}{f}} \end{aligned} \quad (12.12)$$

Аналогично, повторяя эти вычисления при  $v' < v$ , определим положение  $u_r$  наиболее далекой допустимой плоскости расположения объекта.

$$u_r = \frac{uf(a-b)}{af - bu} = \frac{u(a-b)}{a - \frac{bu}{f}} \quad (12.13)$$

**107 Определение.** Глубина поля зрения (depth of field) равна расстоянию между дальней и ближней плоскостями расположения объекта при заданных параметрах изображения и предельном диаметре кружка рассеяния  $b$ .

В большинстве ситуаций  $u > f$ , тогда из последнего выражения в уравнении (12.12) следует, что  $u_n < u$ . Кроме того, если зафиксировать все величины, кроме фокусного расстояния  $f$ , то при его уменьшении ближняя плоскость будет смещаться ближе к камере. Аналогичные рассуждения показывают, что  $u_r > u$  и при уменьшении  $f$  дальняя плоскость будет удаляться от камеры. Следовательно, глубина поля зрения у короткофокусных линз больше, чем у длиннофокусных. (К сожалению, короткофокусные линзы обычно характеризуются сильной радиальной дисторсией.)

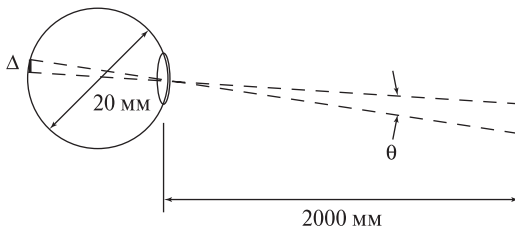
**Связь разрешения и размытия.** ПЗС-камера с идеальным объективом в наилучшем случае позволяет различать  $n/2$  отдельных линий среди  $n$  строк

пикселей. Промежуток между двумя последовательными линиями равен одному пикселу. С помощью ПЗС-матрицы размерами  $512 \times 512$  можно обнаружить сетку из 256 темных линий, разделенных светлыми линиями шириной в один пиксел. (Для оптимального совмещения сеточного шаблона со строками пикселей можно немного смещать камеру перпендикулярно оптической оси.) Если размытие образов точек будет превышать размер пиксела, то это приведет к смазу линий и вместо сетки будет наблюдаться однородное серое изображение.

Приведенные выше формулы позволяют определить параметры оборудования, необходимого для получения изображений в конкретной прикладной задаче. При подборе ПЗС-камеры и объектива целесообразно учитывать, какие характерные признаки требуется обнаруживать в решаемой задаче и на каких расстояниях от камеры будут располагаться наблюдаемые объекты.

**108 Определение. Разрешающая способность** (resolving power) камеры определяется как  $R_p = 1/(2\Delta)$  в единицах линий на миллиметр (или дюйм), где  $\Delta$  равно промежутку между пикселями в миллиметрах (или дюймах).

Допустим, квадратная ПЗС-матрица со стороной 10 мм содержит  $500 \times 500$  пикселей. Разрешающая способность этого датчика равна  $1/(2 \times 2 \times 10^{-2})$  мм/линий или 25 линий/мм. Предположим, что молекулы бромистого серебра на черно-белой фотопленке расположены друг от друга на расстоянии примерно  $5 \times 10^{-3}$  мм. В таком случае разрешающая способность фотопленки составляет 100 линий/мм или 2500 линий/дюйм. Цветочувствительные колбочки человеческого глаза особенно плотно расположены в центральной области сетчатки, называемой *фовеа*. Промежутки между ними равны примерно  $\Delta = 0.0025$  мм. Это значение позволяет оценить разрешающую способность сетчатки глаза как 200 линий/мм (5000 линий/дюйм). Диаметр глаза равен примерно 20 мм. Вычислим стягиваемый угол (см. рис 12.31). Он равен  $\theta \approx \sin(\theta) = 2\Delta/20$  мм =  $2.5 \times 10^{-4}$  радиан, т. е. примерно одной угловой минуте. Это значит, что человек способен различить линию толщиной 0.5 мм на стене, расположенной на расстоянии 2 метра.



**Рис. 12.31.** Образ малого объекта на сетчатке человеческого глаза.

## 12.8 ИТОГИ

В этой главе описывались взаимосвязи между структурными компонентами двумерных изображений и поверхностями и объектами в трехмерном пространстве. Люди пользуются многими из этих взаимосвязей и их комбинациями для восприятия окружающего мира и ориентации в нем. Хотя нами подробно рассматривались только вопросы определения глубины с помощью стереоскопических изображений и на основе фокусировки, известны количественные модели и для многих других явлений, в том числе для определения формы по данным освещенности или по текстурной информации. Эти модели могут оказаться полезными для художников, особенно тех, кто пользуется инструментами компьютерной графики для представления трехмерной структуры объектов с помощью двумерного графического дисплея. В гл. 13 показано применение некоторых из этих методов для автоматического распознавания структуры трехмерных сцен по двумерным изображениям. Хотелось бы предупредить читателя, что некоторые из этих моделей являются неустойчивыми и неточными при использовании где-либо за пределами контролируемых лабораторных условий. Разработка зрительных систем реального времени для навигации роботов в естественной среде представляет собой сложную задачу, которая является темой интенсивных современных исследований.

---

### Упражнение 12.26

Найдите в художественной книге иллюстрацию картины с изображением площади или акрополя и сделайте ее ксерокопию. Обозначьте на ней точки схода и линию горизонта, которые использовались художником при написании картины.

## 12.9 ЛИТЕРАТУРА

Классическая монография в области изучения признаков, используемых при визуальном восприятии информации человеком, принадлежит специалисту по зрительной физиологии Дж. Дж. Гибсону [J.J. Gibson, 1950]. Многие работы по компьютерному зрению, выполненные в 1980-х гг., основаны на материале этой книги.

Ряд экспериментов был выполнен под влиянием подхода Дэвида Марра [David Marr, 1982], который предложил *парадигму информационной обработки*. В рамках этой парадигмы предполагается, что можно сначала изолировать информацию, используемую в процессе принятия решения или восприятия, затем построить математические модели и в результате получить возможные реализации моделей операций зрительной системы человека. Марр также считал, что в зрительной системе человека выполняется формирование достаточно полных описаний поверхностей сцен. В настоящее время эта точка зрения распространена не слишком широко.

В книге физиолога Ирвинга Рока [Irving Rock, 1983], представлен ретроспективный обзор экспериментальных исследований и делается заключение о том, что визуальное восприятие требует интеллектуальных операций и для него характерны многие свойства мыслительной деятельности. В этой книге описываются и свойства человеческого зрения, и методология его изучения.

Промежуточное представление данных изображений в виде внутренних изображений было предложено в статье [Barrow, Tenenbaum, 1978]. Эта модель похожа на 2,5-мерный чертеж, предложенный Дэвидом Марром и описанный в его книге в 1982 г. Наше рассмотрение внутреннего изображения во многом следует гл. 3 учебника [Charniak, McDermott, 1985].

В работах [Huffman, 1971] и [Clowes, 1971] рассмотрены стыковые ограничения на множестве объектов с плоскими гранями. Расширение этого множества было выполнено в работе [Waltz, 1975] для учета теневых областей и нетрехгранных углов, что привело к увеличению каталога стыков до тысяч элементов — вероятно, это слишком большое количество для человека, но вполне допустимое для компьютерной обработки. Вальс (Waltz) разработал эффективный алгоритм маркировки отрезков, который часто называется *фильтрацией Вальса*. Учебник по искусственному интеллекту [Winston, 1977] может служить источником подробной информации по вопросу интерпретации формы в сценах объектов с плоскими гранями. В нем также описан способ формирования каталогов стыков.

Рассматривавшийся нами параллельный релаксационный метод описан в работе [Rosenfeld и др., 1976], и многих других похожих работах, некоторые из которых основаны на результатах Вальса. В книге [Sugihara, 1986] можно найти информацию о геометрических ограничениях, предотвращающих интерпретацию контурных изображений, которые недопустимы в сценах объектов с плоскими гранями. В работе [Malik, 1987] показано, как расширить каталог меток отрезков и стыков для учета большого класса объектов с криволинейными поверхностями. В работе [Stockman и др., 1990] демонстрируется использование разреженных карт глубины для реконструкции и интерпретации неполных контурных изображений сцен гораздо более общего характера по сравнению со сценами с объектами с плоскими гранями и трехгранными углами. Перспективное преобразование подробно рассматривается в двухтомнике [Haralick, Shapiro, 1992/93].

В известной работе [Marr, Poggio, 1979] стереоскопическая зрительная система человека описывается в терминах информационной обработки и предлагается модель, похожая на релаксационную процедуру. В статье [Tanimoto, 1998] есть раздел о том, как стереопары можно использовать для повышения интереса к изучению математики; в частности, приведено несколько цветных стереопар, по которым люди способны воспринимать трехмерные объекты. Создание материалов для стереоскопического восприятия людьми остается областью активных современных исследований. Например, в статье [Peleg, Ben-Ezra, 1999] описывается построение стереоскопических изображений ис-



торических памятников с помощью одной движущейся камеры. Устройства автоматической фокусировки широко представлены на рынке видео- и фотокамер. Очевидно, что для вычисления глубины поверхностей сцены можно использовать быстрые и недорогие микропроцессорные устройства. Базовый материал по данному вопросу представлен в работах [Krotkov, 1987], [Nayar и др., 1992], [Subbarao, Tyan, 1998].

1. Barrow, H., and J. Tenenbaum. 1978. Recovering intrinsic scene characteristics from images. In *Computer Vision Systems*, A. Hansom and E. Riseman, eds. Academic Press, New York.
2. Charniak, E., and D. McDermott. 1985. *Artificial Intelligence*. Addison-Wesley, Reading, MA.
3. Clowes, M. 1971. On seeing things. *Artificial Intelligence*, v. 2:79–116.
4. Gibson, J. J. 1950. *The Perception of the Visual World*. Houghton-Mifflin, Boston.
5. Haralick, R., and L. Shapiro. 1992/3. *Computer and Robot Vision, Volumes I and II*. Addison-Wesley, Reading, MA.
6. Huffman, D. 1971. Impossible objects as nonsense sentences. In *Machine Intelligence*, v. 6, B. Meltzer and D. Michie, eds. Elsevier, New York, 295–323.
7. Kender, J. 1980. *Shape from Texture*, Ph.D. dissertation. Dept. of Computer Science, Carnegie Mellon Univ., Pittsburgh, PA.
8. Koenderink, J. 1984. What does the occluding contour tell us about solid shape? *Perception*, v. 13.
9. Krotkov, E. 1987. Focusing, *Int. J. Comput. Vision*, v. 1:223–237.
10. Malik, J. 1987. Interpreting line drawings of curved objects. *Int. J. Comput. Vision*, v. 1(1).
11. Marr, D., and T. Poggio. 1979. A computational theory of human stereo vision. *Proc. Royal Society*, v. B 207:207–301.
12. Marr, D. 1982. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. W. H. Freeman and Co., New York.
13. Nayar, S. 1992. Shape from focus system. *Proc. Comput. Vision and Pattern Recog.*, Champaign, Illinois (June 1992), 302–308.
14. Peleg, S., and M. Ben-Ezra. 1999. Stereo panorama with a single camera. *Proc. Comput. Vision and Pattern Recog.*, Fort Collins, CO (23–25 June 1999), v. I:395–401.
15. Rock, I. 1983. *The Logic of Perception*. A Bradford Book, MIT Press, Cambridge, MA.
16. Rosenfeld, A., R. Hummel, and S. Zucker. 1976. Scene labeling by relaxation processes. *IEEE Trans. SMC*, v. 6.

17. Stockman, G., G. Lee, and S. W. Chen. 1990. Reconstructing line drawings from wings: the polygonal case. *Proc. of Int. Conf. Comput. Vision 3*, Osaka, Japan.
18. Subbarao, M., and J-K. Tyan. 1998. Selecting the optimal focus measure for autofocusing and depth-from-focus. *IEEE-T-PAMI*, v. 20(8):864–870.
19. Sugihara, K. 1986. *Machine Interpretation of Line Drawings*. MIT Press, Cambridge, MA.
20. Tanimoto, S. 1998. Connecting middle school mathematics to computer vision and pattern recognition. *Int. J. Pattern Recog. and Artificial Intelligence*, v. 12(8):1053–1070.
21. Waltz, D. 1975. Understanding line drawings of scenes with shadows. In *The Psychology of Computer Vision*, P. Winston, ed. McGraw-Hill, New York, 19–91.
22. Winston, P. 1977. *Artificial Intelligence*. Addison-Wesley, Reading, MA.

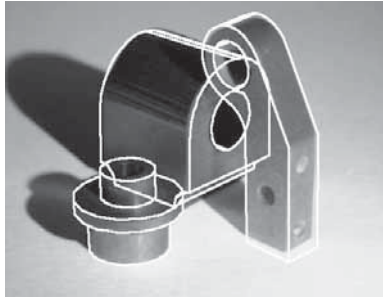
# Восприятие трехмерных сцен. Оценка пространственного положения и ориентации объектов

Основная цель данной главы заключается в установлении количественных взаимосвязей между структурами на двумерных изображениях и соответствующими трехмерными структурами реального мира. В предыдущей главе эти взаимосвязи исследовались качественно. Теперь будут описаны методы измерений, пригодные для использования в задачах распознавания и инспекционного контроля трехмерных объектов, а также в информационных системах манипуляционных и мобильных роботов.

На рис. 13.1 в качестве примера прикладной задачи показаны изображения, обрабатываемые измерительной системой для определения позы водителя автомобиля. Эта система была разработана в рамках исследований по проектированию дизайна салона и органов управления автомобилем. В другом приложении (рис. 13.2) зрительная система должна распознавать трехмерные



**Рис. 13.1.** Изображения водителя, полученные с помощью двух из четырех установленных в автомобиле видеокамер. Измерительная система с несколькими камерами применяется для вычисления трехмерных координат характерных точек на теле водителя (на рисунке они отмечены яркими эллипсами) и последующего определения его позы. (С любезного разрешения Herbert Reynolds, Лаборатория эргономики Университета штата Мичиган.)



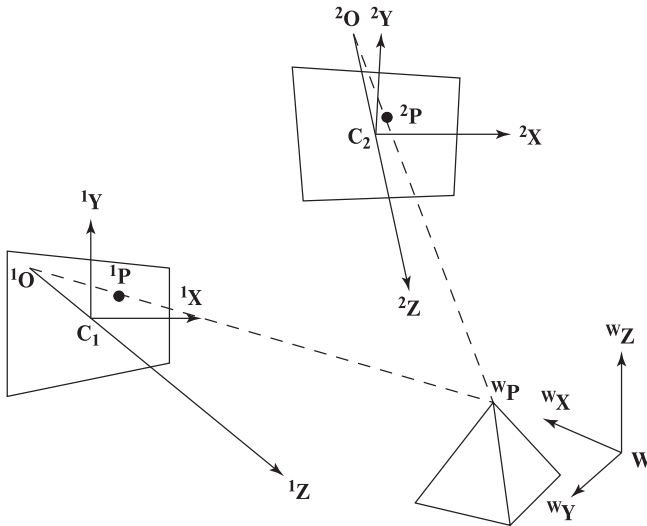
**Рис. 13.2.** Оверлейное наложение компьютерной графики на изображение трех трехмерных объектов в программе распознавания и локализации трехмерных объектов. Для выполнения наложения система распознавания сопоставляет двумерные фрагменты изображения с частями трехмерной модели и вычисляет параметры геометрического преобразования, необходимого для получения наблюдаемого изображения из моделей объектов. Затем программе управления роботом передаются идентификаторы и параметры расположения каждой детали. (С любезного разрешения Mauro Costa.)

объекты, вычислять их местоположение и ориентацию для автоматического захвата объектов манипулятором робота. В данном приложении зрительная система является частью информационной системы робота и в качестве результатов по требованию управляющего программного обеспечения робота выдает трехмерные мировые координаты обнаруженных объектов.

В данной главе рассматриваются некоторые технические и математические аспекты трехмерного компьютерного зрения. Задачи формулируются в интуитивно понятных геометрических терминах. Затем для них разрабатываются математические модели. Важную роль в них играют алгебраические преобразования в трехмерном пространстве. В главе описывается значение трехмерных моделей объектов, а также обсуждаются несколько конструкций зрительных систем и процедуры их калибровки.

### **13.1 УСТРОЙСТВО СТЕРЕОСКОПИЧЕСКОЙ СИСТЕМЫ КОМПЬЮТЕРНОГО ЗРЕНИЯ ОБЩЕГО НАЗНАЧЕНИЯ**

На рис. 13.3 приведена общая схема зрительной системы из 2 камер, наблюдающих одну и ту же область пространства. В компьютерной графике часто используется правая система координат, у которой отрицательное направление оси  $z$  соответствует направлению «от камеры», так что более далекие от камеры объекты имеют большие по модулю отрицательные координаты глубины. В данной главе в большинстве моделей будем считать глубину положительной, но иногда будем пользоваться другой системой координат, чтобы придерживаться соответствия с общепринятой формой записи уравнений. На рис. 13.3 приведена общая схема стереоскопической зрительной системы

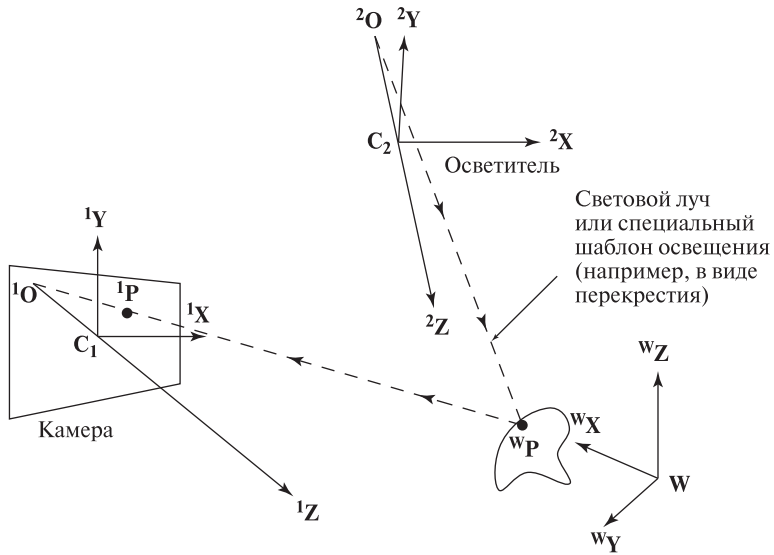


**Рис. 13.3.** Две камеры  $C_1$  и  $C_2$  направлены на одну область трехмерного пространства. Точка  $P$  изображается в виде точки  ${}^1P$  на изображении камеры  $C_1$  и как точка  ${}^2P$  на изображении камеры  $C_2$ .

без специальных ограничений на расположение камер (в отличие от гл. 12). Обе камеры направлены на один участок рабочего пространства. В данном случае рабочим пространством является все трехмерное пространство. С ним связана глобальная система координат  $W$ . Из интуитивных соображений можно заключить, что координаты трехмерной точки  ${}^wP = [{}^wP_x, {}^wP_y, {}^wP_z]^t$  в рабочем пространстве можно получить путем вычисления точки пересечения двух лучей зрения  ${}^wP{}^1O$  и  ${}^wP{}^2O$ . Ниже будут приведены алгебраические выражения для нахождения этой точки. Они просты, но на практике их приходится усложнять для учета ошибок измерений.

Для выполнения вычислений трехмерных координат точки в общей стереоскопической схеме (рис. 13.3) должны быть известны следующие элементы:

- Должно быть известно расположение камеры  $C_1$  в рабочем пространстве  $W$  и ряд внутренних параметров камеры, например, фокусное расстояние. Все эти параметры можно записать в форме *матрицы камеры*, которая математически описывает лучи в трехмерном пространстве для формирования каждой точки изображения  ${}^1P$ . В разделах 13.3 и 13.7 описываются процедуры калибровки камеры, предназначенные для экспериментального определения этих параметров.
- Аналогично, должны быть известны расположение и внутренние параметры камеры  $C_2$ , т. е. нам требуется матрица камеры  $C_2$ .
- Надо установить соответствие между точкой трехмерного пространства и точками двух изображений:  $({}^wP, {}^1P, {}^2P)$ .



**Рис. 13.4.** Осветитель может заменить одну камеру в стереоскопической системе. Остаются справедливыми все геометрические и алгебраические связи, показанные на рис. 13.3. Осветитель может добавить характерные свойства у поверхности, не обладающей таковыми без подсветки.

- Требуется формула для вычисления координат  ${}^wP$  по двум проектирующим лучам  ${}^wP^1O$  и  ${}^wP^2O$ .

Перед рассмотрением перечисленных вопросов опишем некоторые важные разновидности конфигурации, показанной на рис. 13.3.

- Показанная на рис. 13.3 конфигурация содержит две камеры, калиброванные в мировой системе координат. Трехмерные координаты характерных точек вычисляются путем нахождения точек пересечения двух лучей зрения, проходящих через соответствующие точки изображений.
- Одну из камер можно заменить источником света, освещающим одну или несколько точек поверхности световым лучом или специальным шаблоном освещения (например, в виде перекрестия). Подобная система показана на рис. 13.4. Как будет описано далее, калибровку источника света можно выполнить аналогично калибровке камеры: луч источника света математически описывается аналогично лучу зрения камеры. Применение осветителя особенно удобно при выполнении измерений точек поверхности, на которой нет каких-либо внешне различимых особенностей.
- Если известна модель наблюдаемого объекта, то одну камеру можно убрать. Допустим, что на рис. 13.3 задана высота пирамиды. Тогда нам уже известна координата  ${}^wP_z$ , а это значит, что на точку  $P$  накладывается ограничение: она должна лежать в плоскости  $z = {}^wP_z$ . Две

другие координаты точки легко найти как пересечение двух лучей зрения камеры  $C_1$  с известной плоскостью. Информация о модели во многих случаях добавляет достаточно ограничений, позволяющих обойтись для измерений одной камерой.

## 13.2 АФФИННЫЕ ПРЕОБРАЗОВАНИЯ В ТРЕХМЕРНОМ ПРОСТРАНСТВЕ

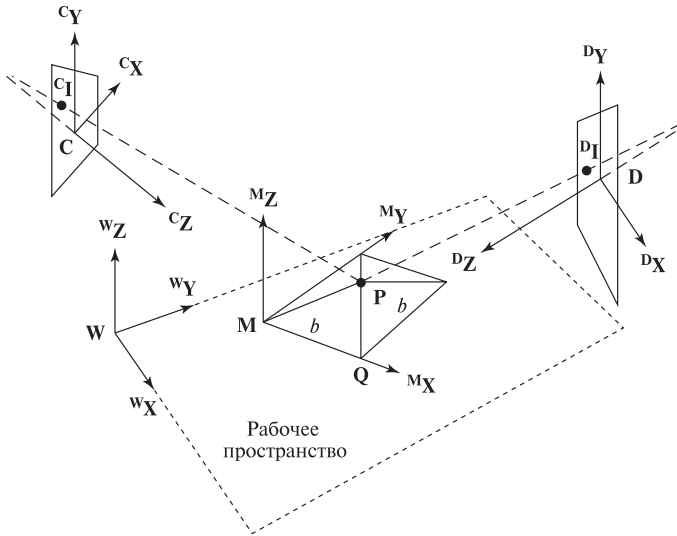
Аффинные преобразования в двумерном пространстве рассматривались в гл. 11. В данной главе эти преобразования будут рассматриваться в трехмерном пространстве. Они широко применяются не только в трехмерном машинном зрении, но и в робототехнике, в системах виртуальной реальности. Основными преобразованиями являются перенос, поворот, масштабирование и сдвиг. Обобщение этих преобразований на трехмерное пространство выполняется довольно просто, но проиллюстрировать некоторые из них оказывается сложнее. Для удобства вычислений снова воспользуемся однородными координатами, т. е. вместо трехмерных координат точки  $[P_x, P_y, P_z]$  будем работать с ее однородными координатами  $[sP_x, sP_y, sP_z, s]$  (где  $s$  — ненулевой масштабный множитель, обычно  $s = 1$ ). Как и раньше, координаты точек будем записывать в виде векторов-столбцов, но иногда, в очевидных местах, будем пропускать символ транспонирования. В данной главе у имен точек часто используются верхние индексы, так как требуется работать с большим, чем в гл. 11, количеством координатных систем. Ниже набор преобразований будет расширен за счет проекционных преобразований, отображающих трехмерное пространство на двумерное: перспективное, ортографическое и слабое перспективное преобразования.

### 13.2.1 Системы координат

*Системы координат* необходимы для введения количественных характеристик расположения точек в пространстве. На рис. 13.5 показана сцена, в которой определены четыре различные системы координат. Таким образом точка  $\mathbf{P}$ , вершина пирамиды, имеет четыре различных координатных представления. Во-первых, точка описывается в модельной системе координат  $\mathbf{M}$  как  ${}^M\mathbf{P} = [{}^MP_x, {}^MP_y, {}^MP_z] = [b/2, b/2, \frac{\sqrt{2}}{2}b]$ , где  $b$  — сторона основания пирамиды. Расположение пирамиды в мировом пространстве показано на рис. 13.5. Координаты вершины пирамиды в мировой системе координат можно записать так:

$${}^W\mathbf{P} = [{}^WP_x, {}^WP_y, {}^WP_z] = \mathbf{TR} \left[ \frac{b}{2}, \frac{b}{2}, \frac{\sqrt{2}}{2}b \right] \quad (13.1)$$

где  $\mathbf{TR}$  — это комбинация поворота и переноса системы координат  $\mathbf{M}$  относительно системы координат  $\mathbf{W}$ . Наконец, если два датчика  $\mathbf{C}$  и  $\mathbf{D}$  видят пирамиду с разных сторон рабочего стола, то отношение «слева-справа» между точками  $\mathbf{P}$  и  $\mathbf{Q}$  в двух системах координат различно, т. е.  ${}^C\mathbf{P} = [{}^CP_x, {}^CP_y, {}^CP_z] \neq {}^D\mathbf{P} = [{}^DP_x, {}^DP_y, {}^DP_z]$ .



**Рис. 13.5.** Точка  $P$  имеет различные координаты в четырех различных системах координат — (1)  $M$  — система координат модели; (2)  $W$  — мировая система координат (или система координат рабочего пространства); (3) датчик  $C$  и (4) датчик  $D$ . В различных системах координат точка  $P$  выглядит по-разному относительно других точек. Например, в системе  $C$  точка  $P$  кажется слева от  $Q$ , но в системе координат  $D$  — уже справа от  $Q$ .

Для определения расположения датчиков относительно друг друга и относительно трехмерных объектов сцены, а также для записи правил манипуляции с их координатами необходимы математические методы для работы с координатами точек в различных системах координат. Эти же методы позволят описать движение объекта в пространстве. Иногда мы будем использовать удобную форму записи для обозначения системы координат, в которой описывается точка, и «направления» преобразования координат. Например, преобразование точки модели  ${}^M P$  из модельной системы координат в мировую систему координат  ${}^W T$  обозначим  ${}^W T_M$ :

$${}^W P = {}^W T_M {}^M P \quad (13.2)$$

Эта форма записи была введена в учебнике по робототехнике [Craig, 1986]: она удобна для рассмотрения перемещения объектов и для их сопоставления. В простых случаях, когда выбор системы координат очевиден, мы будем упрощать эту запись. Далее перейдем к изучению преобразований.

### 13.2.2 Перенос

При переносе компоненты вектора переноса  $x_0, y_0, z_0$  складываются с координатами точки  ${}^1 P$  в системе координат  $1$  для получения координат точки  ${}^2 P$  в системе координат  $2$ . В примере на рис. 13.5 требуется выполнить некоторый



перенос (и поворот), чтобы связать точку в системе координат модели и ее мировые координаты.

$${}^2\mathbf{P} = \mathbf{T}(x_0, y_0, z_0) {}^1\mathbf{P}$$

$${}^2\mathbf{P} = \begin{bmatrix} {}^2P_x \\ {}^2P_y \\ {}^2P_z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & x_0 \\ 0 & 1 & 0 & y_0 \\ 0 & 0 & 1 & z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^1P_x \\ {}^1P_y \\ {}^1P_z \\ 1 \end{bmatrix} \quad (13.3)$$

### 13.2.3 Масштабирование

С помощью матрицы трехмерного масштабирования по каждому из направлений системы координат можно задать различные масштабные множители. Иногда все они одинаковы, например, при замене единиц измерений или при однородном масштабировании для получения экземпляра модели заданного размера.

$${}^2\mathbf{P} = \mathbf{S} {}^1\mathbf{P} = \mathbf{S}(s_x, s_y, s_z) {}^1\mathbf{P}$$

$$\begin{bmatrix} {}^2P_x \\ {}^2P_y \\ {}^2P_z \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & {}^2P_x \\ s_y & {}^2P_y \\ s_z & {}^2P_z \\ 1 & \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^1P_x \\ {}^1P_y \\ {}^1P_z \\ 1 \end{bmatrix} \quad (13.4)$$

### 13.2.4 Поворот

Наиболее просто записывается матрица поворота вокруг одной из координатных осей. Векторами-столбцами этой матрицы будут компоненты преобразованных единичных векторов (т. е. после поворота). (Напомним, что любое трехмерное линейное преобразование полностью определяется результатами преобразования трех базисных векторов.) Поворот вокруг оси  $OZ$  похож на 2-мерное преобразование из гл. 11, за исключением того, что оно теперь сохраняет координату  $z$  у трехмерной точки. На рис. 13.6 показано, как преобразуются единичные вектора при элементарных поворотах вокруг координатных осей.

**Поворот на угол  $\theta$  вокруг оси  $OX$ :**

$${}^2\mathbf{P} = \mathbf{R}({}^1X, \theta) {}^1\mathbf{P}$$

$$\begin{bmatrix} {}^2P_x \\ {}^2P_y \\ {}^2P_z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^1P_x \\ {}^1P_y \\ {}^1P_z \\ 1 \end{bmatrix} \quad (13.5)$$

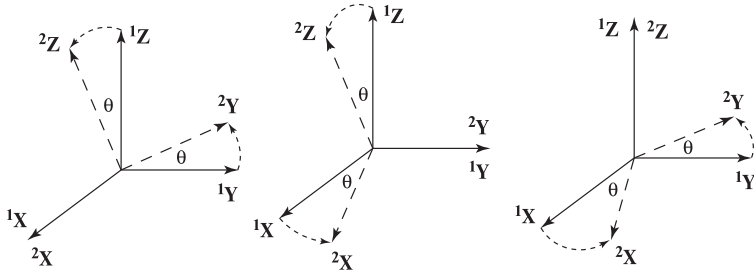


Рис. 13.6. Повороты на угол  $\theta$  вокруг оси  $OX$  (слева),  $OY$  (в центре) и  $OZ$  (справа).

**Поворот на угол  $\theta$  вокруг оси  $OY$ :**

$${}^2\mathbf{P} = \mathbf{R}({}^1Y, \theta) {}^1\mathbf{P}$$

$$\begin{bmatrix} {}^2P_x \\ {}^2P_y \\ {}^2P_z \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^1P_x \\ {}^1P_y \\ {}^1P_z \\ 1 \end{bmatrix} \quad (13.6)$$

**Поворот на угол  $\theta$  вокруг оси  $OZ$ :**

$${}^2\mathbf{P} = \mathbf{R}({}^1Z, \theta) {}^1\mathbf{P}$$

$$\begin{bmatrix} {}^2P_x \\ {}^2P_y \\ {}^2P_z \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^1P_x \\ {}^1P_y \\ {}^1P_z \\ 1 \end{bmatrix} \quad (13.7)$$

### Упражнение 13.1

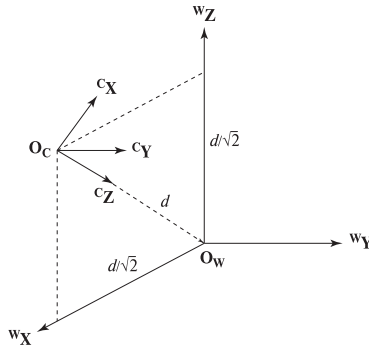
Проверьте, что столбцы матриц поворота вокруг координатных осей ортонормированны. Выполните аналогичную проверку для строк.

### Упражнение 13.2

Постройте матрицу поворота на угол  $\pi/4$  против часовой стрелки вокруг оси, проходящей через начало координат и точку  $[1, 1, 0]^t$ .

### Упражнение 13.3

Покажите, как построить матрицу поворота общего вида для заданного угла  $\theta$  радиан вокруг оси, заданной направляющими косинусами  $[c_x, c_y, c_z]^t$ .



*Пример:* Требуется записать матрицу комбинированного преобразования из поворота и переноса, необходимую для преобразования мировых координат  $\mathbf{W}$  в координаты камеры  $\mathbf{C}$  (см. рис. ниже, начало координат  $\mathbf{O}_w$  в системе  $\mathbf{C}$  имеет координаты  $(0, 0, d)$ ).

Сначала запишем матрицу поворота  $\mathbf{R}$ . Для этого запишем координаты базисных векторов системы  $\mathbf{W}$  в системе координат  $\mathbf{C}$ . Тогда мы сможем преобразовать любую точку из системы координат  $\mathbf{W}$  в систему  $\mathbf{C}$ .

$$\begin{aligned} w_X &= \frac{-\sqrt{2}}{2} c_X + 0 c_Y + \frac{-\sqrt{2}}{2} c_Z \\ w_Y &= 0 c_X + 1 c_Y + 0 c_Z \\ w_Z &= \frac{\sqrt{2}}{2} c_X + 0 c_Y + \frac{-\sqrt{2}}{2} c_Z \end{aligned} \quad (13.8)$$

Координаты преобразованных по формулам (13.8) базисных векторов системы  $\mathbf{W}$  составляют 3 столбца матрицы поворота, описывающей ориентацию системы координат камеры  $\mathbf{C}$  относительно системы координат  $\mathbf{W}$ . Кроме поворота, точки мировой системы координат надо перенести вдоль оси  $z$ , так, чтобы начало координат  $\mathbf{W}$  в системе  $\mathbf{C}$  имело координаты  $[0, 0, d]^t$ . Поэтому искомое преобразование будет иметь вид:

$${}^C_W \mathbf{TR} = \begin{bmatrix} \frac{-\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} & 0 \\ 0 & 1 & 0 & 0 \\ \frac{-\sqrt{2}}{2} & 0 & \frac{-\sqrt{2}}{2} & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (13.9)$$

Проверьте, что  ${}^C_W \mathbf{TR} \mathbf{W} \mathbf{O}_w = {}^C_W \mathbf{TR} [0, 0, 0, 1]^t = [0, 0, d, 1]^t = {}^C \mathbf{O}_w$  и  ${}^C_W \mathbf{TR} \mathbf{W} \mathbf{O}_c = {}^C_W \mathbf{TR} [d\frac{\sqrt{2}}{2}, 0, d\frac{\sqrt{2}}{2}, 1]^t = [0, 0, 0, 1]^t = {}^C \mathbf{O}_c$ .

#### Упражнение 13.4

В данном упражнении используем системы координат из предыдущего примера. Поместите единичный куб в начале мировой системы координат  $\mathbf{O}_w$ .

Преобразуйте координаты вершин куба  $K_j$  в систему координат камеры. Проверьте для каких-либо четырех ребер, что они имеют единичную длину, вычисляя  $\|K_i - K_j\|$  в системе координат камеры.

### 13.2.5 Поворот вокруг произвольной оси

Поворот вокруг произвольной оси можно представить в виде уравнения (13.10). Матрица коэффициентов  $r_{ij}$  является ортонормированной: все ее строки (и столбцы) являются взаимно ортогональными единичными векторами. Все матрицы элементарных поворотов (вокруг координатных осей) также обладают этим свойством. Любой поворот в трехмерном пространстве можно представить в виде поворота на некоторый угол  $\theta$  вокруг некоторой оси  $\mathbf{A}$ . Это может быть произвольная ось, не обязательно совпадающая с одной из координатных осей. Чтобы убедиться в этом, предположим, что базисный вектор  ${}^1\mathbf{X}$  преобразуется в другой вектор  ${}^2\mathbf{X}$ . Ось вращения  $\mathbf{A}$  можно найти, вычисляя векторное произведение  ${}^1\mathbf{X}$  и  ${}^2\mathbf{X}$ . В случае, когда вектор  ${}^1\mathbf{X}$  не изменяется при повороте, этот вектор совпадает с осью вращения.

$${}^2\mathbf{P} = \mathbf{R}(A, \theta) {}^1\mathbf{P}$$

$$\begin{bmatrix} {}^2P_x \\ {}^2P_y \\ {}^2P_z \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^1P_x \\ {}^1P_y \\ {}^1P_z \\ 1 \end{bmatrix} \quad (13.10)$$

В качестве следствия можно отметить: *результат движения твердого тела с момента времени  $t_1$  до  $t_2$  можно представить с помощью одного вектора переноса и одной матрицы поворота, независимо от формы траектории тела между указанными моментами времени.* Для хранения параметров переноса и поворота достаточно одной матрицы преобразования в однородных координатах: в этой матрице 9 элементов задают поворот и 3 элемента — перенос.

$$\begin{bmatrix} {}^2P_x \\ {}^2P_y \\ {}^2P_z \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^1P_x \\ {}^1P_y \\ {}^1P_z \\ 1 \end{bmatrix} \quad (13.11)$$

### Упражнение 13.5

Для рис. 13.7 постройте матрицу преобразования, отображающего все углы первого блока (у которого один из углов совпадает с началом координат) в углы второго блока. Требуемое соответствие углов:  ${}^1\mathbf{O} \rightarrow {}^2\mathbf{O}$ ,  ${}^1\mathbf{P} \rightarrow {}^2\mathbf{P}$ .

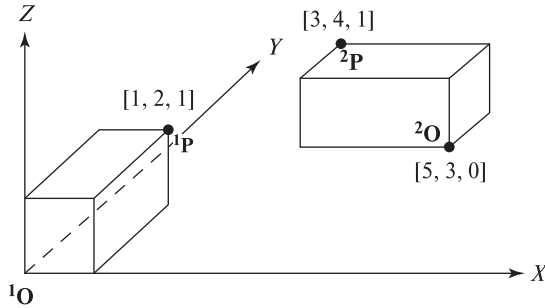


Рис. 13.7. Два экземпляра одной и той же модели блока.

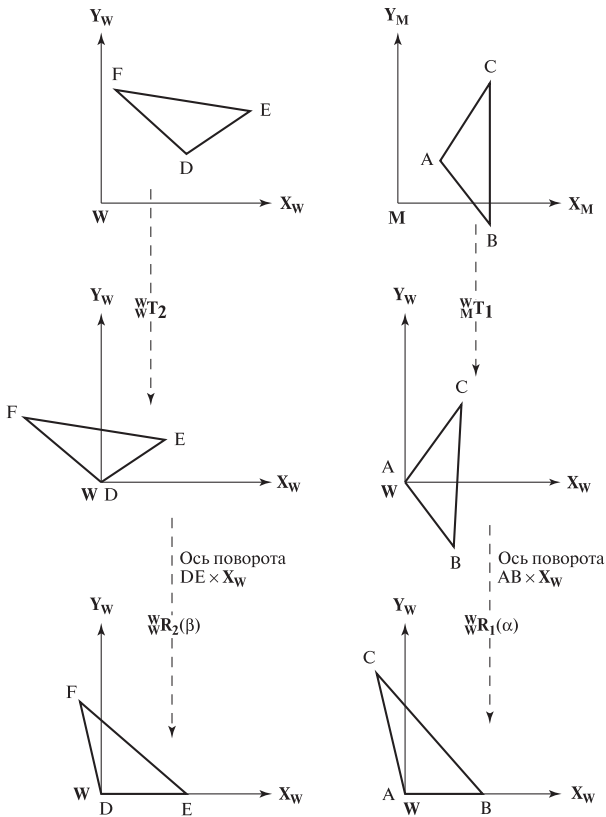
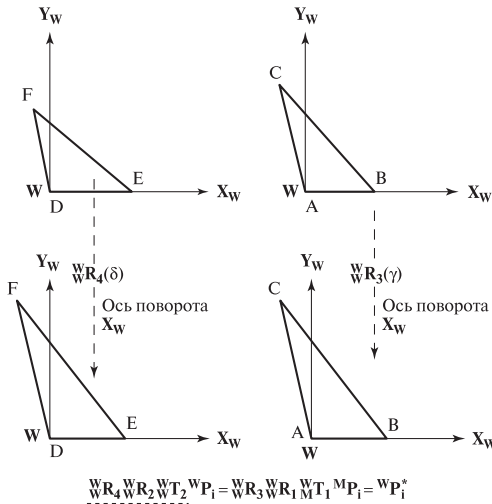


Рис. 13.8. Часть I: Совмещение двух конгруэнтных треугольников.  $\triangle ABC$  является модельным треугольником,  $\triangle DEF$  — треугольником, полученным в результате измерений. Во-первых, точки объектов переносятся так, чтобы  $A$  и  $D$  попали в начало координат. Во-вторых, каждый треугольник поворачивается так, чтобы стороны  $AB$  и  $DE$  попали на координатную ось  $OX$ .



**Рис. 13.9.** Часть II: Продолжение процедуры совмещения двух конгруэнтных треугольников. Каждый треугольник поворачивается так, чтобы треугольник оказался в плоскости  $XW$ . Ось вращения является ось  $OX$ . Углы поворота определяются как углы между сторонами  $AC$  и  $DF$  и плоскостью  $XW$ .

**Упражнение 13.6.** Обращение матрицы поворота

Докажите, что матрица поворота всегда имеет обратную матрицу. Как выражается обратная матрица для поворота  $R(A, \theta)$  из уравнения (13.10)?

**13.2.6 Совмещение фигур посредством аффинных преобразований**

Рассмотрим задачу совмещения модели треугольника с треугольником, построенным по данным измерений. Этот пример убедительно продемонстрирует необходимость точного вычисления параметров преобразований. Данный пример можно будет использовать в качестве основы для совмещения любой модели твердого тела с измеренными точками по соответствию трех точек. Задача будет решаться алгебраически с использованием основных преобразований, которые были рассмотрены выше. Ход решения иллюстрируется геометрическими построениями на рис. 13.8 и 13.9.

Задача заключается в том, что надо найти такое преобразование  ${}^W M T$ , которое преобразует вершины модельного треугольника  $A, B$  и  $C$  в вершины конгруэнтного треугольника  $D, E, F$  в рабочем пространстве.

Преобразуем оба треугольника так, чтобы сторона  $AB$  попала на ось  ${}^W O W X$  и чтобы весь треугольник лежал в плоскости  $XW$  мировой системы координат  $W$ . В такой конфигурации вершины  $A$  и  $D$  будут совпадать, то же можно сказать про  $B$  и  $E, C$  и  $F$ . Уравнение преобразования, выполняющего указанное совмещение вершин, можно переписать таким образом, чтобы получить

**Вычисление параметров преобразования  ${}^W_M T$ , совмещающего точки модели A, B, C с измеренными точками в мировой системе координат D, E, F.**

1. Ввод трехмерных координат трех точек модели A, B, C и соответствующих измеренных точек D, E, F.
2. Построение преобразования переноса  ${}^W_M T_1$ , обеспечивающего попадание модельной точки A в начало мировой системы координат W. Построение преобразования переноса  ${}^W_M T_2$ , обеспечивающего попадание измеренной (в мировой системе координат) точки D в начало мировой системы координат. Таким образом, на этом шаге выполнено совмещение точек A и D в системе координат W.
3. Построение преобразований поворота  ${}^W_M R_1$  и  ${}^W_M R_2$  для совмещения сторон AB и DE с осью X. Таким образом, выполнено совмещение сторон AB и DE в системе координат W.
4. Построение преобразований поворота  ${}^W_M R_3$  и  ${}^W_M R_4$  для помещения точек C и F на плоскость XY. Теперь все три пары точек совмещены в системе координат W.
5. Модельный треугольник и треугольник, построенный по результатам измерений в мировой системе координат, совмещены. Это выражается уравнением:

$${}^W_M R_3 {}^W_M R_1 {}^W_M T_1 {}^M P_i = {}^W_M R_4 {}^W_M R_2 {}^W_M T_2 {}^W P_i \quad (13.12)$$

$${}^W P_i = (T_2^{-1} R_2^{-1} R_4^{-1} R_3 R_1 T_1) {}^M P_i \quad (13.13)$$

6. В качестве результата работы алгоритма возвращается преобразование  ${}^W_M T = (T_2^{-1} R_2^{-1} R_4^{-1} R_3 R_1 T_1)$

**Алгоритм 13.1.** Вычисление твердотельного преобразования для совмещения модельного треугольника с конгруэнтным треугольником в мировой системе координат.

искомое преобразование  ${}^W_M T$ , отображающее точки  ${}^M P_i$  в соответствующие точки  ${}^W P_i$ . Необходимые операции представлены в виде алгоритма 13.1. Очевидно, что каждая операция может быть выполнена и для каждой операции существует обратная. Программирование этих операций требует аккуратного выполнения и продумывания структур данных. Совмещение фигур является очень важной процедурой. Теоретически, любые два конгруэнтных твердых тела можно совместить, совместив любые три соответствующие точки. На практике ошибки измерений и вычислительные погрешности могут привести к значительным ошибкам и попыткам совмещения точек, сильно отстоящих от  $\triangle ABC$ . Обычно для совмещения объектов применяется процедура оптимизации, использующая большее количество точек.

### 13.3 МОДЕЛЬ КАМЕРЫ

Цель настоящего раздела — продемонстрировать, что в уравнении (13.14) модель камеры  $\mathbf{C}$  является матрицей перспективной проекции, а затем выяснить способы определения элементов этой матрицы при фиксированном положении камеры. Элементы этой матрицы могут быть использованы в компьютерных программах для выполнения трехмерных измерений.

$$\mathbf{I}_P = \mathbf{I}_W \mathbf{C} \mathbf{W}_P \quad (13.14)$$

$$\begin{bmatrix} s & I_{P_r} \\ s & I_{P_c} \\ s & \end{bmatrix} = \mathbf{I}_W \mathbf{C} \begin{bmatrix} W_{P_x} \\ W_{P_y} \\ W_{P_z} \\ 1 \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & 1 \end{bmatrix} \begin{bmatrix} W_{P_x} \\ W_{P_y} \\ W_{P_z} \\ 1 \end{bmatrix}$$

$$I_{P_r} = \frac{[c_{11} \ c_{12} \ c_{13} \ c_{14}] \circ [W_{P_x} \ W_{P_y} \ W_{P_z} \ 1]}{[c_{31} \ c_{32} \ c_{33} \ 1] \circ [W_{P_x} \ W_{P_y} \ W_{P_z} \ 1]}$$

$$I_{P_c} = \frac{[c_{21} \ c_{22} \ c_{23} \ c_{24}] \circ [W_{P_x} \ W_{P_y} \ W_{P_z} \ 1]}{[c_{31} \ c_{32} \ c_{33} \ 1] \circ [W_{P_x} \ W_{P_y} \ W_{P_z} \ 1]}$$

#### Упражнение 13.7. Обращение матрицы переноса

Докажите, что матрица переноса всегда имеет обратную матрицу. Как выражается обратная матрица для переноса  $\mathbf{T}(t_x, t_y, t_z)$  из уравнения (13.3)?

#### Упражнение 13.8

Очевидно, алгоритм 13.1 будет работать неправильно при  $\|A - B\| \neq \|D - E\|$ . Добавьте в алгоритм соответствующие проверки и возврат кодов ошибок для корректного диагностирования случаев, когда треугольники не являются конгруэнтными.

#### Упражнение 13.9. Программа для совмещения треугольников\*

На основе алгоритма 13.1 напишите и протестируйте программу, выполняющую совмещение модельного треугольника с конгруэнтным треугольником, заданным в мировых координатах. Каждую операцию алгоритма реализуйте в виде отдельной функции.

#### Упражнение 13.10

(а) Докажите, что преобразование, вычисленное алгоритмом 13.1, выполняет отображение точек модели на точки в мировой системе координат  $A \rightarrow D$ ,  $B \rightarrow E$ ,  $C \rightarrow F$ . (б) Докажите, что расстояние между любой точкой модели



и точками А, В, С не меняется после выполнения данного преобразования. (в) Докажите, что для совмещения модели  $n$ -угольного многоугольника с многоугольником, построенным по измеренным точкам, достаточно выполнить совмещение двух треугольников согласно алгоритму 13.1.

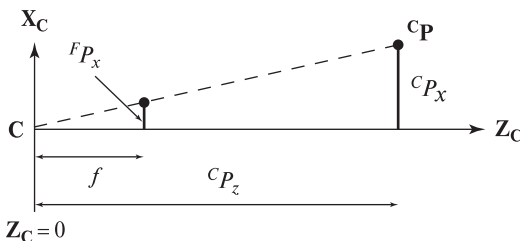
Далее будет показано, что матрица камеры размерами  $3 \times 4$   ${}^I_W C_{3 \times 4}$  является матрицей перспективной проекции, которая преобразует точку трехмерного мирового пространства  ${}^W P = [{}^W P_x, {}^W P_y, {}^W P_z]^t$  в точку изображения  ${}^I P = [{}^I P_r, {}^I P_c]^t$ . Эта матрица содержит достаточное количество параметров для моделирования изменения координат при переходе от мировой системы координат  $W$  к системе координат камеры  $C$  и для описания изменения масштаба вследствие перспективного проектирования и масштабирования мировых координат объекта для попадания на поверхность датчика изображения. В матричном уравнении используются однородные координаты: удаление масштабного множителя  $s$  с помощью скалярного произведения также показано в записи уравнения (13.14). Перейдем теперь к получению выражений для определения элементов матрицы  ${}^I_W C$ .

### 13.3.1 Матрица перспективной проекции

Уравнения перспективной проекции были получены в гл. 12. Они приведены ниже, под номером (13.15). Напомним, что эти уравнения были получены для простого случая, когда система координат камеры совпадает с мировой системой координат. Более того, размерность координат изображения  $[{}^F P_x, {}^F P_y]$  совпадает с размерностью мировой системы координат, т. е. координаты изображения измеряются не в пикселях. (Верхний индекс  $F$  означает «вещественное число», а не фокусное расстояние — оно обозначается буквой  $f$ .)

$$\begin{aligned} {}^F P_x / f &= {}^C P_x / {}^W P_z & \text{или} & \quad {}^F P_x = (f / {}^C P_z) {}^C P_x \\ {}^F P_y / f &= {}^C P_y / {}^W P_z & \text{или} & \quad {}^F P_y = (f / {}^C P_z) {}^C P_y \end{aligned} \quad (13.15)$$

Перспективное преобразование, показанное на рис. 13.10, определено с использованием единственного параметра — фокусного расстояния  $f$ . В уравнении (13.16) используется матрица  ${}^F C \Pi(f)$  размера  $4 \times 4$  — в таком виде ее можно



**Рис. 13.10.** Если начало системы координат камеры располагается в центре проектирования, то всегда  ${}^F P_z = f$ .

использовать в комбинации с матрицами других аффинных преобразований. Третья строка, которая дает  ${}^F P_z = f$ , в действительности не нужна, она полностью игнорируется и часто не записывается. Обратите внимание, что ранг этой матрицы равен 3, а не 4 — значит, у нее нет обратной матрицы.

$${}^F \mathbf{P} = {}^F \mathbf{C} \Pi(f) {}^C \mathbf{P}$$

$$\begin{bmatrix} s & {}^F P_x \\ s & {}^F P_y \\ s & {}^F P_z \\ s & \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix} \begin{bmatrix} {}^C P_x \\ {}^C P_y \\ {}^C P_z \\ 1 \end{bmatrix} \quad (13.16)$$

Перспективное преобразование можно определить иначе, если поместить начало системы координат камеры так, чтобы было  ${}^F P_z = 0$  (рис. 13.11). Матрица перспективной проекции тогда примет вид (13.17). Преимуществом данного способа записи является то, что при  $f \rightarrow \infty$  очевидно приближение перспективной проекции к прямоугольной.

$${}^F \mathbf{P} = {}^F \mathbf{C} \Pi(f) {}^C \mathbf{P}$$

$$\begin{bmatrix} s & {}^F P_x \\ s & {}^F P_y \\ s & {}^F P_z \\ s & \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1/f & 1 \end{bmatrix} \begin{bmatrix} {}^C P_x \\ {}^C P_y \\ {}^C P_z \\ 1 \end{bmatrix} \quad (13.17)$$

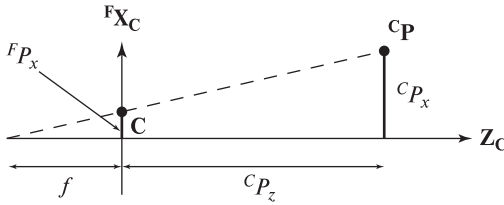
В общем случае (рис. 13.3) система координат камеры отличается от мировой системы координат  $\mathbf{W}$ . Для преобразования мировых координат точки  ${}^W \mathbf{P}$  в координаты камеры  ${}^C \mathbf{P}$  требуется выполнить поворот и перенос. Для описания этих преобразований необходимы три параметра поворота и три параметра переноса. Они комбинируются сложным способом и в результате дают элементы матрицы преобразования (как описано в предыдущих разделах).

$${}^C \mathbf{P} = \mathbf{T}(t_x, t_y, t_z) \mathbf{R}(\alpha, \beta, \gamma) {}^W \mathbf{P}$$

$${}^C \mathbf{P} = {}^C \mathbf{W} \mathbf{T} \mathbf{R}(\alpha, \beta, \gamma, t_x, t_y, t_z) {}^W \mathbf{P}$$

$$\begin{bmatrix} {}^C P_x \\ {}^C P_y \\ {}^C P_z \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^W P_x \\ {}^W P_y \\ {}^W P_z \\ 1 \end{bmatrix} \quad (13.18)$$

Можно сформировать композицию преобразований для перехода из системы координат  $\mathbf{W}$  в систему  $\mathbf{C}$  с последующим перспективным преобразованием точек  ${}^C \mathbf{P}$  в точки на плоскости изображения  ${}^F \mathbf{P}$ . Третья строка матрицы преобразования может быть отброшена, так как она дает постоянное значение для  ${}^F P_z$ . Координаты точки  ${}^F \mathbf{P}$  на плоскости изображения являются действительными числами, поэтому для преобразования этих значений в номера столбца и строки соответствующего пиксела требуется применить преобра-



**Рис. 13.11.** Если начало системы координат камеры располагается в центре изображения, то всегда  ${}^F P_z = 0$ .

зование масштабирования, которое даст координаты точки изображения  ${}^I \mathbf{P}$ . Напомним, что умножение матриц, которое можно рассматривать как последовательное применение линейных преобразований векторного пространства, является ассоциативным.

$$\begin{aligned} {}^F \mathbf{P} &= {}^F \mathbf{C} \Pi(f) {}^C \mathbf{P} \\ &= {}^F \mathbf{C} \Pi(f) ({}^C \mathbf{W} \mathbf{TR}(\alpha, \beta, \gamma, t_x, t_y, t_z) {}^W \mathbf{P}) \\ &= ({}^F \mathbf{C} \Pi(f) {}^C \mathbf{W} \mathbf{TR}(\alpha, \beta, \gamma, t_x, t_y, t_z)) {}^W \mathbf{P} \end{aligned}$$

$$\begin{bmatrix} s {}^F P_x \\ s {}^F P_y \\ s \end{bmatrix} = \begin{bmatrix} d_{11} & d_{12} & d_{13} & d_{14} \\ d_{21} & d_{22} & d_{23} & d_{24} \\ d_{31} & d_{32} & d_{33} & 1 \end{bmatrix} \begin{bmatrix} {}^W P_x \\ {}^W P_y \\ {}^W P_z \\ 1 \end{bmatrix} \quad (13.19)$$

Матричные элементы в (13.19) обозначены как  $d_{ij}$ , а не  $c_{ij}$ , так как приведенная в этом уравнении матрица еще не является матрицей камеры, вывод которой является целью настоящего раздела. Это объясняется тем, что в данном уравнении пока используются единицы измерения мировой системы координат, например, миллиметры или дюймы, и в него не входит масштабирование точек изображения для получения номеров строк и столбцов пикселей на плоскости изображения. Масштабные множители для перехода от миллиметров к номерам строк/столбцов пикселей изображения легко добавить в (13.19) и получить таким образом матрицу камеры  $\mathbf{C}$ . Предположим, что  $d_x$  и  $d_y$  являются размерами (шириной и высотой) пиксела изображения в единицах мировой системы координат. Вместо вещественных координат  $[{}^F P_x, {}^F P_y]$ , заданных системе координат изображения с началом  $(0, 0)$  в левом нижнем углу изображения, нам требуется перейти к целочисленным координатам  $[r, c]$ , которые являются номерами строк и столбцов пикселей изображения, а начало координат располагается в левом верхнем пикселе изображения. Преобразование вещественных координат в пиксельные координаты, с учетом изменения направления вертикальной оси, выражается так:

$${}^I \mathbf{P} = \begin{bmatrix} sr \\ sc \\ s \end{bmatrix} = {}^I \mathbf{F} \mathbf{S} \begin{bmatrix} s {}^F P_x \\ s {}^F P_y \\ s \end{bmatrix} \quad (13.20)$$

где преобразование  ${}^{\mathbf{I}}_{\mathbf{F}}\mathbf{S}$  имеет вид:

$${}^{\mathbf{I}}_{\mathbf{F}}\mathbf{S} = \begin{bmatrix} 0 & -\frac{1}{d_y} & 0 \\ \frac{1}{d_x} & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (13.21)$$

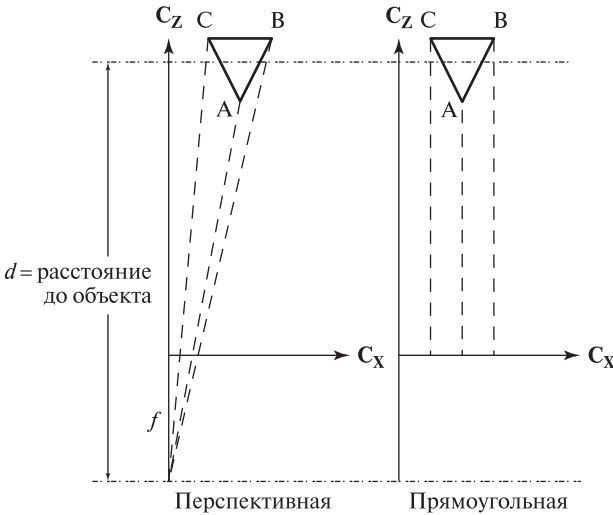
Окончательный результат — полная матрица камеры, выполняющая преобразование трехмерных координат точек мировой системы координат в пиксельные координаты изображения — имеет вид:

$${}^{\mathbf{I}}_{\mathbf{P}} = ({}^{\mathbf{I}}_{\mathbf{F}}\mathbf{S} \mathbf{F} \Pi(f) \mathbf{C} \mathbf{W} \mathbf{TR}(\alpha, \beta, \gamma, t_x, t_y, t_z)) \mathbf{W} \mathbf{P}$$

$$\begin{bmatrix} s^I P_r \\ s^I P_c \\ s \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & 1 \end{bmatrix} \begin{bmatrix} W P_x \\ W P_y \\ W P_z \\ 1 \end{bmatrix} \quad (13.22)$$

Это и есть матрица камеры из уравнения (13.14).

Кратко повторим, что было проделано выше для моделирования восприятия трехмерного мира датчиком изображений — камерой. Во-первых, мы поместили камеру так, чтобы ее система координат совпала с мировой ( $\mathbf{W}$ ). Затем мы повернули камеру ( $\mathbf{W}'\mathbf{R}$ ) в ее окончательное положение относительно системы  $\mathbf{W}$ . Потом камера была перенесена ( $\mathbf{C} \mathbf{W}'\mathbf{T}$ ) для того, чтобы рабочее пространство было видно с соответствующей точки зрения. Теперь все трехмерные точки могут быть спроектированы на плоскость изображения камеры с помощью перспективного преобразования ( $\mathbf{F} \Pi(\mathbf{f})$ ). Наконец, после масштабирования вещественных координат на плоскости изображения  $[{}^F P_x, {}^F P_y]$  и изменения направления вертикальной оси были получены пиксельные координаты  $[{}^i P_r, {}^i P_c]$ . Для описания всех перечисленных шагов была использована единая форма обозначений преобразований. На практике обычно сложно с использованием измерений длин и углов выполнить подобную процедуру с удовлетворительной точностью, чтобы получить матрицу камеры  $\mathbf{C}$  для использования в вычислениях. Вместо этой процедуры для получения матрицы камеры обычно выполняется процедура *калибровки камеры*. Форма матрицы камеры поясняется приведенными выше рассуждениями, а конкретные значения ее элементов вычисляются согласно процедуре подбора для опорных точек, которая будет описана ниже. Перед рассмотрением калибровки продемонстрируем несколько важных случаев использования матрицы камеры, которая вычисляется в процессе калибровки.



**Рис. 13.12.** Сравнение перспективной (слева) и прямоугольной (справа) проекций.

### Упражнение 13.11

Из материала данного раздела легко увидеть, что матрица камеры имеет следующую форму:

$$\begin{bmatrix} s^I P_r \\ s^I P_c \\ s \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & c_{34} \end{bmatrix} \begin{bmatrix} W P_x \\ W P_y \\ W P_z \\ 1 \end{bmatrix} \quad (13.23)$$

Покажите, что 11 элементов матрицы в уравнении (13.22) можно получить из матрицы (13.23) путем масштабирования матрицы камеры на  $1/c_{34}$ . Проверьте, что эти 11 элементов выполняют то же самое преобразование трехмерных точек мировой системы координат в двумерные точки изображения.

### 13.3.2 Прямоугольная и слабая перспективная проекции

При прямоугольном (ортографическом) проектировании координаты точки  $C_P$  получаются просто путем отбрасывания координаты  $z$  соответствующей точки в мировой системе координат. То есть проектирование всех точек мировой системы выполняется вдоль направления, параллельного оптической оси камеры. Прямоугольная и перспективная проекции сравниваются на рис. 13.12. Прямоугольную проекцию можно представить как перспективную проекцию с бесконечно большим фокусным расстоянием (13.24). Прямоугольная проекция часто применяется в компьютерной графике для правильного отображения поперечных сечений объектов. Она также используется в теоретических построениях в компьютерном зрении — так как ее выражение проще, чем у

перспективной проекции, то во многих случаях ее можно считать достаточно хорошим приближением для проверки теоретических моделей.

$$\begin{aligned} {}^F\mathbf{P} &= {}^F\mathbf{C}\Pi(\infty) {}^C\mathbf{P} \\ \begin{bmatrix} {}^FP_x \\ {}^FP_y \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} {}^CP_x \\ {}^CP_y \\ {}^CP_z \\ 1 \end{bmatrix} \end{aligned} \quad (13.24)$$

Часто перспективную проекцию можно приближенно представить как прямоугольную проекцию с последующим однородным масштабированием вещественной плоскости изображения. Эта приближенная модель была названа *слабой перспективной проекцией* (weak perspective). В качестве масштабного множителя можно взять отношение расстояния до объекта к фокусному расстоянию камеры ( $s = f/d$  на рис. 13.12).

$$\begin{aligned} {}^F\mathbf{P} &= {}^F\mathbf{C}\Pi(s) {}^C\mathbf{P} \\ \begin{bmatrix} {}^FP_x \\ {}^FP_y \end{bmatrix} &= \begin{bmatrix} s & 0 & 0 & 0 \\ 0 & s & 0 & 0 \end{bmatrix} \begin{bmatrix} {}^CP_x \\ {}^CP_y \\ {}^CP_z \\ 1 \end{bmatrix} \end{aligned} \quad (13.25)$$

Известен эмпирический способ оценки пригодности модели слабой перспективной проекции, согласно которому расстояние до объекта должно примерно в 20 раз превышать его размер. Точность приближения также зависит от расстояния объекта от оптической оси камеры — чем ближе он расположен, тем выше точность. Если треугольный объект (рис. 13.12) будет передвигаться еще дальше вправо от оптической оси, то перспективные проекции точек А и В будут сближаться, пока образ точки В не закроет образ точки А. Но при прямоугольной проекции расстояние между образами точек А и В будет оставаться постоянным. Во многих робототехнических и промышленных зрительных системах предусмотрено аппаратное центрирование объекта интереса в поле зрения. Это также обычно обеспечивается при выполнении аэрофото-съемок. В подобных случаях слабая перспективная проекция может оказаться подходящей моделью для описания процесса формирования изображений.

### Упражнение 13.12

Получите уравнение (13.24) из (13.16) при  $f \rightarrow \infty$ .

Математические модели и алгоритмы обычно легче разрабатывать с использованием слабой перспективной проекции, а не настоящей перспективной проекции. Приближенные результаты обычно оказываются удовлетворительными для задач распознавания объектов. Кроме того, аналитическое решение, полученное для слабой перспективной проекции, может быть использовано

**Таблица 13.1.** Сравнение перспективной и слабой перспективной проекций

${}^w P_x$	$f = 5 \text{ мм}$	$s = 5/1000$	$f = 20 \text{ мм}$	$s = 20/1000$	$f = 50 \text{ мм}$	$s = 50/1000$
0	0.000	0.000	0.000	0.000	0.000	0.000
10	0.051	0.050	0.204	0.200	0.510	0.500
20	0.102	0.100	0.408	0.400	1.020	1.000
50	0.255	0.250	1.020	1.000	2.551	2.500
100	0.510	0.500	2.041	2.000	5.102	5.000
200	1.020	1.000	4.082	4.000	10.204	10.000
500	2.551	2.500	10.204	10.000	25.510	25.000
1000	5.102	5.000	20.408	20.000	51.020	50.000

Значения  ${}^i P_x$  для трехмерных точек  $[{}^c P_x, 0, 980]^t$  вычислены с помощью перспективного преобразования  $\mathbf{F}_C \Pi_{\text{pers}}(\mathbf{f})$  и с помощью слабого перспективного преобразования  $\mathbf{F}_C \Pi_{\text{weak}}(\mathbf{s})$  при масштабном множителе  $s = f/1000$ . Фокусное расстояние принято равным 5, 20 и 50 мм. Масштабный множитель слабой перспективной проекции выбран  $f/1000$  для расстояния до объекта, равного 1000 мм.

в качестве начальной точки для более сложного итерационного алгоритма, в котором используется настоящая перспективная проекция. Этот вопрос подробно рассматривается в работе [Huttenlocher, Ullman, 1988]. В табл. 13.1 представлены численные данные для сравнения настоящей и слабой перспективной проекции: эти данные свидетельствуют о хорошем приближении в используемом диапазоне параметров.

Выполнив подстановку в определение в форме уравнения (13.25), можно получить, что слабая перспективная проекция задается 8-ю параметрами:

$$\mathbf{F}_C \mathbf{P} = \mathbf{F}_C \Pi_{\text{weak}} \quad \mathbf{C}_W \mathbf{T} \mathbf{R} \quad \mathbf{W}_P$$

$$\begin{bmatrix} {}^F P_x \\ {}^F P_y \end{bmatrix} = \begin{bmatrix} s & 0 & 0 & 0 \\ 0 & s & 0 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^W P_x \\ {}^W P_y \\ {}^W P_z \\ 1 \end{bmatrix} \quad (13.26)$$

$$= \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \end{bmatrix} \begin{bmatrix} {}^W P_x \\ {}^W P_y \\ {}^W P_z \\ 1 \end{bmatrix} \quad (13.27)$$

### Упражнение 13.13

Хотя в (13.27) показаны 8 параметров слабого перспективного преобразования, только 7 из них являются независимыми. Укажите их.

### 13.3.3 Вычисление трехмерных координат с использованием нескольких камер

В данном разделе описывается использование модели зрительной системы с двумя камерами для вычисления трехмерных мировых координат точки  $[x, y, z]$  по двумерным пиксельным координатам двух образов этой точки  $[r_1, c_1]$  и  $[r_2, c_2]$ , которые были измерены с помощью двух калиброванных камер. Так как теперь уже назначение координатных систем очевидно, то будем опускать соответствующие индексы в обозначениях точек. На рис. 13.3 показана схема рассматриваемой зрительной системы, а уравнение (13.14) определяет модель каждой камеры. Следующие уравнения, описывающие процесс получения изображений, являются двумя системами из 4 линейных уравнений от 3 неизвестных  $x, y$  и  $z$ .

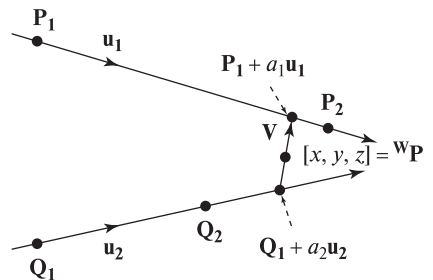
$$\begin{bmatrix} sr_1 \\ sc_1 \\ s \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} tr_2 \\ tc_2 \\ t \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (13.28)$$

Исключая из (13.28) однородные координаты  $s$  и  $t$ , получим следующие четыре линейных уравнения от трех неизвестных.

$$\begin{aligned} r_1 &= (b_{11} - b_{31}r_1)x + (b_{12} - b_{32}r_1)y + (b_{13} - b_{33}r_1)z + b_{14} \\ c_1 &= (b_{21} - b_{31}c_1)x + (b_{22} - b_{32}c_1)y + (b_{23} - b_{33}c_1)z + b_{24} \\ r_2 &= (c_{11} - c_{31}r_2)x + (c_{12} - c_{32}r_2)y + (c_{13} - c_{33}r_2)z + c_{14} \\ c_2 &= (c_{21} - c_{31}c_2)x + (c_{22} - c_{32}c_2)y + (c_{23} - c_{33}c_2)z + c_{24} \end{aligned} \quad (13.29)$$

Для вычисления координат  $[x, y, z]$  достаточно решить систему любых трех из четырех приведенных уравнений. Однако разные подмножества системы (13.29) будут давать немного разные значения координат  $(x, y, z)$ . Все четыре уравнения обычно оказываются несовместными, что объясняется ошибками в



**Рис. 13.13.** Наименьшее расстояние между скрещивающимися прямыми измеряется вдоль отрезка прямой, перпендикулярной к обеим скрещивающимся прямым.



определении параметров камеры и в измеренных координатах точек. То есть два проектирующих луча в математическом трехмерном пространстве оказываются непересекающимися «с абсолютной точностью». Более корректное решение этой проблемы основано на определении «сближения» проектирующих лучей (это скрещивающиеся прямые), т. е. вычислении длины наименьшего отрезка, соединяющего лучи. Если его длина оказывается «достаточно малой», то можно принять середину этого отрезка за точку пересечения проектирующих лучей, т. е. за точку  $[x, y, z]$  (см. рис. 13.13). Если этот отрезок оказывается слишком длинным, то будем полагать, что при определении координат соответствующих точек  $[r_1, c_1]$  и  $[r_2, c_2]$  имеются ошибки.

Точки  $\mathbf{P}_1$  и  $\mathbf{P}_2$  лежат на одной прямой (13.13), а  $\mathbf{Q}_1$  и  $\mathbf{Q}_2$  — на другой. Единичные векторы  $\mathbf{u}_1$  и  $\mathbf{u}_2$  являются направляющими векторами этих прямых. Вектор  $\mathbf{V} = \mathbf{P}_1 + a_1\mathbf{u}_1 - (\mathbf{Q}_1 + a_2\mathbf{u}_2)$  является кратчайшим (свободным) вектором, соединяющим две прямые. Скалярные множители  $a_1$  и  $a_2$  надо определить. Это можно сделать путем вычислений, минимизирующих длину  $\mathbf{V}$ , но проще учесть тот факт, что вектор  $\mathbf{V}$  должен быть перпендикулярен обоим векторам  $\mathbf{u}_1$  и  $\mathbf{u}_2$ .

Применяя это ограничение перпендикулярности, можно записать два линейных уравнения от двух неизвестных  $a_1$  и  $a_2$ .

$$\begin{aligned} ((\mathbf{P}_1 + a_1\mathbf{u}_1) - (\mathbf{Q}_1 + a_2\mathbf{u}_2)) \circ \mathbf{u}_1 &= 0 \\ ((\mathbf{P}_1 + a_1\mathbf{u}_1) - (\mathbf{Q}_1 + a_2\mathbf{u}_2)) \circ \mathbf{u}_2 &= 0 \end{aligned} \quad (13.30)$$

$$\begin{aligned} 1a_1 - (\mathbf{u}_1 \circ \mathbf{u}_2)a_2 &= (\mathbf{Q}_1 - \mathbf{P}_1) \circ \mathbf{u}_1 \\ (\mathbf{u}_1 \circ \mathbf{u}_2)a_1 - 1a_2 &= (\mathbf{Q}_1 - \mathbf{P}_1) \circ \mathbf{u}_2 \end{aligned} \quad (13.31)$$

Уравнения (13.32) легко решить относительно  $a_1$  и  $a_2$  исключением переменных или путем вычисления определителей. В результате получим:

$$\begin{aligned} a_1 &= \frac{(\mathbf{Q}_1 - \mathbf{P}_1) \circ \mathbf{u}_1 - ((\mathbf{Q}_1 - \mathbf{P}_1) \circ \mathbf{u}_2) \circ (\mathbf{u}_1 \circ \mathbf{u}_2)}{1 - (\mathbf{u}_1 \circ \mathbf{u}_2)^2} \\ a_2 &= \frac{((\mathbf{Q}_1 - \mathbf{P}_1) \circ \mathbf{u}_1)(\mathbf{u}_1 \circ \mathbf{u}_2) - (\mathbf{Q}_1 - \mathbf{P}_1) \circ \mathbf{u}_2}{1 - (\mathbf{u}_1 \circ \mathbf{u}_2)^2} \end{aligned} \quad (13.32)$$

Проверив, что  $\|s\mathbf{V}\|$  меньше некоторого порога, мы можем заключить, что два проектирующих луча пересекаются в точке  $[x, y, z]^t = (1/2)[(\mathbf{P}_1 + a_1\mathbf{u}_1) + (\mathbf{Q}_1 + a_2\mathbf{u}_2)]$ . Сейчас важно вернуться к началу раздела и напомнить, что все приведенные вычисления зависят от определения каждого проектирующего луча, т. е. от возможности указания пары точек (например,  $\mathbf{P}_1$  и  $\mathbf{P}_2$ ) на каждом луче. Часто каждый луч задается оптическим центром камеры и точкой на изображении. Если оптический центр неизвестен, точку на проектирующем луче первой камеры можно найти, выбрав некоторое значение  $z = z_1$  и решив (13.29) относительно  $x$  и  $y$ . Если луч близок к параллельности оси  $z$ , то следует выбрать значение  $x = x_0$ . Аналогичным способом можно выбрать все четыре точки на скрещивающихся прямых.

---

**Упражнение 13.14**

На любом языке программирования реализуйте функцию для вычисления расстояния между двумя скрещивающимися прямыми и координат средней точки отрезка, соединяющего эти прямые. Функция должна в качестве параметров получать трехмерные координаты четырех точек и производить вычисления, описанные в данном разделе.

---

**Упражнение 13.15**

Чтобы для решения уравнений (13.31) относительно  $a_1$  и  $a_2$  использовать метод Крамера, необходимо, чтобы определитель матрицы коэффициентов был ненулевым. Докажите, что он будет ненулевым, если в обеих камеры берется изображение одной и той же точки.

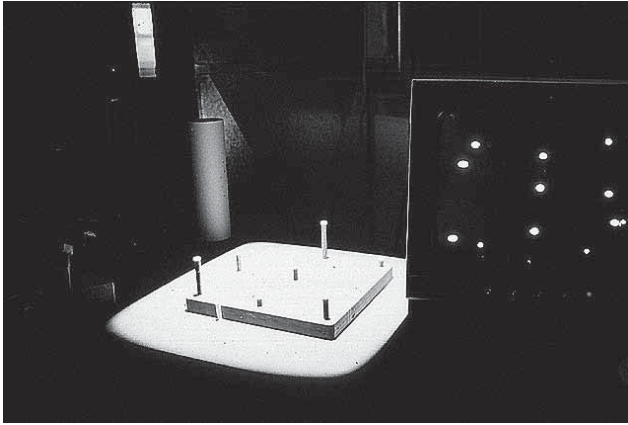
Для проведения трехмерных измерений вместо двух камер можно использовать систему из одной камеры и направленного осветителя. Геометрические и математические выкладки остаются в точности такими же, как и для системы из двух камер. Большим преимуществом стереоскопической системы с осветителем является то, что с его помощью можно искусственным образом формировать текстуру на гладкой поверхности, так что будет легче найти точки, для которых надо выполнять трехмерные измерения координат. Использование структурной подсветки будет рассмотрено ниже, после описания процедуры калибровки для получения параметров модели камеры (или модели направленного осветителя).

**13.4 НАИЛУЧШАЯ АФФИННАЯ КАЛИБРОВОЧНАЯ МАТРИЦА**

Задача калибровки камеры заключается в том, чтобы определить, как расположены пиксели на плоскости изображения данной камеры относительно точек трехмерного пространства, изображения которых и требуется получать с помощью камеры. Процедура калибровки обычно предшествует другим процедурам обработки изображений, в которых используются вычисления трехмерных координат и ориентации объектов в трехмерном пространстве, или производятся измерения размеров трехмерных объектов. Калибровка также необходима для триангуляционного вычисления расстояний (которая была описана в разделе 13.3.3).

В разделе 13.3 было показано, что матрица камеры с 11 параметрами из уравнения (13.14) является искомой математической моделью камеры. Теперь покажем, как можно определить значения этих 11 параметров с помощью метода наименьших квадратов. Положение и фокусное расстояние камеры будем считать фиксированными. На наблюдаемой сцене расположен специальный объект — *калибровочный стенд*, характерные размеры которого известны (см.

рис. 13.14). Затем вводится множество из  $n$  пар точек  $\langle {}^I\mathbf{P}_j, {}^W\mathbf{P}_j \rangle$ : пиксельные координаты  ${}^I\mathbf{P}_j = [{}^IP_r, {}^IP_c]$  заданы на плоскости изображения, а трехмерные координаты  ${}^W\mathbf{P}_j = [{}^WP_x, {}^WP_y, {}^WP_z]$  — в мировой системе координат. Число точек  $n$  должно быть не меньше 6, но наилучшие результаты получаются при  $n \geq 25$ .



**Рис. 13.14.** На калибровочном стенде (слева) расположены 9 штырьков различной высоты (25 различных точек для калибровки можно получить, повернув стенд в 3 различных положения). Изображение стенда показано в правой части рисунка.

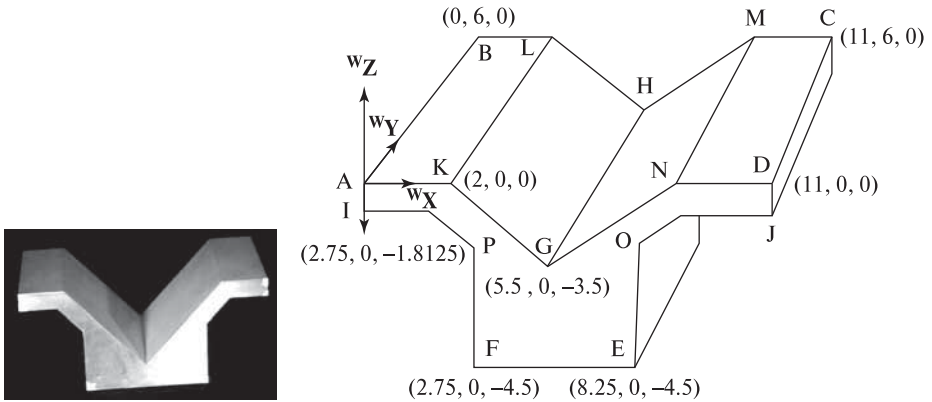
### 13.4.1 Калибровочный стенд

*Калибровочный стенд* нужен для обеспечения набора точек с известными трехмерными координатами. На рис. 13.14, 13.15 и 13.22 показаны три различных стенда. Стенд требуется точно разместить в мировой системе координат  $\mathbf{W}$ . Можно даже задать мировую систему координат с его помощью — тогда сразу будут известны трехмерные координаты калибровочных точек  $[{}^WP_x, {}^WP_y, {}^WP_z]$ . Затем с помощью камеры вводится изображение стенда и определяются двумерные пиксельные координаты калибровочных точек  $[{}^IP_r, {}^IP_c]$ . В качестве калибровочных стендов также используются жесткие проволочные объекты или твердые пластины с метками специальной формы.

### 13.4.2 Задача для решения методом наименьших квадратов

Уравнение (13.33) получено путем исключения однородного масштабного множителя  $s$  из модели процесса получения изображений. Теперь у нас есть два линейных уравнения, описывающих геометрические свойства каждого проектирующего луча, и для каждой калибровочной точки имеется один проектирующий луч. Для упрощения обозначений и чтобы избежать путаницы с символами, будем обозначать координаты точки в мировой системе  ${}^W\mathbf{P}_j =$

Координаты остальных углов можно определить на основе свойств симметрии детали



**Рис. 13.15.** В качестве калибровочного стенда можно использовать точно изготовленную деталь с большим количеством углов. Показанная деталь имеет длину 11 дюймов, ширину 6 дюймов и высоту 4.5 дюйма. Все трехмерные координаты углов приведены на рис. 13.18.

$[{}^W P_x, {}^W P_y, {}^W P_z]$  как  $[x_j, y_j, z_j]$ ; а пиксельные координаты точки  ${}^I P_j = [{}^I P_r, {}^I P_c]$  как  $[u_j, v_j]$ . Для каждой калибровочной точки можно записать два уравнения:

$$\begin{aligned} u_j &= (c_{11} - c_{31}u_j)x_j + (c_{12} - c_{32}u_j)y_j + (c_{13} - c_{33}u_j)z_j + c_{14} \\ v_j &= (c_{21} - c_{31}v_j)x_j + (c_{22} - c_{32}v_j)y_j + (c_{23} - c_{33}v_j)z_j + c_{24} \end{aligned} \quad (13.33)$$

Эти уравнения можно переписать так, чтобы разделить известные и неизвестные величины в отдельные векторы. Все величины в левой части являются известными (из пар калибровочных точек), а в правой части все величины  $c_{km}$  являются неизвестными и их необходимо определить.

$$\begin{bmatrix} x_j & y_j & z_j & 1 & 0 & 0 & 0 & 0 & -x_j u_j & -y_j u_j & -z_j u_j \\ 0 & 0 & 0 & 0 & x_j & y_j & z_j & 1 & -x_j v_j & -y_j v_j & -z_j v_j \end{bmatrix} \begin{bmatrix} c_{11} \\ c_{12} \\ c_{13} \\ c_{14} \\ c_{21} \\ c_{22} \\ c_{23} \\ c_{24} \\ c_{31} \\ c_{32} \\ c_{33} \end{bmatrix} = \begin{bmatrix} u_j \\ v_j \end{bmatrix} \quad (13.34)$$

Каждый проектирующий луч дает два таких уравнения, таким образом для  $n$  калибровочных точек мы получаем  $2n$  линейных уравнений. Эти уравнения кратко можно представить в матричной форме, где  $\mathbf{x}$  — это вектор-столбец неизвестных, а  $\mathbf{b}$  — вектор-столбец координат на изображении.

$$\mathbf{A}_{2n \times 11} \mathbf{x}_{11 \times 1} \approx \mathbf{b}_{2n \times 1} \quad (13.35)$$

Так как в системе уравнений есть 11 неизвестных, а количество уравнений — 12 и более, то данная система является переопределенной. Не существует такого вектора  $\mathbf{x}$ , который являлся бы решением всех уравнений. Для нахождения наилучшего решения в данном случае подходит метод наименьших квадратов. С его помощью надо определить набор неизвестных параметров, для которых оказывается минимальной сумма (по всем уравнениям) квадратов разностей наблюдаемых координат и координат, вычисленных с помощью матрицы камеры. На рис. 13.16 показаны четыре такие разности для двух калибровочных точек. Будем называть их *невязками* (как и в гл. 11). На рис. 13.17 показано абстрактное представление решения методом наименьших квадратов: мы хотим вычислить коэффициенты  $c_{km}$ , которые в линейной комбинации со столбцами матрицы  $\mathbf{A}$  оказываются наиболее близкими к  $\mathbf{b}$ . Для нахождения этого решения можно заметить, что вектор невязок  $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}$  ортогонален столбцам матрицы  $\mathbf{A}$ , т. е.  $\mathbf{A}^t \mathbf{r} = \mathbf{0}$ . Подставляя сюда  $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}$ , получим  $\mathbf{A}^t \mathbf{A}\mathbf{x} = \mathbf{A}^t \mathbf{b}$ . Матрица  $\mathbf{A}^t \mathbf{A}$  является симметричной и положительно определенной, поэтому у нее есть обратная матрица. Ее можно использовать для решения и нахождения  $\mathbf{x} = (\mathbf{A}^t \mathbf{A})^{-1} \mathbf{A}^t \mathbf{b}$ . Для решения этого уравнения существует ряд готовых библиотек численных методов. (С помощью пакета MATLAB решение можно получить просто оператором  $\mathbf{X} = \mathbf{A} \setminus \mathbf{B}$ . После вычисления методом наименьших квадратов решения  $\mathbf{X}$ , можно найти вектор невязок  $\mathbf{R} = \mathbf{B} - \mathbf{A}\mathbf{X}$ . Справочную информацию см. в учебнике по линейной алгебре [Heath, 1997].)

На рис. 13.18 приведены результаты вычисления матрицы камеры для калибровочного стенда, показанного на рис. 13.15. Углы объекта-стенда помечены буквами от 'A' до 'P', их мировые координаты  $[X, Y, Z]$  перечислены на рис. 13.18 вместе с соответствующими пиксельными координатами  $[U, V]$ . Угловые точки 'B', 'C' и 'M' при выбранной ориентации стенда не видны, поэтому для них нет пиксельных координат. Матрица камеры, полученная совмещением 13 соответствующих точек, показана в нижней части рис. 13.18, в правой части приведены невязки. Из 26 координат, вычисленных с использованием матрицы камеры, 16 координат совпадают с известными трехмерными координатами с точностью менее 1 пиксела, 10 координат — с точностью более пиксела, но только две координаты отличаются от известных трехмерных больше, чем на 2 пиксела. Приведенный пример подтверждает обоснованность аффинной модели камеры, но также иллюстрирует возникновение ошибок при измерении угловых точек и искажения, имеющиеся у короткофокусного объектива.

**Упражнение 13.16.** Повторение вывода модели камеры

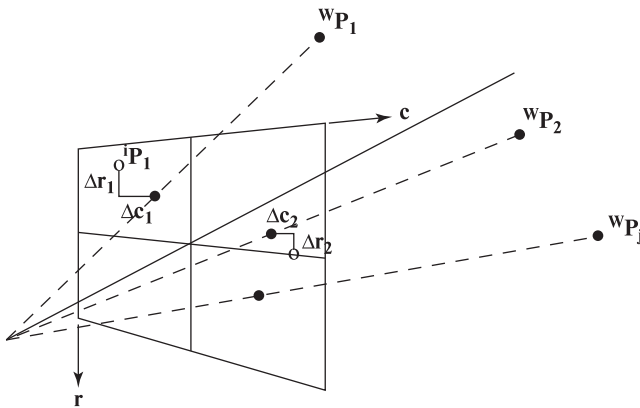
(а) Выберите готовый программный пакет с реализацией метода наименьших квадратов. Введите соответствующие точки с рис. 13.18 и вычислите матрицу камеры. Сравните ее с матрицей, показанной на рис. 13.18. (б) Удалите три точки с максимальными невязками и получите новую матрицу камеры. Есть ли новые невязки, большие 2 пикселей? (в) Задайте трехмерные координаты единичного куба, лежащего на одной из верхних горизонтальных граней объекта-стенда с рис. 13.15. С помощью полученной матрицы камеры преобразуйте трехмерные координаты 8 вершин куба в пиксельные координаты изображения. Также преобразуйте координаты четырех вершин грани, на которой лежит куб. Нарисуйте изображение по вычисленным точкам и соедините эти точки ребрами. Имеет ли смысл полученное изображение куба?

**Упражнение 13.17.** Субпиксельная точность

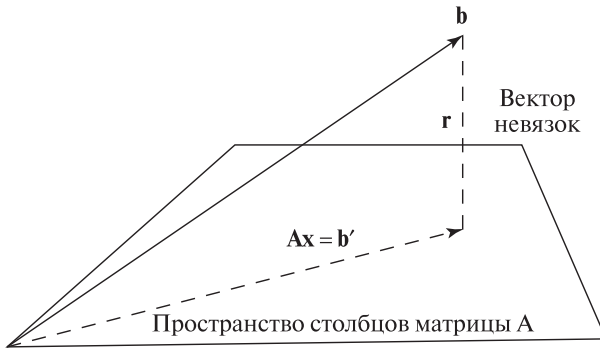
См. рис. 13.14. С помощью методов, рассмотренных в гл. 3, центры штырьков можно определить с точностью менее пиксела. Как именно? Можно ли использовать для калибровки вещественные, а не целочисленные координаты  $[{}^I P_r, {}^I P_c]$ ?

**Упражнение 13.18.** Наилучшая слабopersпективная модель камеры

По данным из рис. 13.18 найдите наилучшую матрицу камеры со слабой перспективной проекцией. Внимательно выполните вывод уравнений, описывающих



**Рис. 13.16.** Невязки на плоскости изображения — это разности между наблюдаемыми точками (обозначены окружностями) и точками, вычисленными с использованием матрицы камеры из уравнения (13.14) (обозначены закрашенными окружностями).



**Рис. 13.17.** Решение, полученное методом наименьших квадратов для системы  $Ax \approx b$ . В виде плоскости изображено 11-мерное пространство столбцов матрицы  $A_{2n \times 11}$ . Все линейные комбинации  $Ax$  должны лежать в этом пространстве, а  $b_{2n \times 1}$  — нет. Метод наименьших квадратов вычисляет  $b'$  как проекцию  $b$  на 11-мерное пространство, выбирая в этом пространстве в качестве  $b'$  точку, ближайшую к  $b$ .

вающих процесс получения изображения — должна получиться новая система уравнений  $Ax = b$ . Сравните значения невязок полученной матрицы камеры с невязками, приведенными в правой части рис. 13.18.

### Упражнение 13.19. Калибровка камеры

В табл. 13.2 приведены трехмерные координаты 16 угловых точек объекта-стенда с рис. 13.15. В действительности, эти координаты измерены на двух независимых изображениях. Вычислите матрицу камеры, используя столбцы 2–6 для 5 любых точек из табл. 13.2.

### Упражнение 13.20. Вычисления для стереосистемы

(а) Используя данные из таблицы 13.2, вычислите две матрицы камеры, одну для столбцов 2–6, а вторую — для столбцов 2–4 и 7–8. (б) Примените метод из раздела 13.3.3 для вычисления координат точки  $A$  с помощью двух матриц камер и пиксельных координат образа этой точки из столбцов 5–8 таблицы 13.2. Сравните полученный результат с координатами в столбцах 2–4 этой таблицы. (в) Рассмотрим тупой угол между угловыми точками  $I$  и  $P$  объекта-стенда (рис. 13.15). Обозначим вершину этого угла буквой  $Q$ . Допустим, эта точка на стереопаре имеет координаты (196, 135) и (281, 237). Примените модель стереосистемы для вычисления трехмерных мировых координат точки  $Q$  и оцените корректность полученного результата.

```

#
# ИЗОБРАЖЕНИЕ: glview1.ras
#
#           ВХОДНЫЕ ДАННЫЕ
#
Точка   Координаты          Трехмерные координаты   |   Совмещенные          Невязки X Y
        на изображении    (X, Y, Z)                |   двумерные точки
        (U, V)
-----|-----
A       95.00  336.00      0.00  0.00  0.00 |   94.53  337.89 |   0.47  -1.89
B                               0.00  6.00  0.00 |
C                               11.00  6.00  0.00 |
D       592.00 368.00     11.00  0.00  0.00 |   592.21 368.36 |  -0.21  -0.36
E       472.00 168.00     8.25  0.00 -4.50 |   470.14 168.30 |   1.86  -0.30
F       232.00 155.00     2.75  0.00 -4.50 |   232.30 154.43 |  -0.30  0.57
G       350.00 205.00     5.50  0.00 -3.50 |   349.17 202.47 |   0.83  2.53
H       362.00 323.00     5.00  6.00 -3.50 |   363.44 324.32 |  -1.44  -1.32
I       97.00  305.00     0.00  0.00 -0.75 |   97.90 304.96 |  -0.90  0.04
J       592.00 336.00     11.00  0.00 -0.75 |   591.78 334.94 |   0.22  1.06
K       184.00 344.00     2.00  0.00  0.00 |   184.46 343.40 |  -0.46  0.60
L       263.00 431.00     2.00  6.00  0.00 |   261.52 429.65 |   1.48  1.35
M                               9.00  6.00  0.00 |
N       501.00 363.00     9.00  0.00  0.00 |   501.16 362.78 |  -0.16  0.22
O       467.00 279.00     8.25  0.00 -1.81 |   468.35 281.09 |  -1.35  -2.09
P       224.00 266.00     2.75  0.00 -1.81 |   224.06 266.43 |  -0.06  -0.43

#
#           КАЛИБРОВОЧНАЯ МАТРИЦА
#
44.84      29.80      -5.504      94.53
2.518      42.24      40.79      337.9
-0.0006832  0.06489    -0.01027    1.000

```

**Рис. 13.18.** Результаты калибровки камеры, полученные для калибровочного стенда с рис. 13.15.

### 13.4.3 Обсуждение аффинного метода

Важный вопрос, который надо рассмотреть — в действительности ли необходимо выполнять оценку 11 параметров модели камеры. Как было показано ранее, для задания положения камеры в мировой системе координат необходимы 3 независимых параметра матрицы поворота и 3 независимых параметра матрицы переноса. Необходимы также 2 масштабных множителя для перехода от вещественных координат изображения к пиксельным и фокусное расстояние  $f$ . Таким образом, не все 11 параметров модели камеры являются независимыми. Если считать их независимыми, то можно считать, что параметры вращения описываются неортономмированной матрицей поворота. В случае качественно изготовленной камеры это ограничение является лишним. Но если плоскость изображения не перпендикулярна оптической оси, то эти лишние параметры дают лучшую модель камеры. Для оценки большего количества параметров требуется больше калибровочных точек, и эти параметры не являются непо-



**Таблица 13.2.** Трехмерные стереоскопические координаты характерных точек калибровочного стенда

Точка	$w_x$	$w_y$	$w_z$	${}^1u$	${}^1v$	${}^2u$	${}^2v$
A	0.0	0.0	0.0	167	65	274	168
B	0.0	6.0	0.0	96	127	196	42
C	11.0	6.0	0.0	97	545	96	431
D	11.0	0.0	0.0	171	517	154	577
E	8.25	0.0	-4.5	352	406	366	488
F	2.75	0.0	-4.5	347	186	430	291
G	5.5	0.0	-3.5	311	294	358	387
H	5.5	6.0	-3.5	226	337	NA	NA
I	0.0	0.0	-0.75	198	65	303	169
J	11.0	0.0	-0.75	203	518	186	577
K	2.0	0.0	0.0	170	143	248	248
L	2.0	6.0	0.0	96	198	176	116
M	9.0	6.0	0.0	97	465	114	363
N	9.0	0.0	0.0	173	432	176	507
O	8.25	0.0	-1.81	245	403	259	482
P	2.75	0.0	-1.81	242	181	318	283

Трехмерные координаты  $w_x$ ,  $w_y$ ,  $w_z$  измерены в дюймах. Пиксельные координаты на первом изображении обозначены  ${}^1u$  и  ${}^1v$ , на втором —  ${}^2u$  и  ${}^2v$ .

средственно внутренними характеристиками камеры. Но у аффинного метода есть ряд положительных свойств — он хорошо работает при наклоне между строками и столбцами изображения или между плоскостью изображения и оптической осью; он работает и с пиксельными, и с вещественными координатами на плоскости изображения; решение легко вычислить без использования итерационных методов. В разделе 13.7 будет представлен иной способ калибровки, в котором принимается больше ограничений и преодолеваются некоторые недостатки аффинного метода.

### Упражнение 13.21. Калибровка бытовой камеры

Возьмите бытовую камеру и жесткую коробку. Нарисуйте несколько символов 'X' на каждой стороне коробки. Измерьте координаты  $(x, y, z)$  каждого угла и каждого символа 'X' относительно модельной системы координат. Начало модельной системы координат совпадает с одним из углов коробки, а оси направлены вдоль ее ребер. Получите изображение коробки. Найдите и отметьте на ней 15 углов и символов 'X'. С помощью линейки измерьте координаты этих точек в миллиметрах. Следуя примеру, рассмотренному в данном разделе, получите матрицу камеры и вычислите невязки. Проверьте, как эта матрица

камеры работает несколькими точками, которые не использовались для ее вычисления. Повторите эксперимент, используя в качестве единиц измерения на плоскости изображения миллиметры, а не пиксели.

### Упражнение 13.22. Искусственное формирование текстуры

Нанесите текстуру на изображение коробки, полученное в процессе выполнения предыдущего упражнения. (а) Во-первых, получите .pgm-файл с изображением коробки. (б) С помощью методов из гл. 11 постройте отображение одной стороны коробки (в двумерных координатах) на массив пикселей вашего изображения. (в) Обновите содержимое .pgm-файла с изображением коробки, записывая в него пиксели из вашего изображения. Подсказка: отображение двух треугольников дает лучшие результаты, чем отображение одного параллелограмма. Почему?

## 13.5 ИСПОЛЬЗОВАНИЕ СТРУКТУРНОЙ ПОДСВЕТКИ

Принцип трехмерного восприятия со *структурной подсветкой* был рассмотрен в первом разделе и проиллюстрирован на рис. 13.4. К настоящему моменту мы обладаем всеми необходимыми математическими инструментами для алгоритмической реализации такого подхода. На рис. 13.19 показана более подробная схема. Поверхности объектов освещаются через шаблон в виде сетки: в данном случае это реализуется с помощью слайд-проектора, проектирующего регулярную сетку из светлых линий на поверхности объектов. Камера при формировании изображений передает результат искажения сетки за счет формы и ориентации поверхности. Так как структура прямоугольной сетки хорошо известна, то зрительная система имеет априорную информацию о том, какие именно проектирующие лучи формируют точки изображения, соответствующие пересечениям линий сетки. Предположим, что известно, что точка сетки  ${}^G P_{lm}$  на изображении видна как точка  ${}^I P_{uv}$ . Тогда имеются четыре уравнения, которые можно использовать для решения и нахождения трехмерных координат точки поверхности  ${}^W P_{lm}$ , которая подсвечена специальным световым шаблоном. Для этого необходимо знать калибровочную матрицу камеры  $\mathbf{C}$  и калибровочную матрицу осветителя  $\mathbf{D}$ . Система уравнений  $\mathbf{D}{}^W P_{lm} = {}^G P_{lm}$  и  $\mathbf{C}{}^W P_{lm} = {}^I P_{uv}$  решается так же, как и раньше (в общем случае для стереосистемы из двух камер, см. раздел 13.3.3).

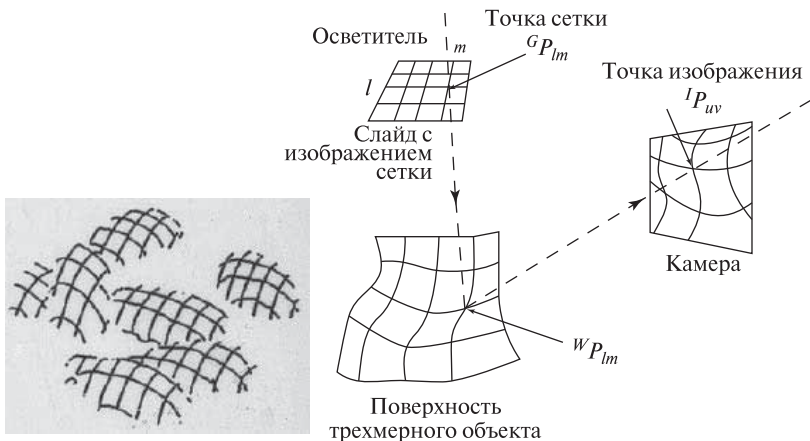
Калибровку осветителя можно произвести аналогично камере. Осветитель включается так, чтобы он освещал плоскую поверхность на рабочем столе. На столе размещаются точно изготовленные блоки, так, чтобы один из углов блока точно попадал в точку пересечения линий сетки осветителя. В такой позиции известны координаты пары точек  $\langle [{}^G P_l, {}^G P_m], [{}^W P_x, {}^W P_y, {}^W P_z] \rangle$ , где  ${}^G P_l, {}^G P_m$  — это целочисленные номера линий сетки, определяющие точку пересечения на сетке, а  ${}^W P_x, {}^W P_y, {}^W P_z$  — измеренные мировые координаты угловой точки

детали-блока. Обращаем внимание при выполнении процедуры аффинной калибровки, что можно простым образом упорядочить линии сетки на слайде проектора с помощью нумерации  $m = 1, 2, \dots$  или  $l = 1, 2, \dots$ , так как эта процедура может применяться с любым масштабным множителем.

### Упражнение 13.23

Сделайте систему со структурной подсветкой и выполните ее калибровку следующим образом. Выберите какой-нибудь графический редактор и нарисуйте в нем изображение светлой сетки на темном фоне (или нарисуйте такой шаблон на бумаге, а потом отсканируйте). Присоедините свой ноутбук к компьютерному проектору и выведите на экран ноутбука изображение сетки. Проектор будет проектировать это изображение в пространство. Расположите проектор так, чтобы он освещал крышку рабочего стола. Поместите на стол несколько объектов известной формы с плоскими гранями, получите калибровочные точки и выполните процедуру аффинной калибровки. Напишите отчет о выполненной работе.

Проблема определения соответствующих точек остается, как и в стереосистеме с двумя камерами, но решение упрощается. Возвращаясь к рис. 13.19, можно сказать, что задачей зрительной системы является определение того, какие именно пересечения линий сетки видны на изображении. Если



**Рис. 13.19.** (Слева) картофелины, освещенные через слайд с изображением сетки. (Справа) принцип действия структурной подсветки: если зрительная система может определить, какой луч  $G_{Plm}$  создал определенный яркий признак в точке  $I_{P_{uv}}$ , то можно записать 4 уравнения для определения трехмерных координат точки поверхности  $W_{Plm}$ . Для этого также должны быть известны матрица камеры и матрица осветителя.

**Алгоритм вычисляет трехмерные координаты точек дискретной сетки путем обработки изображения сцены с искусственно сформированными светлыми линиями.**

**Операции, выполняемые на этапе предварительной обработки:**

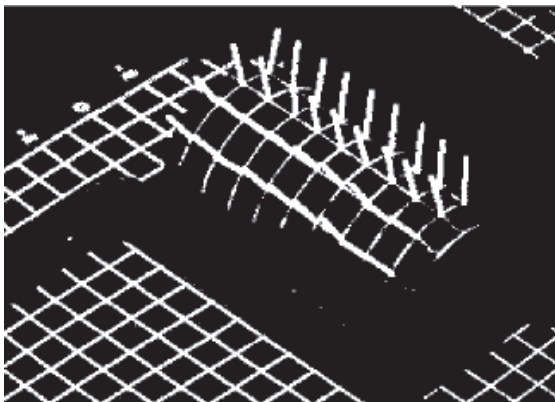
1. Калибровка камеры для получения матрицы камеры  $C$ .
2. Калибровка осветителя для получения матрицы осветителя  $D$ .

**Операции, выполняемые в реальном времени:**

1. Ввод элементов матриц камеры и осветителя  $C$  и  $D$ .
2. Ввод изображения сцены, на котором присутствует поверхность со светлыми линиями сетки на ней.
3. Поиск на изображении сетки из ярких линий и определение точек их пересечения.
4. Вычисление целочисленных координат  $l$  и  $m$  для всех точек пересечения на сетке.
5. С использованием уравнений стереосистемы и матриц  $C$  и  $D$  вычисление трехмерных координат точек поверхности  $P$ , которые на изображении видны как точки  $P_{uv}$ , полученные с помощью проектирующих лучей  $P_{lm}$ .
6. В качестве результатов работы алгоритма возвращается сетка в форме графа трехмерных вершин и соединяющих их ребер (светлых полос на изображении).

**Алгоритм 13.2.** Вычисление трехмерных координат точек поверхности с помощью зрительной системы из одной камеры и одного направленного осветителя структурной подсветки.

требуется определение только формы поверхности, а не ее расположения в пространстве, то обычно достаточно найти только относительное расположение точек пересечения на сетке (относительно друг друга). См. рис. 13.20 и работы [Hu, Stockman, 1989] и [Shrikhande, Stockman, 1989]. Для решения этой проблемы были предложены различные технические решения, например, идентификация линий сетки путем изменения их формы или цвета. Другой вариант — быстро менять форму сетки во времени, и получать с помощью зрительной системы последовательность изображений, по которым можно уникальным образом определить соответствие наблюдаемых точек точкам проектируемого шаблона освещения. Осветители белого цвета имеют ограниченную глубину резкости, на которой сетка видна резко. Лазерные осветители свободны от этого недостатка, но от некоторых материалов от-

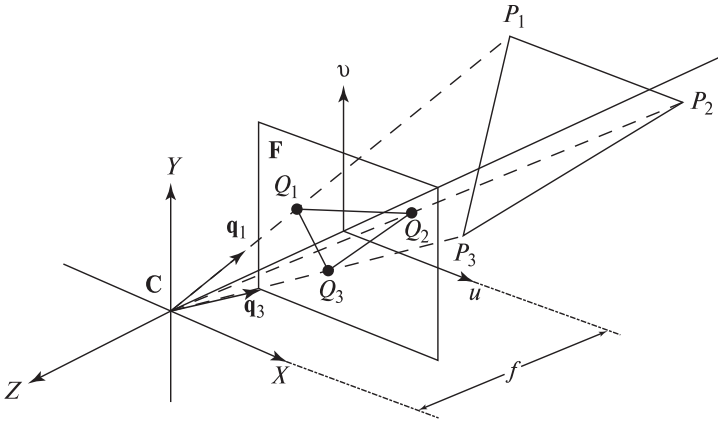


**Рис. 13.20.** Нормали к поверхности, вычисленные с использованием спроектированной на поверхность сетки. Нормали можно вычислить, полагая, что линии сетки упорядочены, хотя точное относительное расположение линий соблюдать не обязательно (для осветителя и для камеры использовалась модель со слабой перспективой). (С любезного разрешения Shrikhande, Stockman.)

ражение получается малым из-за малой мощности и особенностей спектра типичного лазера. Во многих искусственно сформированных сценах лазерная структурная подсветка оказывается очень эффективной. Ряд компаний предлагают готовые модели подобных устройств подсветки — некоторые обеспечивают один луч, некоторые одну или две перпендикулярные светлые полосы.

### 13.6 ПРОСТАЯ ПРОЦЕДУРА ОЦЕНКИ ПОЛОЖЕНИЯ НАБЛЮДАЕМОГО ОБЪЕКТА

Для решения прикладных задач необходимы средства для определения с помощью камер геометрических свойств и ориентации объектов в пространстве. В данном разделе будет рассмотрен простой способ определения ориентации объекта по трем точкам на изображении этого объекта. Будем полагать, что известны геометрическая модель объекта и фокусное расстояние камеры. Основным достоинством описываемого метода является не столько возможность практического применения, сколько рассмотрение ряда важных понятий в простом контексте. Одно из этих понятий — *обратное перспективное преобразование* — относится к вычислению трехмерных свойств на основе свойств двумерных изображений, полученных с использованием перспективной проекции. Еще один важный вопрос — процедура оптимизации для поиска наилучшего множества параметров, обеспечивающих соответствие трехмерных точек объекта и двумерных точек изображения. Самым сильным упрощающим допущением будет то, что мы будем считать мировую систему координат и систему координат камеры совпадающими. Поэтому при обозначении точек будем опускать верхние индексы, поскольку нет необходимости обозначать координатную систему ( ${}^W P_j = {}^C P_j \equiv P_j$ ). Еще одно упрощение — рас-



**Рис. 13.21.** Простой пример задачи оценки ориентации объекта. Треугольник (в пространстве)  $P_1P_2P_3$  со сторонами известной длины на плоскости изображения  $u-v$  виден как треугольник  $Q_1Q_2Q_3$ . Координаты точек  $P_j$  можно вычислить по точкам на изображении  $Q_j$ . После этого можно определить расположение относительно системы координат камеры того объекта, которому принадлежат точки  $P_j$ . Фокусное расстояние  $f$  — это расстояние от центра перспективной проекции  $C$  до плоскости изображения. Относительно системы координат камеры  $C$  точки изображения имеют координаты  $Q_j = [u_j, v_j, -f]$ .

смотрение будем проводить только в геометрическом пространстве с вещественными координатами, без перехода к дискретным пиксельным координатам.

Иллюстрация к *перспективной задаче 3 точек* (Perspective 3 Point Problem, P3P) приведена на рис. 13.21. Три трехмерные точки сцены  $P_i$  в плоскости изображения  $u-v$  видны как точки  $Q_i$ . Координаты точек  $P_i$  неизвестны, и их требуется найти. Мы предположили, что знаем, какие именно точки объекта наблюдаем (сильное допущение), поэтому мы знаем расстояния между парами точек. Для твердого тела эти три расстояния при любых перемещениях тела в пространстве остаются постоянными. В приложении к человеко-машинному интерфейсу точки  $P_i$  могут быть характерными точками на лице оператора, например, глаза и кончик носа. Лицо оператора можно обмерить и получить эти расстояния. Вычислив ориентацию лица, можно понять, куда смотрит оператор. В приложении навигации, три точки  $P_i$  могут быть географическими ориентирами, расположение которых точно известно и задано на карте местности. Управляемый робот или беспилотный летательный аппарат могут определять свое расположение относительно ориентиров с помощью описанного ниже метода.

На изображении положение точек  $Q_i$  известно. Обозначим через  $\mathbf{q}_i$  единичный вектор, направленный из начала координат в точку  $Q_i$ . Точка трехмерного пространства  $P_i$  лежит в том же направлении. Следовательно, координаты  $P_i$  по координатам  $Q_i$  можно вычислить как три скалярные величины  $a_i$  такие, что:

$$P_i = a_i \mathbf{q}_i \quad (13.36)$$

Из трех уравнений системы (13.36) можно записать три других уравнения для расстояний между точками (которые известны из модели).

$$d_{mn} = \|P_m - P_n\| \quad (\text{при } m \neq n) \quad (13.37)$$

Подставим в (13.37) вместо координат  $P_i$  их выражение через измеренные координаты  $Q_i$  и выразим расстояния между точками в трехмерном пространстве, используя скалярное произведение и тот факт, что  $\mathbf{q}_i \circ \mathbf{q}_i = \mathbf{1}$ .

$$\begin{aligned} d_{mn}^2 &= \|a_m \mathbf{q}_m - a_n \mathbf{q}_n\|^2 & (13.38) \\ &= (a_m \mathbf{q}_m - a_n \mathbf{q}_n) \circ (a_m \mathbf{q}_m - a_n \mathbf{q}_n) \\ &= a_m^2 - 2a_m a_n (\mathbf{q}_m \circ \mathbf{q}_n) + a_n^2 \end{aligned}$$

Теперь у нас есть три квадратных уравнения с тремя неизвестными  $a_i$ . Значения в левой части  $d_{mn}$  известны из модели объекта, а три скалярных произведения  $\mathbf{q}_m \circ \mathbf{q}_n$  известны из точек изображения  $Q_i$ . Наша задача РЗР о вычислении трехмерных координат трех точек  $P_i$  оказалась сведена к решению трех квадратных уравнений с тремя неизвестными. Теоретически у системы (13.38) возможны 8 различных решений  $[a_1, a_2, a_3]$ . Расширяя рис. 13.21, можно заметить, что для каждого расположения трех точек с одной стороны координатной системы существует аналогичное зеркально отраженное расположение с другой стороны с координатами  $[-a_1, -a_2, -a_3]$ . Ясно, что если одна тройка координат является решением (13.38), то и вторая тройка тоже будет решением. Следовательно, из восьми теоретически возможных решений в действительности могут возникнуть только четыре, так как объект может располагаться только по одну сторону плоскости изображения. В работе [Fischler, Bolles, 1981] показано, что все четыре местоположения возможны только в особых случаях; а чаще всего наблюдаются только два решения.

Теперь покажем, как решить эту систему относительно  $a_i$  (и, значит, вычислить точки  $P_i$ ) с использованием нелинейной оптимизации. Это рассмотрение поможет понять остальные методы оптимизации, рассматриваемые в последующих разделах. Математически, нам требуется найти три корня

следующих функций от  $a_i$ :

$$\begin{aligned} f(a_1, a_2, a_3) &= a_1^2 - 2a_1a_2(\mathbf{q}_1 \circ \mathbf{q}_2) + a_2^2 - d_{12}^2 \\ g(a_1, a_2, a_3) &= a_2^2 - 2a_2a_3(\mathbf{q}_2 \circ \mathbf{q}_3) + a_3^2 - d_{23}^2 \\ h(a_1, a_2, a_3) &= a_1^2 - 2a_1a_3(\mathbf{q}_1 \circ \mathbf{q}_3) + a_3^2 - d_{13}^2 \end{aligned} \quad (13.39)$$

Предположим, что мы нашли точку в окрестности корня  $[a_1, a_2, a_3]$ , но  $f(a_1, a_2, a_3) \neq 0$ . Мы хотим вычислить небольшие смещения  $[\Delta_1, \Delta_2, \Delta_3]$ , такие, что, в идеале,  $f(a_1 + \Delta_1, a_2 + \Delta_2, a_3 + \Delta_3) = 0$ , а на практике желательно чтобы  $f(a_1 + \Delta_1, a_2 + \Delta_2, a_3 + \Delta_3) \rightarrow 0$ . Мы можем линеаризовать функцию  $f$  в окрестности  $[a_1, a_2, a_3]$  и затем вычислить смещения  $[\Delta_1, \Delta_2, \Delta_3]$ , которые приближают значение  $f$  к 0.

$$f(a_1 + \Delta_1, a_2 + \Delta_2, a_3 + \Delta_3) = f(a_1, a_2, a_3) + \begin{bmatrix} \frac{\partial f}{\partial a_1} & \frac{\partial f}{\partial a_2} & \frac{\partial f}{\partial a_3} \end{bmatrix} \begin{bmatrix} \Delta_1 \\ \Delta_2 \\ \Delta_3 \end{bmatrix} + \mathbf{h.o.t.} \quad (13.40)$$

Слагаемые высоких порядков ( $\mathbf{h.o.t.}$ ) в (13.40) отбрасываем и приравняем левую часть к 0. Таким образом получаем линейное уравнение относительно неизвестных  $[\Delta_1, \Delta_2, \Delta_3]$ . Выполняя аналогичные преобразования к функциям  $g(a_1, a_2, a_3)$  и  $h(a_1, a_2, a_3)$ , приходим к следующему матричному уравнению:

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} f(a_1, a_2, a_3) \\ g(a_1, a_2, a_3) \\ h(a_1, a_2, a_3) \end{bmatrix} + \begin{bmatrix} \frac{\partial f}{\partial a_1} & \frac{\partial f}{\partial a_2} & \frac{\partial f}{\partial a_3} \\ \frac{\partial g}{\partial a_1} & \frac{\partial g}{\partial a_2} & \frac{\partial g}{\partial a_3} \\ \frac{\partial h}{\partial a_1} & \frac{\partial h}{\partial a_2} & \frac{\partial h}{\partial a_3} \end{bmatrix} \begin{bmatrix} \Delta_1 \\ \Delta_2 \\ \Delta_3 \end{bmatrix} \quad (13.41)$$

Матрица частных производных — это якобиан, обозначаемый как  $\mathbf{J}$ . Если он обратим в нашей точке  $[a_1, a_2, a_3]$ , то мы получим следующее решение — искомые смещения для обращения функций  $f$ ,  $g$  и  $h$  в 0.

$$\begin{bmatrix} \Delta_1 \\ \Delta_2 \\ \Delta_3 \end{bmatrix} = -\mathbf{J}^{-1}(a_1, a_2, a_3) \begin{bmatrix} f(a_1, a_2, a_3) \\ g(a_1, a_2, a_3) \\ h(a_1, a_2, a_3) \end{bmatrix} \quad (13.42)$$

Новый вектор мы можем получить, складывая старый с вычисленными смещениями. Обозначим значения параметров  $[a_1, a_2, a_3]$  на  $k$ -м шаге итераций как  $A^k$ , тогда мы приходим к известной форме метода Ньютона. Вектор  $\mathbf{f}$  — это вектор значений функций  $f$ ,  $g$  и  $h$ .

$$A^{k+1} = A^k - \mathbf{J}^{-1}(A^k)\mathbf{f}(A^k) \quad (13.43)$$



**Упражнение 13.24.** Определение якобиана

Покажите, что якобиан функции  $\mathbf{f}(a_1, a_2, a_3)$  имеет следующее выражение, в котором  $t_{mn}$  означает скалярное произведение  $\mathbf{q}_m \circ \mathbf{q}_n$ .

$$\mathbf{J}(a_1, a_2, a_3) \equiv \begin{bmatrix} J_{11} & J_{12} & J_{13} \\ J_{21} & J_{22} & J_{23} \\ J_{31} & J_{32} & J_{33} \end{bmatrix} \\ = \begin{bmatrix} (2a_1 - 2t_{12}a_2) & (2a_2 - 2t_{12}a_1) & 0 \\ 0 & (2a_2 - 2t_{23}a_3) & (2a_3 - 2t_{23}a_2) \\ (2a_1 - 2t_{31}a_3) & 0 & (2a_3 - 2t_{31}a_1) \end{bmatrix}$$

**Упражнение 13.25.** Вычисление обратной матрицы для якобиана

Используя символы  $J_{ij}$  из предыдущего упражнения, получите выражение обратной матрицы для якобиана — матрицы  $\mathbf{J}^{-1}$ .

Метод вычисления трехмерных координат трех точек модели в системе координат камеры представлен в алгоритме 13.3. Опыт показывает, что в большинстве случаев при поиске решения алгоритм сходится за 5–10 итераций. Однако неясно, как управлять алгоритмом для поиска нескольких решений. Нелинейная оптимизация во многих отношениях напоминает искусство, и читатель может ознакомиться с различными нюансами в справочной литературе.

**Таблица 13.3.** Итерации при решении задачи РЗР

It. k	$ f(A^k) $	$ g(A^k) $	$ h(A^k) $	$a_1$	$a_2$	$a_3$
1	6.43e + 03	3.60e + 03	1.09e + 04	1.63e + 02	1.65e + 02	1.63e + 02
2	1.46e + 03	8.22e + 02	2.48e + 03	1.06e + 02	1.08e + 02	1.04e + 02
3	2.53e + 02	1.51e + 02	4.44e + 02	8.19e + 01	9.64e + 01	1.03e + 02
...	...	...	...	...	...	...
8	2.68e + 00	6.45e - 01	5.78e + 00	8.414e + 01	9.127e + 01	8.926e + 01
9	5.00e - 02	3.87e - 02	1.71e - 01	8.414e + 01	9.126e + 01	8.925e + 01

It. k	$P_{1x}$	$P_{1y}$	$P_{1z}$	$P_{2x}$	$P_{2y}$	$P_{2z}$	$P_{3x}$	$P_{3y}$	$P_{3z}$
1	-36.9	-58.4	147.6	-34.4	-14.4	160.7	0.0	-14.5	162.4
2	-24.0	-38.0	96.0	-22.5	-9.3	105.2	0.0	-9.3	103.6
...	...	...	...	...	...	...	...	...	...
8	-19.1	-30.2	76.2	-19.1	-7.9	88.9	0.0	-7.9	88.9
9	-19.1	-30.2	76.2	-19.1	-7.9	88.9	0.0	-7.9	88.9

Для фокусного расстояния  $f = 30$  смоделировано получение точек изображения  $Q_j$  для точек  $P_1 = (-19.05, -30.16, 76.20)$ ,  $P_2 = [-19.05, -7.94, 88.90]$  и  $P_3 = (0.00, -7.94, 88.90)$ . В качестве начальных значений параметров были выбраны  $A^0 = (300, 300, 300)$  и  $\Delta = 0.2$ . После 9 итераций решение задачи РЗР сходится к исходным значениям  $P_j$  с точностью до двух десятичных знаков.

### Вычисление трехмерных координат трех точек по двумерным координатам образов этих точек на изображении.

Входные данные: три пары  $({}^M\mathbf{P}_i, {}^F\mathbf{Q}_i)$  соответствующих трехмерных и двумерных точек.

Каждая точка  ${}^M\mathbf{P}_i$  задана в системе координат модели, а  ${}^F\mathbf{Q}_i$  — в действительной системе координат изображения.

В качестве входных данных также задаются фокусное расстояние камеры  $f$  и допуск по расстоянию  $\Delta$ .

Выходные данные: координаты  ${}^C\mathbf{P}_i$  трех точек модели относительно камеры.

#### 1. Инициализация:

- (a) По координатам точек модели  ${}^M\mathbf{P}_i$  вычисляются квадраты расстояний  $d_{mn}^2$
- (b) По координатам точек изображения  ${}^F\mathbf{Q}_i$  вычисляются единичные векторы  $\mathbf{q}_i$  и скалярные произведения  $2\mathbf{q}_m \circ \mathbf{q}_n$
- (c) Выбираются начальные значения параметров в виде вектора  $\mathbf{A}^1 = [a_1, a_2, a_3]$  (Как именно?)

#### 2. Итерации: выполняются, пока не будет $\mathbf{f}(\mathbf{A}^k) \approx 0$

- (a)  $\mathbf{A}^{k+1} = \mathbf{A}^k - \mathbf{J}^{-1}(\mathbf{A}^k)\mathbf{f}(\mathbf{A}^k)$ 
  - i.  $\mathbf{A}^k = \mathbf{A}^{k+1}$
  - ii. Если существует  $\mathbf{J}^{-1}$ , то вычисляется  $\mathbf{J}^{-1}(\mathbf{A}^k)$
  - iii. Вычисляется  $\mathbf{f}(\mathbf{A}^k) = [f(a_1^k, a_2^k, a_3^k), g(a_1^k, a_2^k, a_3^k), h(a_1^k, a_2^k, a_3^k)]^t$
- (b) Цикл прекращается, когда  $\mathbf{f}(\mathbf{A}^{k+1})$  будет равно 0 в пределах допуска  $\Delta$  или, если количество итераций превысит заданный порог.

#### 3. Вычисление положения: по значениям $\mathbf{A}^{k+1}$ вычисляются все координаты ${}^C\mathbf{P}_i = a_i^{k+1}\mathbf{q}_i$

### Алгоритм 13.3. Итерационное решение задачи РЗР.

В таблице 13.3 приведены значения, полученные в процессе итерационного поиска решения задачи РЗР. Смоделировано проектирование точек  $P_i$  камерой с фокусным расстоянием  $f = 30$  для получения смоделированных точек изображения  $Q_i$ . Начальные значения параметров были выбраны далеко от истинных значений. После нескольких приближенных шагов алгоритм попадает в окрестность истинного решения и выдает координаты точек, совпадающие с

исходными координатами  $P_i$  в пределах двух десятичных знаков. Как показано в столбцах 3–5 таблицы 13.3, после 9 итераций разность между длиной стороны модели и измеренной длиной оказывается меньше 0,2 единиц. Если в качестве начальной точки выбрать  $a_i \approx 100$ , то количество итераций сократится примерно вдвое. Если расстояние от камеры до объекта примерно известно, то это хорошее начальное значение для каждого параметра.

В работе [Ohmura и др., 1988] описывается система для определения ориентации головы человека с частотой 10 Гц. На лице оператора закрепляются синие точки-метки для обозначения характерных точек модели  ${}^M P_j$  — слева от левого глаза, справа от правого глаза и ниже носа (эти точки почти не искажаются при различных выражениях лица). Синие точки легко найти на изображении, поэтому координаты  ${}^F Q_j$  вычисляются надежно и быстро. Ориентацию и местоположение лица затем можно определить по вычисленным точкам  ${}^C P_j$  и преобразованию, отображающему точки  ${}^M P_j$  в  ${}^C P_j$  (его можно определить с помощью алгоритма 13.1). В работе [Ballard, Stockman, 1995] описана система для определения положения глаз и носа без каких-либо искусственных меток. Но производительность системы оказалась намного ниже за счет серьезного усложнения обработки изображений. В обеих работах отмечалось, что ошибка вектора нормали к плоскости, построенной по трем точкам, была порядка нескольких градусов. Если плоскость точек  ${}^C P_j$  почти перпендикулярна плоскости изображения, то небольшие ошибки в определении координат наблюдаемых точек  ${}^F Q_j$  могут приводить к большим ошибкам при вычислении ориентации плоскости в трехмерном пространстве. Чтобы учесть этот эффект, в работе [Ohmura и др., 1988] ось камеры расположена под углом примерно 20 градусов к характерному направлению лица.

Уравнения (13.38) всегда должны иметь решение. Значит, мы можем вычислить ориентацию самолета по трем точкам с изображения лягушки! Для программной обработки очень важно выбрать подходящую модель: грубо говоря, для этого можно использовать тот факт, что самолеты не могут быть зелеными или не могут быть на данном изображении и т. п. Верификация модели также важна: для этого можно рассчитать предположительное положение дополнительных точек модели и проверить, есть ли они на самом деле на изображении. Например, чтобы отличить два возможных положения лица, можно после вычисления положения глаз и носа оценить расположение ушей, подбородка и бровей. Вопрос верификации будет подробнее рассмотрен в следующих разделах. Также будет учтено, что точки на цифровых изображениях имеют дискретные пиксельные координаты и что вместе с перспективным проектированием требуется учитывать радиальную дисторсию объектива камеры.

---

### Упражнение 13.26

Обсудите, как решение задачи **РЗР** можно использовать для решения задачи **Р5Р**.

## 13.7 УЛУЧШЕННЫЙ МЕТОД КАЛИБРОВКИ КАМЕРЫ\*

В данном разделе будет описан метод калибровки, разработанный Роджером [Tsai, 1987]. Этот метод широко используется в промышленных приложениях компьютерного зрения. Из опубликованных работ известно, что при аккуратной реализации этот метод обеспечивает трехмерные измерения с точностью  $1/4000$ , что является очень хорошим показателем. Общая идея калибровки была подробно рассмотрена в разделе 13.3, и в данном разделе мы будем пользоваться упрощенной записью.

- $\mathbf{P} = (x, y, z)$  — точка в трехмерной мировой системе координат.
- $\mathbf{p} = (u, v)$  — точка на вещественной плоскости изображения. (Можно считать, что ось  $u$  — это горизонтальная ось, направленная вправо, а ось  $v$  — вертикальная ось, направленная вверх.)
- $\mathbf{a} = [r, c]$  — пиксел изображения, адресуемый с помощью двух целых чисел.  $r$  — это номер строки, а  $c$  — номер столбца изображения, определяющие положение данного пиксела. (Относительно осей  $u$  и  $v$  — ось  $c$  горизонтальна и направлена вправо, ось  $r$  является вертикальной и направлена вниз.)

Процедура калибровки представляется как процедура восстановления параметров камеры. Требуется определить *параметры камеры*, которые характеризуют ее геометрические свойства и расположение. Оцениваемые параметры делятся на две группы:

1. внутренние параметры;
2. внешние параметры.

---

### Упражнение 13.27. Задача WP3P\*

Получите копию статьи [Huttenlocher, Ullman, 1988] с описанием решения задачи вычисления положения жесткой конфигурации из трех точек на основе одной матрицы слабой перспективной проекции. В этой статье решение приведено в аналитическом виде и явно дает два варианта решения, в отличие от решения задачи P3P в данной главе. Напишите программу, реализующую описанный в статье метод и проверьте ее на смоделированных данных. Эти данные получите математически, проектируя 3-точечную конфигурацию для получения трех соответствующих пар точек  $\langle P_j, Q_j \rangle$ .

#### 13.7.1 Внутренние параметры камеры

*Внутренние* параметры действительно являются характеристиками камеры, они зависят от особенностей конкретного используемого устройства. К ним относятся следующие параметры:

- главная точка  $(u_0, v_0)$ : точка пересечения оптической оси и плоскости изображения;

- масштабные множители  $\{d_x, d_y\}$  для горизонтального и вертикального размеров пиксела;
- множитель формата кадра  $\tau_1$ : масштабный множитель, применяемый для учета отношения размеров кадра у данной камеры;
- фокусное расстояние  $f$ : расстояние от оптического центра до плоскости изображения;
- множитель дисторсии объектива ( $\kappa_1$ ): масштабный множитель, применяемый для моделирования радиальной дисторсии объектива.

В этих определениях есть несколько ссылок на оптический центр объектива камеры. В этой точке располагается начало системы координат камеры. Оптическая ось — это перпендикуляр к плоскости изображения, проходящий через оптический центр. Часто, но не всегда, главная точка располагается в центре изображения. Масштабные множители  $d_x$  и  $d_y$  представляют горизонтальный и вертикальный размеры пиксела в вещественных единицах измерения мировой системы координат, например, в мм. Будем полагать, что для данной камеры известны параметры  $u_0, v_0, d_x, d_y$  и  $\tau_1$ . Следовательно, в процессе калибровки надо определить только фокусное расстояние  $f$  и множитель дисторсии  $\kappa_1$ .

### 13.7.2 Внешние параметры камеры

*Внешние* параметры описывают местоположение и ориентацию системы координат камеры в трехмерном мировом пространстве. К этим параметрам относятся:

- параметры переноса:

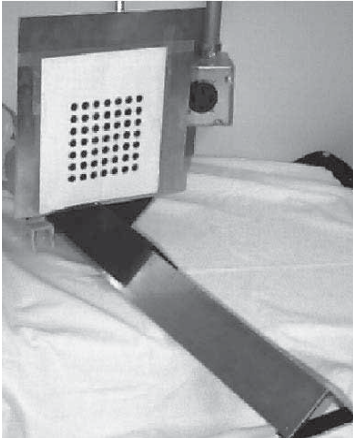
$$\mathbf{t} = [t_x \quad t_y \quad t_z]^T \quad (13.44)$$

- параметры поворота:

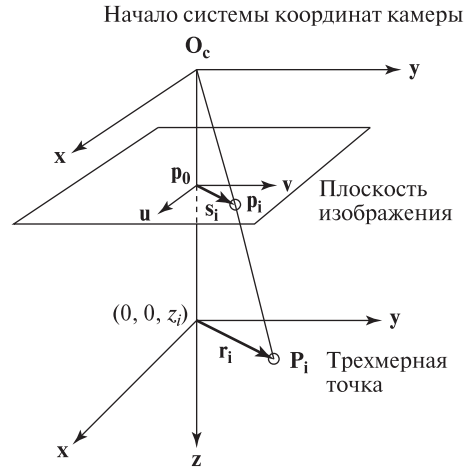
$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (13.45)$$

Параметры переноса описывают местоположение камеры в мировой системе координат, а параметры поворота — ее ориентацию. Особо отметим, что **есть только три независимых параметра поворота, а не девять**.

Описываемый ниже метод калибровки является автономным, достаточно точным, эффективным и гибким (см. [Tsai, 1987]). Более того, его можно использовать с любой готовой камерой и объективом, хотя он может не вполне удовлетворительно моделировать объективы некоторых типов. На рис. 13.22 показан калибровочный стенд, который был использован в описываемой далее системе восстановления параметров трехмерного объекта. Стенд содержит металлическую пластину, на которой в виде сетки  $7 \times 7$  нанесены черные окружности. Центры окружностей используются в качестве характерных точек. Пластина закреплена на горизонтальной направляющей. Пластина пер-



**Рис. 13.22.** Калибровочный стенд, в котором для получения множества характерных точек трехмерного пространства используется перемещаемый двумерный шаблон.



**Рис. 13.23.** Геометрическая модель метода калибровки Цая. Точка изображения  $\mathbf{p}_i = (u_i, v_i)$  соответствует точке калибровочного объекта  $\mathbf{P}_i = (x_i, y_i, z_i)$ . Точка  $\mathbf{p}_0 = (u_0, v_0)$  — это главная точка. Любая радиальная дисторсия должна смещать точку изображения  $\mathbf{p}_i$  вдоль направления  $\mathbf{p}_0 - \mathbf{p}_i$ .

пендикулярна направляющей и может перемещаться вдоль нее с шагом 10 мм. Положение направляющей задает трехмерную мировую систему координат.

На стенде, показанном на рис. 13.22, снимается несколько изображений для различных положений калибровочного объекта на направляющей, т. е. на разных расстояниях от камеры. Камера не должна ни перемещаться, ни фокусироваться в процессе калибровки или при последующем использовании. На каждом изображении выполняется поиск окружностей и определяются их центры. Результатом обработки изображений является множество соответствующих точек трехмерного пространства и двумерных точек изображения. Требуется  $n > 5$  пар точек, мы будем обозначать их так:

$$\{([x_i, y_i, z_i], [u_i, v_i]) \mid i = 1, \dots, n\}$$

Действительные координаты на изображении  $(u, v)$  вычисляются по пиксельным координатам  $(r, c)$  по формулам

$$u = \tau_1 d_x (c - u_0) \quad (13.46)$$

$$v = -d_y (r - v_0) \quad (13.47)$$

где  $d_x$  и  $d_y$  — расстояния между центрами пикселей в горизонтальном и вертикальном направлении, а  $\tau_1$  — масштабный множитель, учитывающий отношение сторон изображения данной камеры.

На рис. 13.23 приведена геометрическая модель этой процедуры. Точка  $\mathbf{P}_i = (x_i, y_i, z_i)$  — это произвольная точка трехмерного пространства. Соответствующая двумерная точка на изображении обозначена как  $\mathbf{p}_i$ . Вектор  $\mathbf{r}_i$  направлен из точки оптической оси  $(0, 0, z_i)$  к трехмерной точке  $\mathbf{P}_i$ . Вектор  $\mathbf{s}_i$  направлен из главной точки  $\mathbf{p}_0$  к точке изображения  $\mathbf{p}_i$ . Вектор  $\mathbf{s}_i$  параллелен вектору  $\mathbf{r}_i$ . Любая радиальная дисторсия объектива приводит к смещению точки изображения вдоль  $\mathbf{s}_i$ .

Цай (Tsai) сделал следующее наблюдение, которое позволяет вычислить на первом шаге большинство внешних параметров. Так как радиальная дисторсия оказывает влияние вдоль вектора  $\mathbf{s}_i$ , то матрица поворота может быть определена без учета дисторсии. Также, без учета  $\kappa_1$ , могут быть вычислены параметры  $t_x$  и  $t_y$ . Вычисление  $t_z$  должно выполняться на втором шаге калибровки, так как изменение  $t_z$  влияет на изображение похожим образом, что и  $\kappa_1$ .

Вместо непосредственного решения для поиска всех неизвестных сначала получим множество параметров  $\boldsymbol{\mu}$ , на основе которых можно будет вычислить внешние параметры камеры. Для данных  $n$  пар соответствующих точек  $(x_i, y_i, z_i)$  и  $(u_i, v_i)$  (где  $i = 1, \dots, n$ ,  $n > 5$ ) сформируем матрицу  $\mathbf{A}$  со строками  $a_i$ :

$$a_i = [v_i x_i, v_i y_i, -u_i x_i, -u_i y_i, v_i] \quad (13.48)$$

Введем вектор  $\boldsymbol{\mu} = [\mu_1, \mu_2, \mu_3, \mu_4, \mu_5]$ , содержащий неизвестные параметры, выражаемые через параметры поворота  $r_{11}$ ,  $r_{12}$ ,  $r_{21}$  и  $r_{22}$  и параметры переноса  $t_x$  и  $t_y$ :

$$\mu_1 = \frac{r_{11}}{t_y} \quad (13.49)$$

$$\mu_2 = \frac{r_{12}}{t_y} \quad (13.50)$$

$$\mu_3 = \frac{r_{21}}{t_y} \quad (13.51)$$

$$\mu_4 = \frac{r_{22}}{t_y} \quad (13.52)$$

$$\mu_5 = \frac{t_x}{t_y} \quad (13.53)$$

Введем вектор  $\mathbf{b} = (u_1, u_2, \dots, u_n)$ , содержащий координаты  $u_i$   $n$  соответствующих точек изображения. Так как  $\mathbf{A}$  и  $\mathbf{b}$  известны, то систему линейных уравнений

$$\mathbf{A}\boldsymbol{\mu} = \mathbf{b} \quad (13.54)$$

можно решить относительно вектора неизвестных  $\boldsymbol{\mu}$  (методы решения систем линейных уравнений описаны в учебниках по линейной алгебре, например, [Johnson, Riess, Arnold, 1989]). Теперь вектор  $\boldsymbol{\mu}$  можно использовать для вычисления параметров поворота и переноса. Ход решения приведен ниже.

1. Обозначим  $U = \mu_1^2 + \mu_2^2 + \mu_3^2 + \mu_4^2$ . Вычислим квадрат  $y$ -компоненты матрицы переноса  $t_y$  следующим образом:

$$t_y^2 = \begin{cases} \frac{U - [U^2 - 4(\mu_1\mu_4 - \mu_2\mu_3)^2]^{1/2}}{2(\mu_1\mu_4 - \mu_2\mu_3)^2} & \text{если } (\mu_1\mu_4 - \mu_2\mu_3) \neq 0 \\ \frac{1}{\mu_1^2 + \mu_2^2} & \text{если } (\mu_1^2 + \mu_2^2) \neq 0 \\ \frac{1}{\mu_3^2 + \mu_4^2} & \text{если } (\mu_3^2 + \mu_4^2) \neq 0 \end{cases} \quad (13.55)$$

2. Пусть  $t_y = (t_y^2)^{1/2}$  (выберем положительное значение квадратного корня). Вычислим четыре параметра поворота и  $x$ -компоненту переноса  $t_x$  по вычисленным ранее значениям компонент вектора  $\mu$ :

$$r_{11} = \mu_1 t_y \quad (13.56)$$

$$r_{12} = \mu_2 t_y \quad (13.57)$$

$$r_{21} = \mu_3 t_y \quad (13.58)$$

$$r_{22} = \mu_4 t_y \quad (13.59)$$

$$t_x = \mu_5 t_y \quad (13.60)$$

3. Для определения правильного знака  $t_y$  выберем точку объекта  $\mathbf{P}$ , координаты изображения которой  $(u, v)$  лежат далеко от центра изображения (чтобы избежать проблем при численном решении). Обозначим  $\mathbf{P} = (x, y, z)$  и вычислим:

$$\xi_x = r_{11}x + r_{12}y + t_x \quad (13.61)$$

$$\xi_y = r_{21}x + r_{22}y + t_y \quad (13.62)$$

Эти выражения выглядят как применение вычисленных параметров поворота к  $x$ - и  $y$ -координатам точки  $\mathbf{P}$ . Если  $\xi_x$  имеет тот же знак, что и  $u$ , а  $\xi_y$  одного знака с  $v$ , то знак  $t_y$  был выбран правильно. В противном случае знак  $t_y$  надо изменить.

4. Теперь можно вычислить оставшиеся параметры поворота:

$$r_{13} = (1 - r_{11}^2 - r_{12}^2)^{1/2} \quad (13.63)$$

$$r_{23} = (1 - r_{21}^2 - r_{22}^2)^{1/2} \quad (13.64)$$

$$r_{31} = \frac{1 - r_{11}^2 - r_{12}^2}{r_{13}} \quad (13.65)$$

$$r_{32} = \frac{1 - r_{21}^2 - r_{22}^2}{r_{23}} \quad (13.66)$$

$$r_{33} = (1 - r_{31}r_{13} - r_{32}r_{23})^{1/2} \quad (13.67)$$



При записи этих уравнений учитывалось свойство ортонормированности матрицы  $\mathbf{R}$ . Из-за неоднозначности квадратного корня знаки  $r_{23}$ ,  $r_{31}$  и  $r_{32}$  могут быть выбраны неправильно. На данном шаге, знак  $r_{23}$  следует изменить, чтобы сохранить ортонормированность матрицы, если выполняется условие

$$r_{11}r_{21} + r_{12}r_{22} > 0$$

Другие два параметра могут потребовать коррекции после вычисления фокусного расстояния.

5. Фокусное расстояние  $f$  и  $z$ -компонента переноса  $t_z$  вычисляются теперь с помощью второй системы линейных уравнений. Сначала сформируем матрицу  $\mathbf{A}'$ , строки которой выражаются так:

$$a'_i = (r_{21}x_i + r_{22}y_i + t_y, v_i) \quad (13.68)$$

Теперь построим вектор  $\mathbf{b}'$ , содержащий строки:

$$b'_i = (r_{31}x_i + r_{32}y_i)v_i. \quad (13.69)$$

Надо решить систему линейных уравнений

$$\mathbf{A}'\mathbf{v} = \mathbf{b}' \quad (13.70)$$

Относительно  $\mathbf{v} = (f, t_z)^T$ . На данном этапе будут получены оценочные значения  $f$  и  $t_z$ .

6. Если  $f < 0$ , то следует сменить знаки  $r_{13}$ ,  $r_{23}$ ,  $r_{31}$ ,  $r_{32}$ ,  $f$  и  $t_z$ . Это должно привести к использованию правой системы координат.
7. Оценочные значения  $f$  и  $t_z$  можно использовать для вычисления множителя дисторсии объектива  $\kappa_1$  и затем для уточнения значений  $f$  и  $t_z$ . Используемая здесь простая модель дисторсии предполагает, что истинные значения координат на изображении  $(\hat{u}, \hat{v})$  могут быть получены по измеренным координатам с помощью следующих выражений:

$$\hat{u} = u(1 + \kappa_1 r^2) \quad (13.71)$$

$$\hat{v} = v(1 + \kappa_1 r^2) \quad (13.72)$$

где радиус дисторсии  $r$  равен

$$r = (u^2 + v^2)^{1/2} \quad (13.73)$$

С помощью уравнений перспективной проекции, модифицированных для учета дисторсии, можно получить набор нелинейных уравнений следующего вида:

$$\left\{ v_i(1 + \kappa_1 r^2) = f \frac{r_{21}x_i + r_{22}y_i + r_{23}z_i + t_y}{r_{31}x_i + r_{32}y_i + r_{33}z_i + t_z} \right\} \quad i = 1, \dots, n \quad (13.74)$$

Решив эту систему методом нелинейной регрессии, можно получить значения  $f$ ,  $t_z$  и  $\kappa_1$ .

### 13.7.3 Пример калибровки

Рассмотрим пример для пояснения процедуры калибровки. В следующей таблице приведены пять соответствующих точек, полученных в качестве входных данных для калибровки зрительной системы. Единицами измерения и в мировой системе координат, и в системе изображения  $\mathbf{u}\text{-}\mathbf{v}$  выбраны сантиметры.

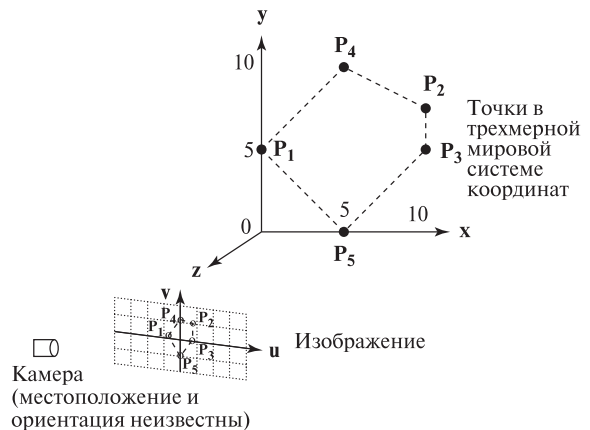
$i$	$x_i$	$y_i$	$z_i$	$u_i$	$v_i$
1	0.00	5.00	0.00	-0.58	0.00
2	10.00	7.50	0.00	1.73	1.00
3	10.00	5.00	0.00	1.73	0.00
4	5.00	10.00	0.00	0.00	1.00
5	5.00	0.00	0.00	0.00	-1.00

На рис. 13.24 в трехмерном пространстве показаны 5 калибровочных точек и их приблизительное положение на плоскости изображения камеры, у которой неизвестны и требуют вычисления положение, ориентация и фокусное расстояние. На рис. 13.25 показаны точки изображения в непрерывной системе координат  $\mathbf{u}\text{-}\mathbf{v}$ .

С использованием 5 соответствующих точек матрицу  $\mathbf{A}$  и вектор  $\mathbf{b}$  из уравнения (13.54) можно записать так:

$$\mathbf{A} = \begin{bmatrix} v_1 x_1 & v_1 y_1 & -u_1 x_1 & -u_1 y_1 & v_1 \\ 0.00 & 0.00 & 0.00 & 2.89 & 0.00 \\ 10.00 & 7.50 & -17.32 & -12.99 & 1.00 \\ 0.00 & 0.00 & -17.32 & -8.66 & 0.00 \\ 5.00 & 10.00 & 0.00 & 0.00 & 1.00 \\ -5.00 & 0.00 & 0.00 & 0.00 & -1.00 \end{bmatrix}$$

**Рис. 13.24.** Трехмерные мировые координаты и соответствующие двумерные точки изображения, которые являются входными данными для калибровки камеры. В результате калибровки надо вычислить параметры камеры, включая ее положение, ориентацию и фокусное расстояние.



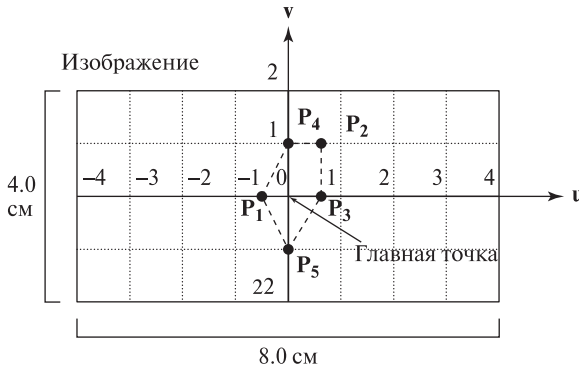


Рис. 13.25. Точки изображения в системе координат изображения  $u-v$ .

и

$$\mathbf{b} = \begin{bmatrix} u_i \\ -0.58 \\ 1.73 \\ 1.73 \\ 0.00 \\ 0.00 \end{bmatrix}$$

Решив систему  $\mathbf{A}\boldsymbol{\mu} = \mathbf{b}$  относительно вектора  $\boldsymbol{\mu}$ , получим:

$$\boldsymbol{\mu} = \begin{bmatrix} \mu_i \\ -0.17 \\ 0.00 \\ 0.00 \\ -0.20 \\ 0.87 \end{bmatrix}$$

Теперь надо вычислить  $U$  и использовать его для нахождения  $t_y^2$  из уравнения (13.55). Имеем:

$$U = \mu_1^2 + \mu_2^2 + \mu_3^2 + \mu_4^2 = .07$$

По первой формуле из (13.55):

$$t_y^2 = \frac{U - [U^2 - 4(\mu_1\mu_4 - \mu_2\mu_3)^2]^{1/2}}{2(\mu_1\mu_4 - \mu_2\mu_3)^2} = 25$$

Выберем  $t_y = 5$  (положительное значение квадратного корня). Тогда:

$$r_{11} = \mu_1 t_y = -0.87$$

$$r_{12} = \mu_2 t_y = 0$$

$$r_{21} = \mu_3 t_y = 0$$

$$r_{22} = \mu_4 t_y = -1.0$$

$$t_x = \mu_5 t_y = 4.33$$

Теперь для проверки знака  $t_y$  вычислим  $\xi_x$  и  $\xi_y$  для точки  $\mathbf{P}_2 = (10.0, 7.5, 0.0)$  и соответствующей точки изображения  $\mathbf{p}_2 = (1.73, 1.0)$ , которая расположена далеко от центра изображения.

$$\xi_x = r_{11}x + r_{12}y + t_x = (-0.87)(10) + 0 + 4.33 = -4.37$$

$$\xi_y = r_{21}x + r_{22}y + t_y = 0 + (-1.0)(7.5) + 5 = -2.5$$

Так как знаки  $\xi_x$  и  $\xi_y$  не совпадают со знаками координат  $\mathbf{p}_2$ , то, значит, знак  $t_y$  выбран неверно и его надо изменить. Это даст:

$$t_y = -5$$

$$r_{11} = 0.87$$

$$r_{12} = 0$$

$$r_{21} = 0$$

$$r_{22} = 1.0$$

$$t_x = -4.33$$

Продолжим вычисления и получим оставшиеся параметры поворота:

$$r_{13} = (1 - r_{11}^2 - r_{12}^2)^{1/2} = 0.5$$

$$r_{23} = (1 - r_{21}^2 - r_{22}^2)^{1/2} = 0.$$

$$r_{31} = \frac{1 - r_{11}^2 - r_{12}r_{21}}{r_{13}} = 0.5$$

$$r_{32} = \frac{1 - r_{21}r_{12} - r_{22}^2}{r_{23}} = 0.$$

$$r_{33} = (1 - r_{31}r_{13} - r_{32}r_{23})^{1/2} = 0.87$$

Проверка знака  $r_{11}r_{21} + r_{12}r_{22} = 0$  показывает, что он положителен, значит, знак  $r_{23}$  менять не нужно.

Теперь сформируем вторую систему линейных уравнений.

$$\mathbf{A}' = \begin{bmatrix} r_{21}x_i + r_{22}y_i + t_y & v_i \\ 0.00 & 0.00 \\ 2.500 & -1.00 \\ 0.00 & 0.00 \\ 5.00 & -1.00 \\ -5.00 & 1.00 \end{bmatrix}$$

и

$$\mathbf{b}' = \begin{bmatrix} (r_{31}x_i + r_{32}y_i)v_i \\ 0.0 \\ 5.0 \\ 0.0 \\ 2.5 \\ -2.5 \end{bmatrix}$$

Решая систему  $\mathbf{A}'\mathbf{v} = \mathbf{b}'$ , найдем вектор  $\mathbf{v} = (f, t_z)$ , компоненты которого равны:

$$\begin{aligned} f &= -1.0 \\ t_z &= -7.5 \end{aligned}$$

Так как значение  $f$  получилось отрицательным, наша система координат не является правой. Для переворота оси  $z$  изменим знаки  $r_{13}$ ,  $r_{23}$ ,  $r_{31}$ ,  $r_{32}$ ,  $f$  и  $t_z$ . В результате имеем:

$$\mathbf{R} = \begin{bmatrix} 0.87 & 0.00 & -0.50 \\ 0.00 & -1.00 & 0.00 \\ -0.50 & 0.00 & 0.87 \end{bmatrix}$$

и

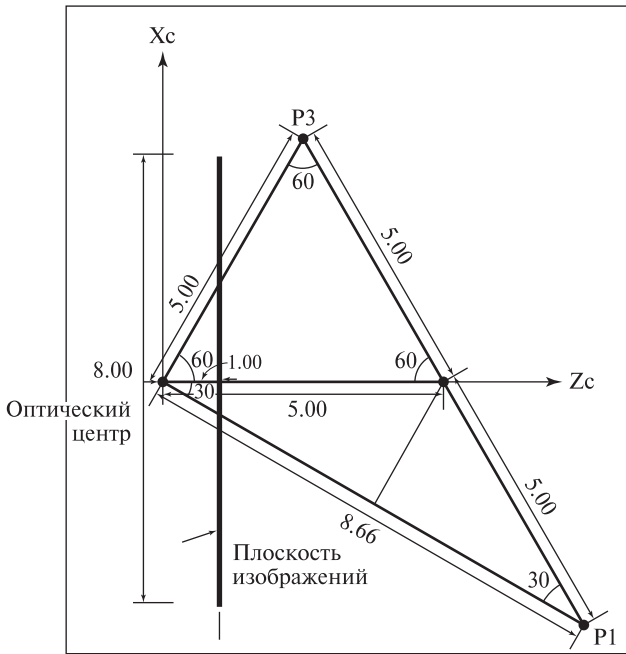
$$\mathbf{T} = \begin{bmatrix} -4.33 \\ -5.00 \\ 7.50 \end{bmatrix}$$

и  $f = 1$ .

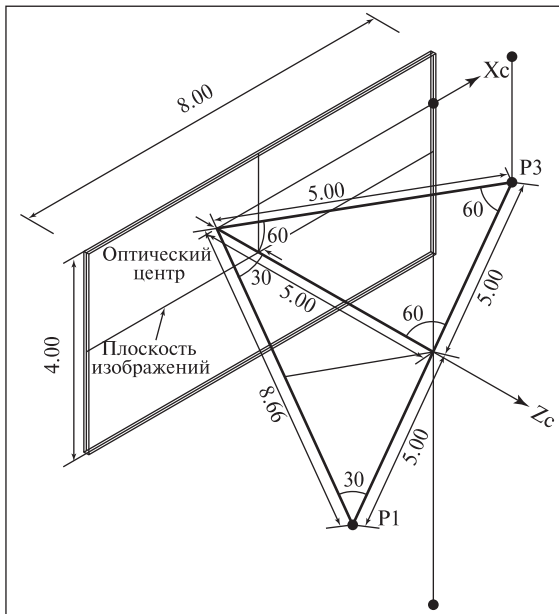
Так как в нашем примере не учитывалась дисторсия, то это и есть окончательные результаты калибровки. На рис. 13.26 результаты калибровки показаны с двух различных точек зрения.

### Упражнение 13.28

Проверьте, что полученная в рассмотренном примере матрица  $\mathbf{R}$  является ортонормированной.



(a) Вид сверху  
(плоскость изображения обозначена толстой черной линией)



(b) Перспективный вид

**Рис. 13.26.** Два вида камеры и плоскости изображения в мировой системе координат по результатам, полученным в примере калибровки методом Цзя (Tsai). (С любезного разрешения Habib Abi-Rached.)

---

**Упражнение 13.29.** Использование параметра камеры  $f$ 

Используя методы евклидовой геометрии в трехмерном пространстве и рис. 13.26 (а), найдите проекции точек  $P_1$  и  $P_3$  на плоскость изображения. (Все прямые на рис. 13.26 (а) являются компланарными.) Проверьте, что ваши результаты согласуются с координатами точек  $p_1$  и  $p_3$ , использованными в примере калибровки.

---

**Упражнение 13.30.** Использование параметров камеры  $\mathbf{R}$  и  $\mathbf{T}$ 

Точки  $P_1$  и  $P_3$  в примере калибровки заданы в мировой системе координат. Найдите их координаты в системе координат камеры двумя способами.

1. С помощью евклидовой геометрии и рис. 13.26 (а).
2. С использованием параметров камеры, полученных в процессе калибровки.

### 13.8 ОЦЕНКА ПОЛОЖЕНИЯ ОБЪЕКТОВ\*

В промышленных приложениях, особенно при управлении роботами, важно уметь определять положение трехмерных объектов в координатах рабочего пространства. Так как положение камеры в рабочем пространстве может быть определено в процессе калибровки, то задача сводится к вычислению положения объекта относительно камеры. В данном разделе приводится метод определения положения объекта, который по точности превосходит простой метод, рассмотренный в разделе 13.6. Использование пар соответствующих точек является простым и чаще всего используемым методом вычисления положения объектов. Об использовании соответствия двумерных и трехмерных координат отрезков, двумерных эллипсов и трехмерных окружностей, а также соответствия между любыми комбинациями пар точек, пар отрезков и пар эллипс-окружность, см. работу [Ji и др., 1998].

#### 13.8.1 Оценка положения на основе соответствия двумерных и трехмерных координат точек

Будем использовать модель камеры из предыдущего раздела и предположим, что камера была калибрована для получения ее внутренних и внешних параметров. Задача заключается в определении положения объекта по  $n$  точкам трехмерной модели объекта и соответствующим двумерным координатам этих точек на изображении. Эта задача является существенно нелинейной. Поэтому необходимы нелинейные методы для оценки параметров положения; однако, при некоторых условиях, может быть найдено приближенное линейное решение.

Обозначим координаты точки модели  $\mathbf{M}\mathbf{P}$  в модельной системе координат  $(x, y, z)$ . Пусть система координат объекта и система координат камеры связаны преобразованием  $\mathbf{M}\mathbf{C}\mathbf{T}\mathbf{r} = \{\mathbf{R}, \mathbf{T}\}$ , которое описывается с помощью матрицы поворота  $\mathbf{R}$  и вектора переноса  $\mathbf{T} = (t_x, t_y, t_z)$ . Тогда перспективное проектирование  $\mathbf{M}\mathbf{P}$  на плоскость изображения дает координаты точки изображения  $(u, v)$

$$u = f \frac{r_{11}x + r_{12}y + r_{13}z + t_x}{r_{31}x + r_{32}y + r_{33}z + t_z} \quad (13.75)$$

и

$$v = f \frac{r_{21}x + r_{22}y + r_{23}z + t_y}{r_{31}x + r_{32}y + r_{33}z + t_z} \quad (13.76)$$

где  $f$  — фокусное расстояние камеры, которое было определено в процессе калибровки.

Преобразование между модельной системой координат и системой координат камеры соответствует положению объекта относительно системы координат камеры. Используя, как и раньше, нашу модель камеры с перспективной проекцией, мы имеем 9 параметров поворота и 3 параметра переноса. Эти параметры входят в 12 уравнений вида:

$$\mathbf{B} \mathbf{w} = 0 \quad (13.77)$$

где

$$\mathbf{B} = \begin{pmatrix} fx_1 & fy_1 & fz_1 & 0 & 0 & 0 & -u_1x_1 & -u_1y_1 & -u_1z_1 & f & 0 & -u_1 \\ 0 & 0 & 0 & fx_1 & fy_1 & fz_1 & -v_1x_1 & -v_1y_1 & -v_1z_1 & 0 & f & -v_1 \\ fx_2 & fy_2 & fz_2 & 0 & 0 & 0 & -u_2x_2 & -u_2y_2 & -u_2z_2 & f & 0 & -u_2 \\ 0 & 0 & 0 & fx_2 & fy_2 & fz_2 & -v_2x_2 & -v_2y_2 & -v_2z_2 & 0 & f & -v_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ fx_6 & fy_6 & fz_6 & 0 & 0 & 0 & -u_6x_6 & -u_6y_6 & -u_6z_6 & f & 0 & -u_6 \\ 0 & 0 & 0 & fx_6 & fy_6 & fz_6 & -v_6x_6 & -v_6y_6 & -v_6z_6 & 0 & f & -v_6 \end{pmatrix} \quad (13.78)$$

и

$$\mathbf{w} = (r_{11} \ r_{12} \ r_{13} \ r_{21} \ r_{22} \ r_{23} \ r_{31} \ r_{32} \ r_{33} \ t_x \ t_y \ t_z)^T \quad (13.79)$$

Однако, если надо вычислить независимые параметры положения, а не просто аффинное преобразование, совмещающее точки модели с точками изображения, то на элементы  $\mathbf{R}$  надо наложить такие условия, чтобы при их выполнении действительно получалась матрица поворота в трехмерном пространстве. В частности, матрица поворота должна быть ортонормированной: ее вектора-строки должны быть по модулю равны 1, и они должны быть ортогональны друг другу. Эти условия можно записать так:

$$\|\mathbf{R}_1\| = r_{11}^2 + r_{12}^2 + r_{13}^2 = 1 \quad (13.80)$$

$$\|\mathbf{R}_2\| = r_{21}^2 + r_{22}^2 + r_{23}^2 = 1$$

$$\|\mathbf{R}_3\| = r_{31}^2 + r_{32}^2 + r_{33}^2 = 1$$



и

$$\begin{aligned} R_1 \circ R_2 &= 0 \\ R_1 \circ R_3 &= 0 \\ R_2 \circ R_3 &= 0 \end{aligned} \quad (13.81)$$

Условия, накладываемые на  $\mathbf{R}$ , приводят задачу к нелинейному виду. Если условия на длину векторов-строк  $\mathbf{R}$  учитывать по очереди, и вычислять независимо, то для определения вектора-строки можно использовать метод линейной оптимизации при наличии ограничений (см. книгу [Faugeras и др., 1993]).

### 13.8.2 Линейная оптимизация при наличии ограничений

Задача состоит в том, чтобы для системы (13.77) найти решение — вектор  $\mathbf{w}$ , который минимизирует  $\|\mathbf{B}\mathbf{w}\|$  при наличии ограничения  $\|\mathbf{w}'\|^2 = 1$ , где вектор  $\mathbf{w}'$  является подмножеством элементов  $\mathbf{w}$ . Если ограничение накладывается на первый вектор-строку  $\mathbf{R}$ , то

$$\mathbf{w}' = \begin{pmatrix} r_{11} \\ r_{12} \\ r_{13} \end{pmatrix}$$

Для решения вышеприведенной задачи, надо переписать исходную систему уравнений  $\mathbf{B}\mathbf{w} = \mathbf{0}$  в следующем виде:

$$\mathbf{C}\mathbf{w}' + \mathbf{D}\mathbf{w}'' = \mathbf{0}$$

где вектор  $\mathbf{w}''$  — это вектор оставшихся элементов  $\mathbf{w}$ , не вошедших в  $\mathbf{w}'$ . С использованием рассмотренного выше примера, т. е., если ограничение накладывается на первую строку  $\mathbf{R}$ , то

$$\mathbf{w}'' = (r_{21} \ r_{22} \ r_{23} \ r_{31} \ r_{32} \ r_{33} \ t_x \ t_y \ t_z)^T$$

Исходная задача теперь может быть сформулирована так: минимизировать целевую функцию  $\mathbf{O} = \mathbf{C}\mathbf{w}' + \mathbf{D}\mathbf{w}''$ , т. е.

$$\min_{\mathbf{w}', \mathbf{w}''} \|\mathbf{C}\mathbf{w}' + \mathbf{D}\mathbf{w}''\|^2 \quad (13.82)$$

при наличии ограничения  $\|\mathbf{w}'\|^2 = 1$ . С помощью метода множителя Лагранжа, последнее выражение можно записать так:

$$\min_{\mathbf{w}', \mathbf{w}''} \left[ \|\mathbf{C}\mathbf{w}' + \mathbf{D}\mathbf{w}''\|^2 + \lambda(1 - \|\mathbf{w}'\|^2) \right] \quad (13.83)$$

Эта задача минимизации может быть решена путем взятия частных производных целевой функции по  $\mathbf{w}'$  и  $\mathbf{w}''$  и последующего приравнивания их к 0.

$$\frac{\partial \mathbf{O}}{\partial \mathbf{w}'} = 2\mathbf{C}^T(\mathbf{C}\mathbf{w}' + \mathbf{D}\mathbf{w}'') - 2\lambda\mathbf{w}' = 0 \quad (13.84)$$

$$\frac{\partial \mathbf{O}}{\partial \mathbf{w}''} = 2\mathbf{D}^T(\mathbf{C}\mathbf{w}' + \mathbf{D}\mathbf{w}'') = 0 \quad (13.85)$$

Уравнение (13.85) эквивалентно уравнению:

$$\mathbf{w}'' = -(\mathbf{D}^T \mathbf{D})^{-1} \mathbf{D}^T \mathbf{C} \mathbf{w}' \quad (13.86)$$

Подставим (13.86) в (13.84) и получим:

$$\lambda \mathbf{w}' = [\mathbf{C}^T \mathbf{C} - \mathbf{C}^T \mathbf{D} (\mathbf{D}^T \mathbf{D})^{-1} \mathbf{D}^T \mathbf{C}] \mathbf{w}' \quad (13.87)$$

Можно увидеть, что  $\lambda$  является собственным вектором матрицы

$$\mathbf{M} = \mathbf{C}^T \mathbf{C} - \mathbf{C}^T \mathbf{D} (\mathbf{D}^T \mathbf{D})^{-1} \mathbf{D}^T \mathbf{C} \quad (13.88)$$

Следовательно, искомое решение для  $\mathbf{w}'$  соответствует наименьшему собственному вектору матрицы  $\mathbf{M}$ . Соответствующий вектор  $\mathbf{w}''$  можно сразу вычислить из уравнения (13.86). Важно отметить, что ограничение на длину было наложено только на один вектор-строку матрицы  $\mathbf{R}$ , поэтому значения  $\mathbf{w}''$  недостоверны и ими пользоваться не следует. Однако вектор  $\mathbf{w}''$  дает важную информацию относительно знака вектора-строки, на который было наложено ограничение. Ограничение накладывалось в виде  $\|\mathbf{w}'\|^2 = 1$ , но знак  $\mathbf{w}'$  этим ограничением не определяется. Следовательно, необходимо проверить, будет или нет найденный вектор  $\mathbf{w}'$  физически возможным решением. В частности, компонента переноса  $t_z$  должна быть положительной, чтобы объект был расположен перед камерой, а не позади нее. Если элемент вектора  $\mathbf{w}''$ , соответствующий  $t_z$ , получился отрицательным, то это значит, что правильно вычислена длина вектора  $\mathbf{w}'$ , но его знак надо изменить. Таким образом, окончательное выражение для вычисляемого вектора  $\mathbf{w}'$  принимает вид:

$$\mathbf{w}' = \text{sign}(w''_9) \mathbf{w}'' \quad (13.89)$$

### 13.8.3 Вычисление преобразования $\mathbf{Tr} = \{\mathbf{R}, \mathbf{T}\}$

Вектор-строка  $\mathbf{R}_1$  вычисляется первым путем вычисления вектора  $\mathbf{w}'$ , так как в описанном выше случае  $\mathbf{R}_1 = \mathbf{w}'$ . Матрицы  $\mathbf{C}$  и  $\mathbf{D}$  имеют вид:

$$\mathbf{C} = \begin{pmatrix} x_1 & y_1 & z_1 \\ 0 & 0 & 0 \\ x_2 & y_2 & z_2 \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ x_6 & y_6 & z_6 \\ 0 & 0 & 0 \end{pmatrix} \quad (13.90)$$

и

$$\mathbf{D} = \begin{pmatrix} 0 & 0 & 0 & -u_1x_1 & -u_1y_1 & -u_1z_1 & f & 0 & -u_1 \\ fx_1 & fy_1 & fz_1 & -v_1x_1 & -v_1y_1 & -v_1z_1 & 0 & f & -v_1 \\ 0 & 0 & 0 & -u_2x_2 & -u_2y_2 & -u_2z_2 & 0 & f & -u_2 \\ fx_2 & fy_2 & fz_2 & -v_2x_2 & -v_2y_2 & -v_2z_2 & 0 & f & -v_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & -u_6x_6 & -u_6y_6 & -u_6z_6 & f & 0 & -u_6 \\ fx_6 & fy_6 & fz_6 & -v_6x_6 & -v_6y_6 & -v_6z_6 & 0 & f & -v_6 \end{pmatrix} \quad (13.91)$$

Затем аналогичным образом вычисляется вектор-строка  $\mathbf{R}_2$ , и теперь ограничение накладывается на его длину, т. е.  $\mathbf{R}_2 = \mathbf{w}'$ . В данном случае матрицы  $\mathbf{C}$  и  $\mathbf{D}$  имеют вид:

$$\mathbf{C} = \begin{pmatrix} 0 & 0 & 0 \\ fx_1 & fy_1 & fz_1 \\ 0 & 0 & 0 \\ fx_2 & fy_2 & fz_2 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \\ fx_6 & fy_6 & fz_6 \end{pmatrix} \quad (13.92)$$

и

$$\mathbf{D} = \begin{pmatrix} fx_1 & fy_1 & fz_1 & -u_1x_1 & -u_1y_1 & -u_1z_1 & f & 0 & -u_1 \\ 0 & 0 & 0 & -v_1x_1 & -v_1y_1 & -v_1z_1 & 0 & f & -v_1 \\ fx_2 & fy_2 & fz_2 & -u_2x_2 & -u_2y_2 & -u_2z_2 & f & 0 & -u_2 \\ 0 & 0 & 0 & -v_2x_2 & -v_2y_2 & -v_2z_2 & 0 & f & -v_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ fx_6 & fy_6 & fz_6 & -u_6x_6 & -u_6y_6 & -u_6z_6 & f & 0 & -u_6 \\ 0 & 0 & 0 & -v_6x_6 & -v_6y_6 & -v_6z_6 & 0 & f & -v_6 \end{pmatrix} \quad (13.93)$$

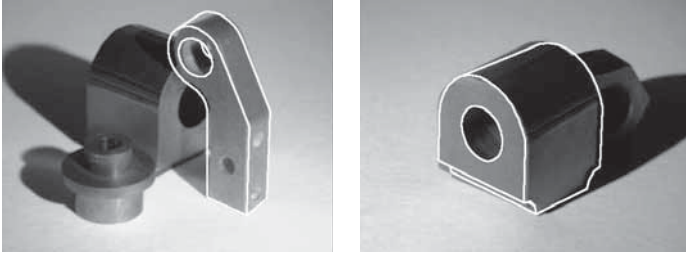
$\mathbf{R}_3$  можно вычислить по аналогии с  $\mathbf{R}_1$  и  $\mathbf{R}_2$ , но это не гарантирует ортогональности  $\mathbf{R}_3$  векторам  $\mathbf{R}_1$  и  $\mathbf{R}_2$ . Поэтому  $\mathbf{R}_3$  лучше вычислить так:

$$\mathbf{R}_3 = \frac{\mathbf{R}_1 \times \mathbf{R}_2}{\|\mathbf{R}_1 \times \mathbf{R}_2\|}. \quad (13.94)$$

Были удовлетворены все ограничения на векторы-строки  $\mathbf{R}$ , кроме одного: нет гарантии, что векторы  $\mathbf{R}_1$  и  $\mathbf{R}_2$  ортогональны. Чтобы избежать этой нежелательной ситуации, надо подвергнуть  $\mathbf{R}_1$ ,  $\mathbf{R}_2$  и  $\mathbf{R}_3$  процедуре ортогонализации. Она должна гарантировать ортонормированность матрицы поворота  $\mathbf{R}$ . Это можно сделать следующим образом: зафиксировать найденные выше векторы  $\mathbf{R}_1$  и  $\mathbf{R}_3$  и заново вычислить  $\mathbf{R}_2$  по формуле:

$$\mathbf{R}_2 = \mathbf{R}_3 \times \mathbf{R}_1 \quad (13.95)$$

Таким образом, были вычислены все параметры поворота, и они удовлетворяют наложенным ограничениям. Вектор переноса  $\mathbf{T}$  рассчитывается путем ре-



**Рис. 13.27.** Примеры вычисления положения по шести соответствующим точкам с использованием линейной оптимизации при наличии ограничений. (С любезного разрешения Mauro Costa.)

пнения методом наименьших квадратов новой неоднородной переопределенной системы из 12 уравнений:

$$\mathbf{A} \mathbf{t} = \mathbf{b} \quad (13.96)$$

где

$$\mathbf{A} = \begin{pmatrix} f & 0 & -u_1 \\ 0 & f & -v_1 \\ f & 0 & -u_2 \\ 0 & f & -v_2 \\ \vdots & \vdots & \vdots \\ f & 0 & -u_6 \\ 0 & f & -v_6 \end{pmatrix} \quad (13.97)$$

и

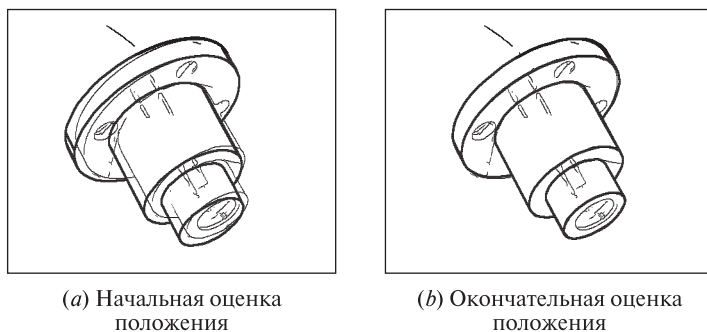
$$\mathbf{b} = \begin{pmatrix} -f(r_{11}x_1 + r_{12}y_1 + r_{13}z_1) + u_1(r_{31}x_1 + r_{32}y_1 + r_{33}z_1) \\ -f(r_{21}x_1 + r_{22}y_1 + r_{23}z_1) + v_1(r_{31}x_1 + r_{32}y_1 + r_{33}z_1) \\ -f(r_{11}x_2 + r_{12}y_2 + r_{13}z_2) + u_1(r_{31}x_2 + r_{32}y_2 + r_{33}z_2) \\ -f(r_{21}x_2 + r_{22}y_2 + r_{23}z_2) + v_1(r_{31}x_2 + r_{32}y_2 + r_{33}z_2) \\ \vdots \\ -f(r_{11}x_6 + r_{12}y_6 + r_{13}z_6) + u_1(r_{31}x_6 + r_{32}y_6 + r_{33}z_6) \\ -f(r_{21}x_6 + r_{22}y_6 + r_{23}z_6) + v_1(r_{31}x_6 + r_{32}y_6 + r_{33}z_6) \end{pmatrix} \quad (13.98)$$

### 13.8.4 Верификация и оптимизация при вычислении положения объектов

Количественные измерения полезны для оценки качества параметров положения. Один пример уже встречался при рассмотрении аффинного метода калибровки камеры: это была сумма квадратов расстояний между точками модели и их образами на изображении. Некоторые соответствующие пары точек надо было игнорировать как выбросы. Некоторые точки модели не имели соответствующих точек на изображении, так как заслонялись другими

точками объекта в его текущем положении. Для измерения расстояний можно использовать другие метрики; например, расстояние Хаусдорфа (см. работы [Hutenlocher и др., 1993], [Dubuisson, Jain, 1984]). Верификация также может быть выполнена с использованием других признаков, например, краев, углов и отверстий.

Качественную меру положения можно использовать для уточнения оценок параметров положения. В принципе, можно организовать вычисления, выполняя малые изменения параметров и в результате выбрать наилучшую комбинацию. Однако метод грубой силы для перебора 10 вариантов каждого из 6 параметров поворота и переноса означает необходимость вычисления более миллиона наборов параметров — а это неприемлемо с вычислительной точки зрения. Методы нелинейной оптимизации, такие, как метод Ньютона или Пауэлла (см. книгу [Vetterling и др., 1993]) оказываются намного быстрее. На рис. 13.28 показана начальная оценка положения для изображения с одним объектом и окончательный результат после выполнения нелинейной оптимизации этого начального решения. Улучшенное положение очевидно лучше подходит для задач визуального контроля и, вероятно, для организации захвата объекта, но, возможно, полученное уточнение не слишком существенно для задач распознавания.



**Рис. 13.28.** Пример уточнения положения объекта методом нелинейной оптимизации. (С любезного разрешения Mauro Costa.)

### 13.9 РЕКОНСТРУКЦИЯ ТРЕХМЕРНЫХ ОБЪЕКТОВ

Восприятие трехмерных сцен важно для построения трехмерных моделей. Мы можем сформировать дальнометрические изображения существующих объектов, чтобы создать их компьютерные модели. Этот процесс *реконструкции трехмерных объектов* имеет приложения в медицине и промышленных зрительных системах, а также применяется для создания моделей объектов для сред виртуальной реальности. По необходимости в данную главу включено некоторое обсуждение моделирования объектов, хотя это будет основной темой

следующей главы. Процесс реконструкции объекта можно разделить на четыре основных этапа:

1. сбор трехмерных (дальнометрических) данных;
2. совмещение собранных дальнометрических данных;
3. построение поверхности;
4. оптимизация.

На этапе сбора данных требуется получить данные о дальности для некоторого набора различных видов, которые содержат всю поверхность объекта. Часто достаточно 8–10 видов, но для сложных объектов и для обеспечения высокой точности необходимо большее количество видов. Конечно, чем больше видов, тем больше вычислений, так что увеличение количества видов не всегда оправданно для решения конкретной задачи.

Каждый полученный вид состоит из одного дальнометрического изображения части поверхности объекта и часто еще из полутонового или цветного изображения. Дальнометрические данные от всех видов должны быть скомбинированы для получения модели поверхности объекта. Данные интенсивности могут быть использованы в процедуре совмещения, но на практике они обычно используются в качестве текстур для последующего реалистичного отображения в графических приложениях. Процесс комбинирования дальнометрических данных путем приведения их к единой трехмерной системе координат называется *совмещением* (registration).

После совмещения видов можно отобразить *скопление трехмерных точек*, но для построения модели объекта требуется дальнейшая обработка. Возможны два трехмерных представления объекта: (1) связный каркас, состоящий из трехмерных точек, соединенных ребрами для представления поверхности объекта и (2) множество трехмерных кубиков, называемых *вокселями*, для представления всего объема, занимаемого объектом. (Подробное описание этих способов представления дано в гл. 14.) Возможно преобразование из одной формы представления в другую.

---

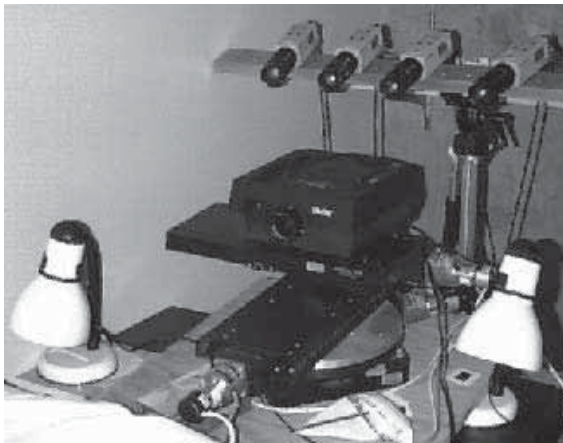
### Упражнение 13.31. Объекты с невидимыми поверхностями

У некоторых объектов есть скрытые поверхности, которые нельзя увидеть на изображении, независимо от количества используемых видов. В качестве примера изобразите один подобный объект. Для простоты можете привести пример двумерного объекта в двумерном пространстве.

#### 13.9.1 Сбор дальнометрических данных

Дальнометрические (дальностные) изображения, совмещенные с цветными, можно получить с помощью нескольких различных коммерческих моделей сканеров. Ниже будет описана лабораторная система из готовых компонент и рассмотрены наиболее важные операции, выполняемые этой системой. На

рис. 13.29 показана специально собранная активная зрительная стереосистема для получения дальнометрических и цветовых данных. В состав установки входят 4 цветные видеокамеры, укрепленные на алюминиевом стержне. Камеры подсоединены к плате-фреймграбберу, у которого есть управляемый от компьютера коммутатор на 4 видеовхода. Фреймграббер позволяет получать цифровые изображения с разрешением  $640 \times 480$  пикселей. Под камерами на управляемом от компьютера поворотном столе установлен слайд-проектор. Слайд-проектор излучает вертикальную белую полосу, которая вручную фокусируется в рабочем объеме пространства. Световые полосы проектируются в затемненной комнате. Два боковых источника света включаются для получения цветного изображения после получения дальнометрических данных.

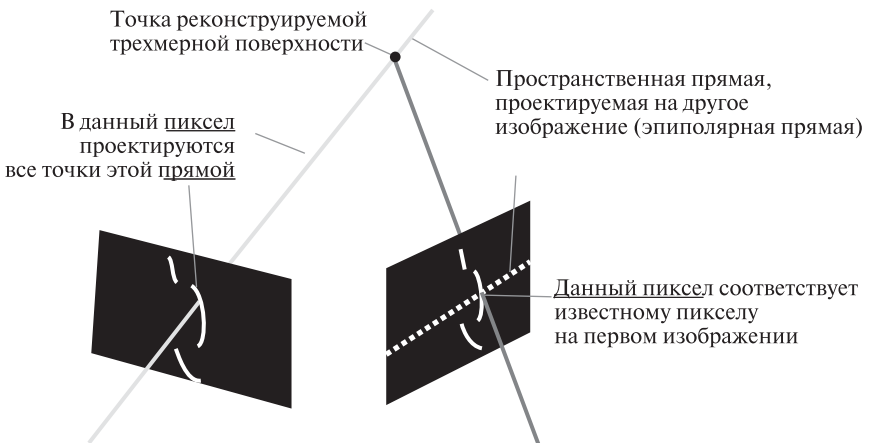


**Рис. 13.29.** Стереосистема из 4 камер для получения дальнометрических и цветовых данных. (С любезного разрешения Kari Pulli.)

Камеры одновременно калиброваны методом Цая ([Tsai, 1987]; см. раздел 13.7). Они могут использоваться в стандартном режиме с двумя камерами или в более надежном стереоалгоритме с четырьмя камерами. В любом случае, проектор используется для проектирования одиночной вертикальной светлой полосы на сканируемый объект. Под управлением компьютера сканирование начинается с левой стороны объекта. Полоса, с помощью поворотного стола, смещается с фиксированным шагом слева направо. Оператор может задать шаг для выбора грубого или точного разрешения сканирования. В каждой позиции полосы камеры в затемненной комнате снимают изображения, на которых на объекте видна светлая полоса. На каждом изображении пересечение этой полосы с эпполярной прямой дает точку для стереосовмещения. На рис. 13.30 поясняется процесс триангуляции с использованием двух камер и одной световой полосы. Два соответствующих пиксела на двух изображениях используются для вычисления одной точки трехмерного пространства. В одном

положении световой полосы выполняется вычисление трехмерных точек для всех пикселей, освещаемых полосой. Затем проектор поворачивается, полоса проектируется в новое положение, вводятся новые изображения и процесс повторяется. В результате получается плотная карта дальнометрических данных, в которой с каждым пикселем левого изображения связана трехмерная точка (если только эта точка видна также правой камере и она принадлежит световой полосе).

Надежность системы получения изображений можно повысить путем увеличения количества камер. Одна камера будет базовой, в ее системе координат будет вычисляться дальнометрическое изображение. Точка на поверхности объекта должна быть видима базовой камерой, световым проектором, и как минимум одной из трех других камер. Если она видима только одной из трех камер, то процесс обработки сводится к обычной обработке в стереосистеме с двумя камерами. Если точка видима двумя или тремя камерами из трех, то дополнительные изображения можно использовать для повышения надежности обработки. В случае, если точка видима на изображении базовой камеры и еще двух камер, то для вычисления трехмерных координат будут получены три пары соответствующих точек. Весьма вероятно, что три отдельно вычисленных результата будут отличаться. Если они отличаются «немного» (например, попадают внутрь объема в семь кубических миллиметров), то все результаты считаются корректными и в качестве окончательного результата — трехмерных координат пиксела — берется среднее значение координат. Или, результат для двух камер с максимальным базовым расстоянием можно считать наиболее надежной оценкой по сравнению с остальными. Если точка видна всеми четырьмя камерами, то будут получены 6 пар соответствующих точек. Снова можем



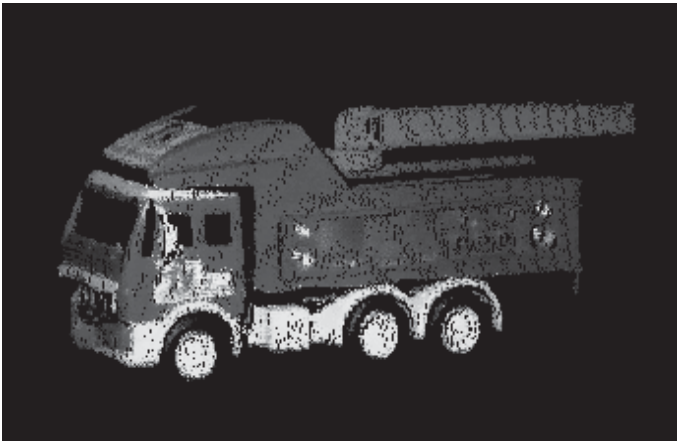
**Рис. 13.30.** Пересечения световых полос с эпиполярными прямыми на двух изображениях дают пару соответствующих точек. (С любезного разрешения Кари Pulli.)



проверить, попадают ли они в малый объем, исключить из рассмотрения выбросы и вычислить среднее из оставшихся результатов или взять результат для пары камер с наибольшим базовым расстоянием. Описанная процедура обеспечивает более высокую точность, чем обработка в обычной системе с двумя камерами. (При измерениях позы человека внутри автомобиля, как на рис. 13.1, ожидаемая ошибка по  $x$ ,  $y$  и  $z$  составляла примерно 2 мм.) На рис. 13.31 показано дальнометрическое изображение игрушечного грузовика, полученное с помощью описанного метода. Набор трехмерных данных для грузовика дает ясное представление о его форме.

### 13.9.2 Совмещение видов

Для полного охвата поверхности объекта надо получить дальнометрические данные для нескольких видов. Преобразование  ${}^2_1T$  из вида 1 в вид 2 получается или за счет точного учета механического перемещения, или посредством определения соответствия изображений. Когда имеется высокоточное устройство для контролируемого перемещения камеры или объекта (например, калиброванный робот или аппарат для измерения координат), то приближенные преобразования между видами в зрительной системе можно получить автоматически. Если перемещение камер или объекта не управляется компьютером, то должен быть предусмотрен метод для определения соответствия между видами, который будет позволять вычислять твердотельное преобразование для отображения данных одного вида на другой вид. Это преобразование можно вычислять автоматически с использованием трехмерных характерных признаков, таких, как угловые точки и отрезки. Они дают некоторое множе-



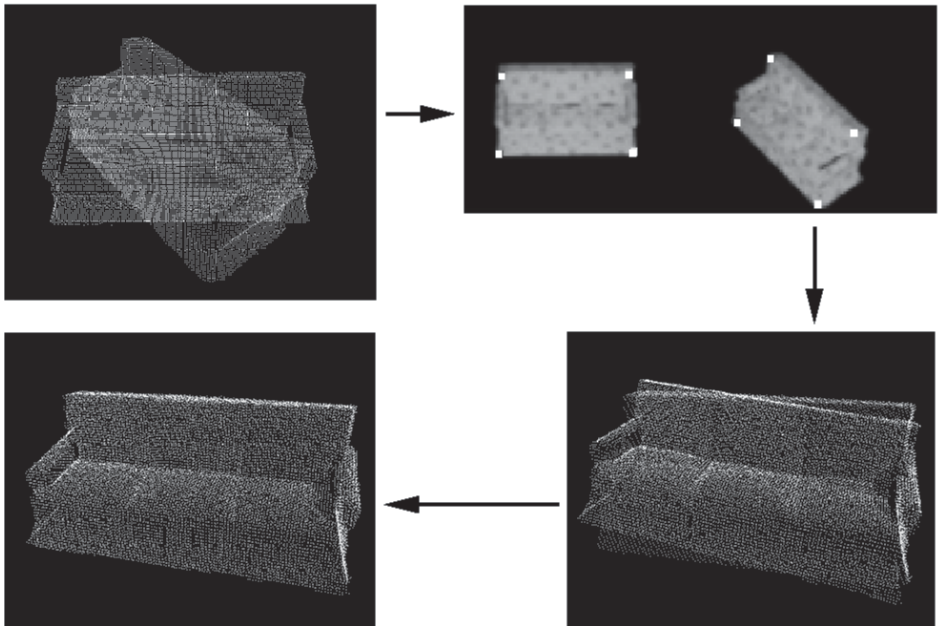
**Рис. 13.31.** Множество дальнометрических данных для игрушечного грузовика, полученное с помощью активной стереосистемы из 4-х камер. Дальность точек обозначается на дисплее с помощью интенсивности цвета. (С любезного разрешения Habib Abi-Rached.)

ство соответствующих пар трехмерных точек, по которым можно вычислить преобразование. Вычисление преобразования можно выполнить интерактивно, давая пользователю возможность указать соответствующие точки на паре изображений объекта. В любом случае, начальная оценка преобразования редко получается удовлетворительной. Роботы и аппаратные устройства перемещения имеют некоторую ошибку перемещения, которая будет увеличиваться с каждым последующим перемещением. Автоматический определитель соответствия будет страдать от проблем, присущих любому алгоритму сравнения. Он может находить ложные пары соответствующих точек, или характерные точки могут немного отличаться от оптимальных. Даже пользователь указывает точки с некоторой погрешностью, или дискретизация приводит к взятию координат соседнего пиксела. Все это приводит к ошибкам при вычислении параметров преобразования.

Для решения перечисленных проблем большинство процедур совмещения реализуют итеративную схему, которая начинается с начальной оценки преобразования  ${}^2_1\mathbf{T}$ , полученной любым способом, и затем уточняет ее с помощью процедуры минимизации. Например, итеративный алгоритм ближайшей точки (iterative closest point, ICP) минимизирует сумму расстояний между трехмерными точками  ${}^2_1\mathbf{T} \mathbf{1P}$  и  $\mathbf{2P}$ , где точка  $\mathbf{1P}$  принадлежит одному виду, а  $\mathbf{2P}$  — ближайшая к ней точка на другом виде. Известен вариант этого метода, который ищет на втором виде точку вдоль нормали, восстановленной из  ${}^2_1\mathbf{T} \mathbf{1P}$  к поверхности, интерполированной по точкам окрестности на другом виде (см., например, работы [Chen и Medioni, 1992], [Dorai и др., 1994]). Если доступны цветные данные, то с помощью оценочного преобразования можно спроектировать цветные данные первого вида на цветные данные другого и определить меру расстояния между изображениями исходя из того, насколько хорошо они оказались совмещены. Это расстояние также можно минимизировать итерационно, что приведет к уточнению оценки преобразования одного множества трехмерных точек в другое. На рис. 13.32 поясняется процесс совмещения двух видов дивана с помощью алгоритма ICP.

### 13.9.3 Реконструкция поверхности

После совмещения дальнометрических данных все они будут приведены к одной системе координат. После этого может начинаться этап реконструкции поверхности. Желательно, чтобы реконструированный объект по форме максимально близко был похож на реальный наблюдаемый объект и передавал его топологию. На рис. 13.33 показана проблема, которая может возникнуть в процессе реконструкции. Совмещенные данные плотны, но довольно сильно зашумлены. Присутствуют лишние точки, лежащие за пределами объема реального стула и, в особенности, между перекладинами спинки. В центре показан результат примитивной реконструкции, рассматривающей дальнометрические данные просто как скопление трехмерных точек и не учитывающей геометрии объекта или соотношений, связывающих соседние точки в

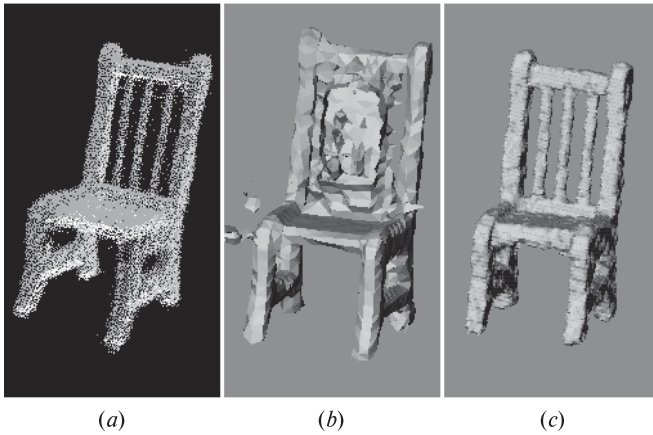


**Рис. 13.32.** Слева сверху показано совмещение двух наборов дальнометрических данных. Пользователь указал 4 точки на полутоновых изображениях, соответствующих двум дальнометрическим видам (вверху справа). Начальная оценка преобразования слегка смещена (снизу справа). После нескольких итераций два дальнометрических вида совмещаются удовлетворительно (снизу слева). (С любезного разрешения Kari Pulli.)

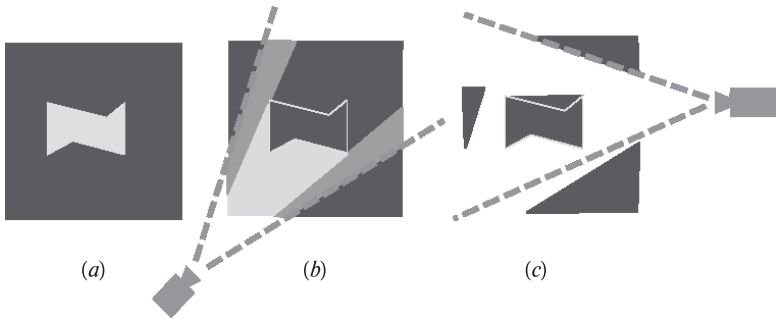
дальнометрических данных. Этот результат неправильно передает топологию объекта. Результат реконструкции, показанный справа, лучше, так как в нем устранено большинство шумовых элементов и присутствуют отверстия между планками спинки. Эта реконструкция выполнена методом *объемного удаления*, который описывается далее.

#### 13.9.4 Алгоритм объемного удаления

Подход объемного удаления (space carving) был представлен в работе [Curless, Levoy, 1996], а описанный ниже метод был реализован описан в статье [Pulli и др., 1998]. На рис. 13.34 поясняется основной принцип алгоритма. Слева показан объект, который надо реконструировать по набору видов. В центре показана одна из камер, получающих один вид объекта. Пространство может быть разбито на области согласно расположению точек относительно объекта и камеры. Камере видны точки слева и снизу от объекта. Объем между сканируемой поверхностью и камерой (светло-серый) расположен перед объектом и может быть удален. Если есть данные от фона, так же как и



**Рис. 13.33.** (a) Совмещенные дальнометрические данные для объекта-стула. (b) Проблемы, возникающие в процессе реконструкции поверхности. (c) Топологически корректная грубая каркасная модель. (С любезного разрешения Kari Pulli.)



**Рис. 13.34.** Принцип объемного удаления: (a) поперечное сечение объекта; (b) вид с камеры 1 позволяет удалить весь материал со светлым затенением (c) вид с камеры 2 позволяет удалить остальной материал (Тем не менее еще остается лишний материал). (С любезного разрешения Kari Pulli.)

от объекта, то можно удалить дополнительные объемы (темно-серый). Точки позади объекта удалить нельзя, так как одна камера не может дать данных, чтобы решить, принадлежат эти точки объекту или нет. Однако вид объекта с другой камеры (рис. 13.34 (c)) позволяет удалить другие объемы пространства. С увеличением количества видов удается удалить большее количество лишнего объема пространства, оставляя только воксельную модель объекта.

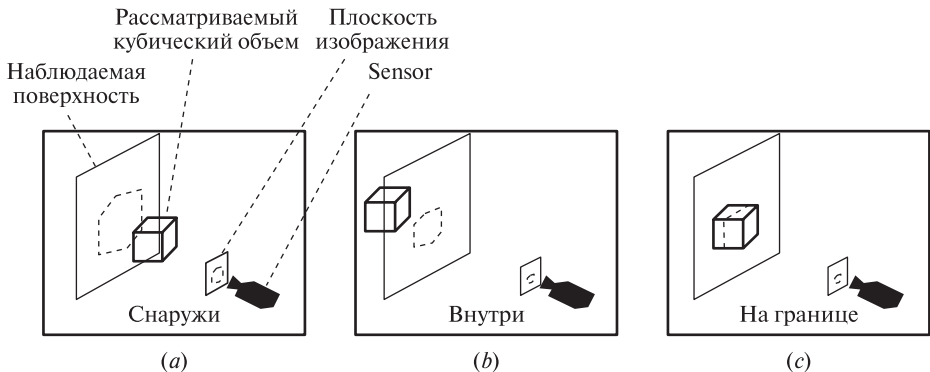
В алгоритме объемного удаления предполагается разбиение пространства на множество элементарных кубических элементов (вокселей), которые можно обрабатывать одновременно. На рис. 13.35 показано, как можно определить

состояние одного выбранного кубического объема по отношению к виду от одной камеры.

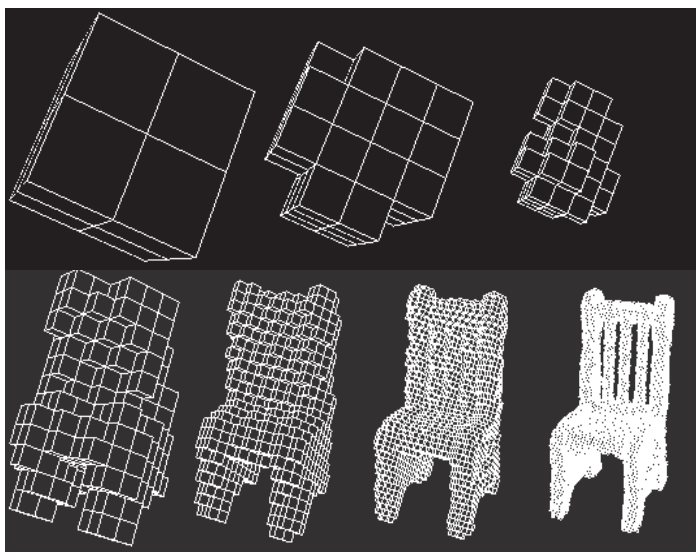
- в случае (а) куб лежит между камерой и дальнометрическими данными. Значит, куб должен лежать за пределами объекта и его можно удалить.
- в случае (б) куб целиком лежит позади дальнометрических данных. С учетом положения камеры, куб считается лежащим внутри объекта.
- в случае (в) куб лежит частично позади, частично перед данными. Тогда он считается пересекающим поверхность объекта.

Маркировку единичных кубов можно выполнить следующим образом. Восемь угловых точек куба проектируются на плоскость изображения, и выпуклая оболочка этих точек обычно образует шестиугольник. Лучи, проведенные из центра камеры через шестиугольник, образуют конус. Он усекается так, чтобы точно охватывал куб. Если все точки данных, спроектированные на шестиугольник, лежат позади усеченного конуса (т. е. точки расположены от камеры дальше, чем самый далекий угол куба), то куб считается лежащим за пределами объекта. Если все эти точки лежат ближе самого ближнего угла куба, то куб считается лежащим внутри объекта. Иначе, куб лежит на границе объекта.

До сих пор мы рассматривали один кубический объем и один датчик (камеру). Для удаления пустого пространства требуется несколько датчиков (или видов). На этапе маркировки кубов обработка каждого куба выполняется для каждого датчика. Если хотя бы один датчик говорит, что куб лежит за пределами объекта, то этот куб считается лежащим за пределами объекта. Если все датчики говорят, что куб лежит внутри объекта, то он считается пока находящимся внутри объекта. На одном из других видов может оказаться, что куб в действительности лежит вне объекта, но пока у нас не было обработано



**Рис. 13.35.** Три возможных положения кубического объема в пространстве относительно реконструируемого объекта. (С любезного разрешения Kari Pulli.)



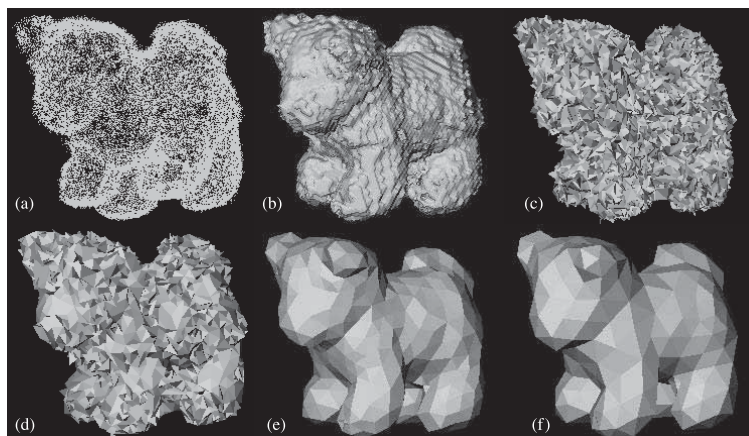
**Рис. 13.36.** Иерархическое объемное удаление: получение каркасной модели стула после семи итераций. (С любезного разрешения Kari Pulli.)

этого вида. В третьем случае, если куб не находится ни внутри, ни снаружи объекта, то он считается граничным.

Вместо использования кубов фиксированного размера более эффективным может оказаться иерархическое выполнение маркировки кубов, с использованием специальной структуры данных — *октантного дерева*<sup>1</sup> (*octree*). Эта структура данных будет подробно описана в следующей главе, но мы воспользуемся ею и здесь, на интуитивно понятном уровне. Вначале все данные считаются вписанными в один большой куб. Поскольку он, по определению, пересекается с данными, то он делится на восемь кубов меньшего размера. Из этих кубов часть отбрасывается, если они лежат за пределами объекта, а те, которые лежат полностью внутри объекта, помечаются как часть объекта. Граничные кубы продолжают разбиваться, и этот процесс продолжается до достижения требуемого разрешения. Результирующее октантное дерево служит представлением трехмерного объекта. На рис. 13.36 поясняется иерархическая процедура объемного удаления для построения модели объекта-стула.

Для удобства отображения представление объекта в виде октантного дерева может быть преобразовано в каркасную модель (см. рис. 13.36). После создания начального каркаса, он может быть оптимизирован с помощью метода, упрощающего каркас и лучше сглаживающего данные. На рис. 13.37 показаны совмещенные дальнометрические данные для объекта — игрушечной

<sup>1</sup> Эта структура данных также встречается под названием «октарное дерево». — Прим. перев.



**Рис. 13.37.** Совмещенные дальнометрические данные и 5 этапов формирования каркасной модели игрушечной собаки. (С любезного разрешения Kari Pulli.)

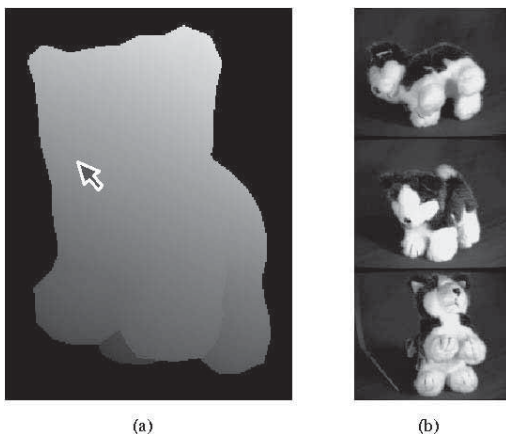
собаки (a), исходная оценка каркасной модели (b) и несколько шагов оптимизирующей процедуры по методу, описанному в работе [Норре и др., 1992] (c)–(f). Окончательная каркасная модель (f) намного точнее и более гладкая, чем исходная. Она может быть использована в графической системе для генерации реалистичных изображений объекта (см. рис. 13.38) или в системе распознавания объектов на основе моделей (см. гл. 14).

### 13.10 ВЫЧИСЛЕНИЕ ФОРМЫ ПО ДАННЫМ ОСВЕЩЕННОСТИ

В гл. 6 и 12 обсуждалось отражение света от гладких криволинейных поверхностей объектов и формирование полутоновых изображений этих объектов. Ниже кратко будет рассмотрен вопрос о том, как при некоторых допущениях можно вычислить форму объекта по данным освещенности, взятым с изображения этого объекта.

Зрение человека устроено так, что он склонен считать плавно затененные поверхности удаляющимися от направления взгляда. Восприятие человеческого лица другими людьми можно изменить с помощью грима. Грим темнее кожи на внешней стороне щек делает лицо «уже», так как более темный цвет заставляет других людей думать, что эти поверхности отклонены от наблюдателя сильнее, чем на самом деле. Похожим образом грим светлее кожи приводит к обратному эффекту, создавая у наблюдателя впечатление полноты лица. С помощью формулы отражения по закону Ламберта можно связать интенсивность (освещенность) элемента поверхности и направление нормали к поверхности в этом месте. В работе [Horn, Bachman, 1978] изучались методы определения рельефа лунной поверхности, наблюдаемой с известного расстояния с Земли и освещенной с известного расстояния Солнцем. Семей-





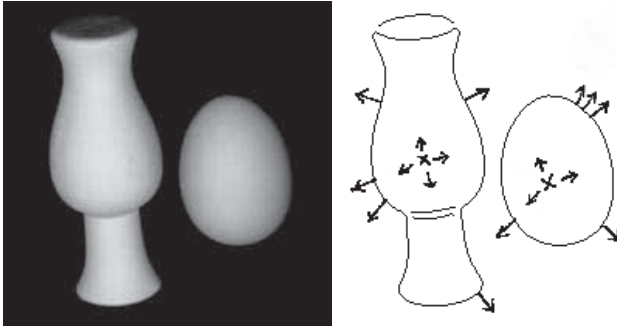
**Рис. 13.38.** (а) Изображение модели игрушечной собаки в псевдоцветах, манипулируя которым пользователь может указать желаемый вид. (б) Трехмерная точка на носу модели, отмеченная указателем, проектируется на 3 цветных изображения собаки для выбора пикселей, которые можно использовать для вычисления реалистичного цвета при отображении модели собаки на экране. (С любезного разрешения Kari Pulli.)

ство методов, которые применяются для расчета нормалей к поверхности по данным о ее освещенности, называются *методами для вычисления формы по освещенности поверхности* (shape from shading, SFS).

**109 Определение.** Метод для вычисления формы по освещенности поверхности вычисляет нормаль к точке поверхности  $\mathbf{n} = f(x, y)$  по данным об интенсивности ее освещенности, полученным с изображения поверхности.  $\mathbf{n}$  — это нормаль к точке поверхности, соответствующей точке изображения  $(x, y)$ , а  $^F I(x, y)$  — интенсивность освещенности этой точки изображения.

Способ вычисления формы по освещенности поверхности поясняется на рис. 13.39. Слева показано изображение объектов, хорошо описываемых законом отражения Ламберта: интенсивность освещенности на изображении пропорциональна углу между нормалью к поверхности и направлением падающего света. Справа на рис. 13.39 показаны нормали к поверхности в нескольких точках на поверхности объектов. Хорошо видно, что более яркие точки изображения соответствуют элементам поверхности, нормали к которым направлены прямо на источник света: нормали к поверхности, направленные «прямо на наблюдателя», на рис. 13.39, справа, отмечены крестиками. В точках лимба нормали к поверхности перпендикулярны и к направлению наблюдения, и к границе поверхности на изображении. Эти условия строго ограничивают нормаль в трехмерном пространстве. Используя эти ограничения, можно распространить векторы нормали на все точки поверхности. Таким образом





**Рис. 13.39.** (Слева) полутоновое изображение объектов с ламбертовым отражением, полученное при освещении источником, расположенным около камеры. (Справа) нормали к поверхности, показанные в нескольких точках на границе контуров объектов. Конттуры были выделены оператором Кэнни.

можно построить частичное внутреннее изображение. Для получения глубины  $z$  в каждой точке изображения мы можем присвоить произвольное значение  $z_0$  одной из наиболее ярких точек и затем распространить значения глубины вдоль изображения, учитывая изменение направлений нормалей.

---

### Упражнение 13.32

Предположим, что куб, поверхность которого описывается ламбертовым отражением, расположен так, что нормали всех граней составляют с направлением освещения угол не менее  $\pi/6$ . Очевидно, наиболее яркие точки изображения не будут соответствовать нормалям, направленным к источнику света. Почему это верно для вазы и яйца, но неверно для куба?

---

### Упражнение 13.33

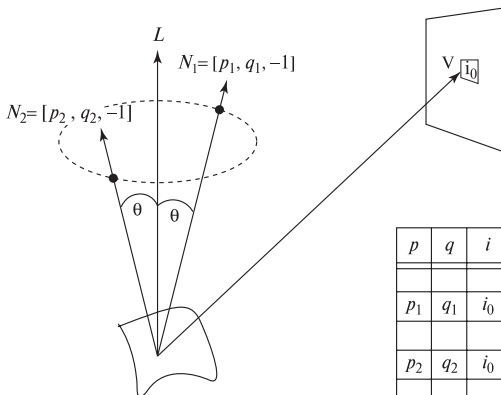
Как по данным об освещенности определить, является граница объекта лимбом или лезвием?

Модель ламбертова отражения записывается в виде  $i = c \cos \theta$ , где константа  $c$  зависит от мощности источника света, альбедо поверхности и расстояний между элементом поверхности и источником света и датчиком. Все перечисленные факторы считаются постоянными. Предположим, что расстояния от объекта до источника света и датчика многократно превышают его диаметр. Часто также предполагается, что направление падающего света известно, однако, как отмечалось выше, направление света иногда можно вычислить с более слабыми предположениями.

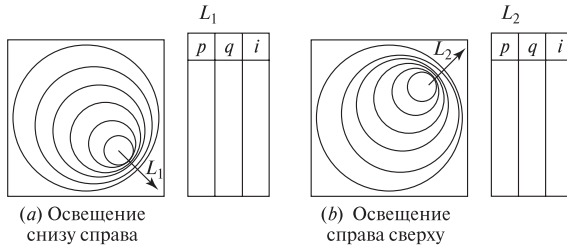
Для дальнейшего изложения наиболее удобна прямоугольная проекция. Также будем пользоваться единственной системой координат — трехмерной

системой координат камеры. Наблюдаемая поверхность представляется в виде  $z = f(x, y)$ : задача заключается в вычислении функции  $f$  для каждой точки изображения исходя из наблюдаемых значений интенсивности  $^F I(x, y)$ . Переписывая и дифференцируя уравнение поверхности, получим:  $\frac{\partial f}{\partial x} \Delta x + \frac{\partial f}{\partial y} \Delta y - \Delta z = 0$  или, в векторной форме:  $[p, q, -1] \circ [\Delta x, \Delta y, \Delta z] = 0$ , где  $p$  и  $q$  — частные производные функции  $f$  по координатам  $x$  и  $y$ , соответственно. Последнее уравнение определяет касательную плоскость в точке поверхности  $(x, y, f(x, y))$ , вектор нормали к которой (неединичный) равен  $(p, q, -1)$ . Если известно, что точка  $(x_0, y_0, z_0)$  принадлежит поверхности, а также известны  $p$  и  $q$ , то вышеприведенная аппроксимация плоскостью позволяет вычислить точку поверхности  $(x_0 + \Delta x, y_0 + \Delta y, z_0 + \Delta z)$  просто путем смещения по касательной плоскости. Мы сможем сделать это, если с учетом наших допущений сможем оценить  $p$  и  $q$  по данным интенсивности с изображения.

Наблюдая известный объект, можно связать векторы нормали к поверхности с интенсивностью пикселей изображения. Всякий раз, когда в точке  $(x, y, f(x, y))$  известна ориентация поверхности  $(p, q, -1)$  и наблюдаемая интенсивность  $I(x, y)$ , то можно помещать тройку  $\langle p, q, I(x, y) \rangle$  в карту отображения, связывающую ориентацию поверхности с интенсивностью ее освещения на изображении. На рис. 13.40 показано, что это отображение типа «многие-к-одному»: все элементы поверхности, составляющие конус с углом  $\theta$  с направлением освещения в качестве оси, будут давать на изображении одинаковую наблюдаемую интенсивность. Наилучшим объектом для подобной калибровки является сфера, так как (а) на ней есть элементы поверхности с любой ориентацией и (б) нормали к поверхности известны по положению точки изображения относительно центра сферы, если ее радиус известен. На рис. 13.41 показан результат, видимый при калибровке со сферой с двумя источниками света. Для каждого источника света можно построить карту



**Рис. 13.40.** (Слева) конус возможных ориентаций нормали, имеющих на изображении одинаковую интенсивность. (Справа) карта отображения связывает векторы нормали со значениями интенсивности: это отображение типа «многие-к-одному».



**Рис. 13.41.** Карты отражения можно построить с помощью калибровки с использованием сферы с ламбертовым отражением из того же материала, что и наблюдаемый объект. На наблюдаемом изображении сферы из аналитической геометрии  $p$  и  $q$  известны для каждой точки  $I(x, y)$ : таким образом, мы можем занести в таблицу тройку  $\langle p, q, I[x, y] \rangle$  для каждой точки изображения. Несколько карт отражения формируются для различных источников света.

отражения, в которой хранится наблюдаемая интенсивность и набор всех ориентаций поверхности, которым соответствует данная интенсивность на изображении.

### Упражнение 13.34

Калибровочная сфера радиусом  $r$  расположена в точке  $(0, 0, 100)$  в системе координат камеры  $\mathbf{C} : (x, y, z)$ . Получите выражения для  $p$  и  $q$  в координатах изображения  $(x, y)$ . Напомним, что прямоугольная проекция фактически сводится к удалению координаты  $z$ .

На рис. 13.39 и 13.41 интенсивность пикселей изображения дает сильное ограничение на ориентацию поверхности, но не определяет нормаль к поверхности уникальным образом. Требуются дополнительные ограничения. Существуют два основных подхода. Первый подход использует информацию о пространственном соседстве; например, пиксел и его 4-связные соседи дают 5 экземпляров уравнения освещенности, которые можно совместно учитывать для вычисления гладкой поверхности, проходящей через эти точки. Второй подход использует несколько полутоновых изображений, чтобы можно было записать несколько уравнений для каждого пиксела независимо от его соседей: этот подход по очевидным причинам называется *стереофотометрическим*.

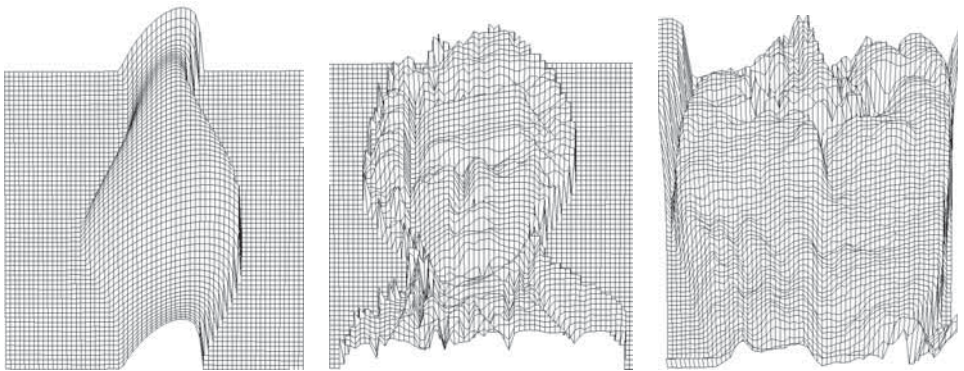
#### 13.10.1 Стереофотометрический метод

В стереофотометрическом методе обрабатывается набор изображений, полученных при освещении объекта различными источниками света. Таким образом, для каждого пиксела получается набор значений интенсивности, и для них составляется таблица с указанием вектора нормали к поверхности. Таблица строится с помощью предварительно выполняемой процедуры фотометриче-

ской калибровки (см. рис. 13.41). Содержание этой процедуры приведено в алгоритме 13.4. Стереофотометрический метод обеспечивает высокую скорость вычислений в контролируемой лабораторной среде. В работе [Ray и др., 1983] сообщается об успешных результатах даже при работе с зеркально отражающими объектами при использовании трех сбалансированных источников света. Однако, если окружающая среда недостаточно управляема, то для трехмерных измерений вместо вычисления формы по освещенности лучше применять структурный подсвет. Этот подход в последнее время стал преобладающим в прикладных задачах.

### 13.10.2 Интегральный учет пространственных ограничений

Для определения функции гладкой поверхности  $z = f(x, y)$  было предложено несколько методов, основанных на учете априорно известных ограничений, накладываемых на значения освещенности соседних пикселей поверхности. Один из этих методов состоит в наращивании поверхности, начиная с наиболее ярких точек изображения, как отмечалось выше. Процедура минимизации находит *лучшее* решение, удовлетворяющее наложенным ограничениям. На рис. 13.42 показан результат работы подобного алгоритма: для двух искусственных и одного реального объектов показана каркасная модель вычисленной поверхности. Эти результаты могут считаться как хорошими, так и плохими, в зависимости от задачи, в которой нужны эти данные. Но в целом этот метод не считается достаточно надежным для практического использования.



**Рис. 13.42.** Результаты применения алгоритма Цая—Ша (Tsai—Shah) к двум искусственно сформированным изображениям и одному реальному изображению. (Слева) поверхность, вычисленная алгоритмом на основе изображения, сгенерированного САПР-программой для модели вазы с диффузным освещением. (В центре) поверхность, вычисленная для искусственно сформированного изображения бюста Моцарта. (Справа) поверхность для реального изображения зеленого перца. (С любезного разрешения Mubarak Shah.)

Задачи класса «форма по данным освещенности» показывают, что информация об освещенности дает жесткие ограничения на форму поверхности. Это отличный пример «чистой» задачи компьютерного зрения — в ней очень хорошо определены входные и выходные данные, а также используемые допущения. Многие математические алгоритмы хорошо работают в определенных специфических случаях. Но практически ни один не дает хороших результатов на широком спектре сцен. Читатель может обратиться за более подробной информацией по этому вопросу к справочной литературе, особенно за деталями математических алгоритмов, которые в данном разделе обозначены только схематично.

### 13.11 ГЕОМЕТРИЧЕСКАЯ СТРУКТУРА ОБЪЕКТОВ ПО ДАННЫМ О ДВИЖЕНИИ

Человек воспринимает много информации о структуре трехмерных сцен за счет движения. Когда перемещается наблюдатель или объект, или они оба, то наблюдатель получает информацию из последовательности изменяющихся изображений. По векторам оптического потока или соответствующим точкам на трехмерных сценах можно восстановить поверхности и углы, а также определить траекторию движения датчика через наблюдаемую сцену. Этот интуитивно понятный процесс можно формализовать и получить целый класс специфических математических задач. Проектирование практически полезных алгоритмов компьютерного зрения для вычисления структуры сцены при движении объекта и/или наблюдателя оказалось сложным делом.

На рис. 13.43 показана общая ситуация, когда могут двигаться и объект, и наблюдатель. Относительное движение объектов и наблюдателя дает на изображении векторы потока. Они могут быть вычислены по соответствию точек или на основе оптического потока. На рис. 13.44 показан случай с двумя существенно различными видами 5 трехмерных точек. В опубликованных работах описано много случаев, различающихся и постановкой задачи, и предлагаемым алгоритмом.

---

#### Упражнение 13.35. Усовершенствование алгоритма 13.4

Измените алгоритм 13.4 за счет переноса вычисления операций пересечения множеств в предварительно выполняемую часть. Объясните, почему это возможно. Какая структура данных подходит для хранения результатов этих вычислений, чтобы ими можно было в дальнейшем пользоваться в той части алгоритма, которая выполняется в процессе штатной обработки?

В постановке задачи могут использоваться следующие трехмерные объекты:

- точки
- прямые

Вычисление нормалей к поверхности  $(p, q)$  для точек сцены, видимых на различных изображениях  ${}^1\mathbf{I}, {}^2\mathbf{I}, {}^3\mathbf{I}$ , полученных с использованием различных источников света  ${}^1\mathbf{L}, {}^2\mathbf{L}, {}^3\mathbf{L}$ .

**Предварительно выполняемая калибровка:**

1. Поместить калибровочную сферу в центр сцены.
2. Для каждого из трех источников света  ${}^j\mathbf{L}$ .
  - (а) Включить источник  ${}^j\mathbf{L}$ .
  - (б) Получить изображение калибровочной сферы
  - (в) Сформировать карту отражения  ${}^j\mathbf{R} = \{(p, q, {}^j\mathbf{I}(\mathbf{x}, \mathbf{y}))\}$ , где пара  $(p, q)$  связана с некоторым значением интенсивности  ${}^j\mathbf{I}(\mathbf{x}, \mathbf{y})$  на изображении  ${}^j\mathbf{I}$ .

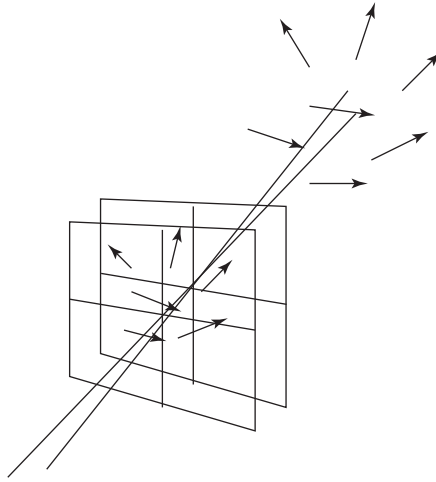
**Операции, выполняемые в процессе штатной обработки для получения данных о трехмерной поверхности**

1. Поместить наблюдаемый объект в центр сцены.
2. Последовательно получить три отдельных изображения  ${}^j\mathbf{I}$  в условиях освещения соответствующим источником света  ${}^j\mathbf{L}$ .
3. Для каждой точки изображения  $(\mathbf{x}, \mathbf{y})$ 
  - (а) Использовать значение интенсивности  $i_j = {}^j\mathbf{I}(\mathbf{x}, \mathbf{y})$  в качестве индекса карты отражения  ${}^j\mathbf{R}$  и получить множество пар  $\mathbf{R}_j = \{(p, q)\}$ , связанных со значением интенсивности  $i_j$ .
  - (б) Вычислить пересечение трех множеств:  $\mathbf{S} = \mathbf{R}_1 \cap \mathbf{R}_2 \cap \mathbf{R}_3$ .
  - (в) Если множество  $\mathbf{S}$  пусто, то принять  $\mathbf{N}(\mathbf{x}, \mathbf{y}) = NULL$ , иначе в качестве  $\mathbf{N}(\mathbf{x}, \mathbf{y})$  взять направление вектора, усредненное по  $\mathbf{S}$ .
4. В качестве результата вернуть значение  $\mathbf{N}(\mathbf{x}, \mathbf{y})$ , которое можно считать частью внутреннего изображения, в котором хранятся нормали к поверхности наблюдаемого объекта.

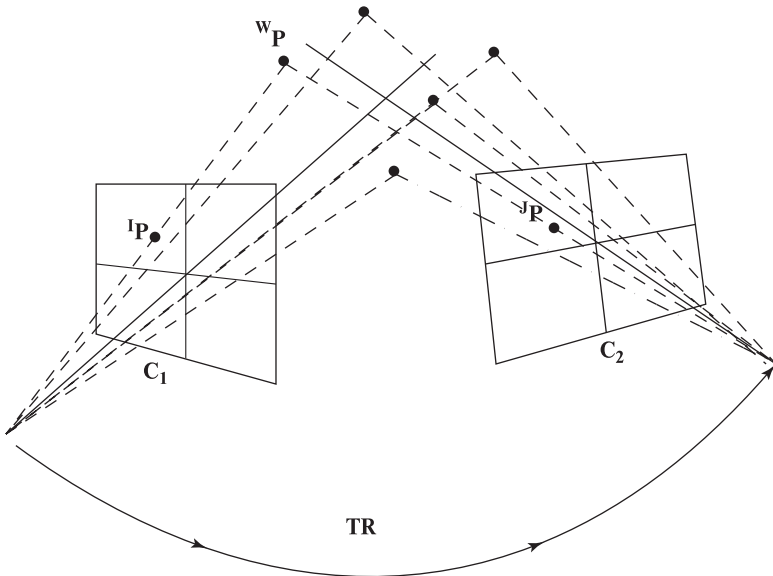
**Алгоритм 13.4.** Стереометрический метод с использованием трех источников света

- куски плоских поверхностей
- куски криволинейных поверхностей

С учетом сделанных допущений, алгоритм должен определять не только трехмерную структуру объекта, но также параметры его движения относительно системы координат камеры. Много работ по проектированию алгоритмов было выполнено в предположении, что можно обеспечить надежное определе-



**Рис. 13.43.** Наблюдатель перемещается через сцену, объекты которой тоже движутся. Движение трехмерных точек представляется в виде двумерных векторов потока, связывающих два близких по времени изображения.



**Рис. 13.44.** Наблюдатель перемещается через сцену с неподвижными объектами. Трехмерные точки  ${}^wP$  проецируются в двумерные точки  ${}^I_1P$  и  ${}^I_2P$  на двух изображениях, существенно отличающихся по времени и пространству, так что найти соответствующие точки может быть сложно. Если соответствующие точки найдены, то задача заключается в вычислении относительного движения  $TR$  и трехмерных координат точек  ${}^wP$ .

ние трехмерных координат точек и поиск соответствующих точек. На практике оказывается, что получение трехмерных координат и поиск соответствия является сложной и подверженной ошибкам процедурой. Алгоритмы, основанные на оптическом потоке, используют небольшие временные промежутки между изображениями и пытаются вычислить плотные трехмерные структуры. Напротив, алгоритмы на основе соответствия характерных признаков, оказываются способными работать с большими временными промежутками, но вычисляют только разреженные трехмерные структуры.

В работе [Ullman, 1979] получил первые результаты, показывающие, что структуру и движение жестких конфигураций из четырех точек теоретически можно вычислить по трем прямоугольным проекциям этих точек, используя неподвижную камеру. Через десять лет в работе [Huand, Lee, 1989] было показано, что эту задачу нельзя решить, располагая только двумя прямоугольными проекциями. Хотя предельные математические задачи из области определения формы по данным о движении являются интересными и могут быть трудными для решения, но они кажутся малопрактичными в связи с ошибками из-за присутствия шума и ошибок при выборе соответствующих точек. Харалик и Шапиро [Haralick, Shapiro, 1993] рассмотрели несколько математических подходов и продемонстрировали методы, повышающие надежность вычислений. Недавно в работе [Brodsky и др., 1999] представили хорошие практические результаты для вычисления плотных форм с помощью подвижной видеокамеры, наблюдающей статическую сцену.

В первой главе этой книги был задан вопрос, можно ли получить трехмерную модель Нотр-Дамского кафедрального собора путем анализа его видеосъемки. Это один из вариантов задачи структуры-по-движению, и в настоящее время для ее решения доступны промышленные устройства с использованием компьютерного зрения. Для обзорного описания используемых методов см. работу [Faugeras и др., 1998]. В данном разделе было дано введение в общую задачу вычисления структуры по движению и рассмотрено несколько ее аспектов. Советуем читателю, желающему изучить вопрос более подробно, обратиться к опубликованным работам.

## 13.12 ЛИТЕРАТУРА

Описание метода аффинной калибровки камеры следует ранним работам [Ballard, Brown, 1982], [Hall и др., 1982]. В более поздних статьях также описывается система со структурной подсветкой, в которой используются калиброванные камера и осветитель. Известны несколько жизнеспособных методов калибровки камеры. Для задач распознавания объектов часто достаточно точным оказывается аффинный метод для перспективной или даже слабой перспективной проекции. Однако для инспекционных систем и точного вычисления положения необходимы методы, учитывающие радиальную дисторсию: здесь широко применяется метод Цая [Tsai, 1987]. Калибровка требуется во



многих приложениях компьютерного зрения. Однако многие задачи могут быть решены и с некалиброванной камерой: например, это ручная съемка бытовой видеокамерой. Нам неизвестно в каждый момент времени фокусное расстояние и положение камеры в какой-либо глобальной системе координат. Но люди успешно воспринимают трехмерную структуру полученных таким образом изображений. Трехмерная структура может быть вычислена для неизвестного масштабного множителя, предполагая только, что применима модель перспективной проекции. В работе [Faugeras и др., 1998] показано, как по последовательности изображений можно формировать трехмерные модели зданий с наложением текстур. В работе [Brodsky и др., 1999] приведены результаты для вычисления структуры поверхностей более общего вида.

Наше решение задачи РЗР соответствует работе [Ohmura и др., 1988]. Похожая работа [Linainmaa и др., 1988] появилась примерно в то же время. Отметим, однако, что авторы статьи [Fischler, Bolles, 1981] изучили эту задачу и опубликовали ее аналитическое решение. Кажется, что итерационные методы имеют преимущества, когда объект прослеживается на последовательности кадров, так как в этом случае есть начальная точка, помогающая отбросить ложное решение. Хорошее альтернативное решение с использованием модели слабой перспективной проекции было дано в работе [Huttenlocher, Ullman, 1988]. В статье [Fischler, Bolles, 1981] впервые была дана формальная постановка и изучена *перспективная N-точечная задача*, а также было проведено аналитическое решение задачи РЗР. Авторы этой статьи также показали, как использовать гипотезу о N-точечном соответствии для вычисления положения объекта, а затем проверять, что другие точки модели проектируются в соответствующие точки изображения. Они назвали свой алгоритм RANSAC, так как предложили случайный выбор соответствующих точек — этого следует избегать, если известны свойства характерных точек.

В последние годы увеличился интерес к задаче построения трехмерных моделей объектов по набору изображений в лабораторной контролируемой среде. Было разработано много систем и алгоритмов. В данной главе рассматривается система для реконструирования объектов, разработанная в Вашингтонском университете [Pulli и др., 1998].

1. Ballard, D., and C. Brown. 1982. *Computer Vision*. Prentice-Hall.
2. Ballard, P., and G. Stockman. 1995. Controlling a computer via facial aspect. *IEEE-Trans-SMC*, April 1995.
3. Brodsky, T., C. Fermuller, and Y. Aloimonos. 1999. Shape from video. *Proceedings of IEEE CVPR 1999*, Ft Collins, Co. (23–25 June 1999), 146–151.
4. Chen, Y., and G. Medioni. 1992. Object modeling by registration of multiple range images. *Int. J. Image and Vision Computing*, v. 10(3):145–155.
5. Craig, J. 1986. *Introduction to Robotics Mechanics and Control*. Addison-Wesley, Reading, MA.

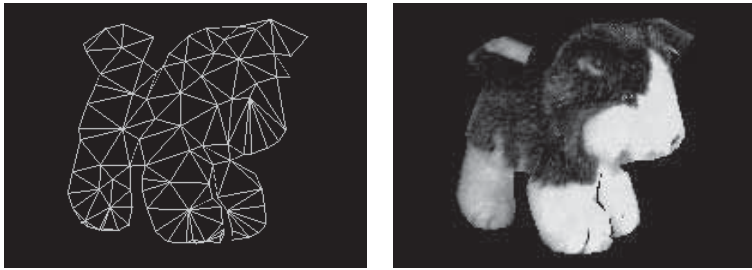
6. Curless, B., and M. Levoy. 1996. A volumetric method for building complex models from range images. *ACM Siggraph '96*, 301–312.
7. Dorai, C., J. Weng, and A. Jain. 1994. Optimal registration of multiple range views. *Proc. 12th Int. Conf. Pattern Recognition*, Jerusalem, Israel (Oct. 1994), v. 1:569–571.
8. Dubuisson, M.-P., and A. K. Jain. 1984. A modified Hausdorff distance for object matching. *Proc. 12th Int. Conf. Pattern Recognition*, Jerusalem, Israel.
9. Faugeras, O. 1993. *Three-Dimensional Computer Vision, a Geometric Viewpoint*. The MIT Press, Cambridge, MA.
10. Faugeras, O., L. Robert, S. Laveau, G. Csurka, C. Zeller, C. Gauclin, and I. Zoghiami. 1998. 3-D reconstruction of urban scenes from image sequences. *Comput. Vision and Image Understanding*, v. 69(3):292–309.
11. Fischler, M., and R. Bolles. 1981. Random consensus: a paradigm for model fitting with applications in image analysis and automated cartography. *Communications of the ACM*, v. 24:381–395.
12. Forsyth, D., and others. 1991. Invariant descriptors for 3-d object recognition and pose. *IEEE Trans. Pattern Analysis and Machine Intelligence*, v. 13(10):971–991.
13. Hall, E., J. Tio, C. McPherson, and F. Sadjadi. 1982. Measuring curved surfaces for robot vision. *Computer*, v. 15(12):385–394.
14. Haralick, R. M., and L. G. Shapiro. 1993. *Computer and Robot Vision, Volume II*. Addison-Wesley, Reading, MA.
15. Heath, M. 1997. *Scientific Computing: An Introductory Survey*. McGraw-Hill, Inc., New York.
16. Hoppe, H., T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. 1992. Surface reconstruction from unorganized points. *Proc. SIGGRAPH '92*, 71–78.
17. Horn, B. K. P., and B. L. Bachman. 1978. Using synthetic images to register real images with surface models. *CACM 21* v. 11:914–924.
18. Hu, G., and G. Stockman. 1989. 3-D surface solution using structured light and constraint propagation. *IEEE-TPAMI*, v. 11(4):390–402.
19. Huang, T. S., and C. H. Lee. 1989. Motion and structure from orthographic views. *IEEE Trans. Pattern Analysis and Machine Intelligence*, v. 11:536–540.
20. Huttenlocher, D., and S. Ullman. 1988. Recognizing solid objects by alignment. *Proc. DARPA Spring Meeting*, 1114–1122.
21. Huttenlocher, D. P., G. A. Klanderman, and W. J. Rucklidge. 1993. Comparing images using the Hausdorff distance. *IEEE Trans. Pattern Analysis and Machine Intelligence*, v. 15(9):850–863.
22. Ikeuchi, K., and B. K. P. Horn. 1981. Numerical shape from shading and occluding boundaries. *Artificial Intelligence*, v. 17(1–3):141–184.

23. Ji, Q., M. S. Costa, R. M. Haralick, and L. G. Shapiro. 1998. An integrated technique for pose estimation from different geometric features. *Proc. Vision Interface '98*, Vancouver (June 18–20), 77–84.
24. Johnson, L. W., R. D. Riess, and J. T. Arnold. 1989. *Introduction to Linear Algebra*. Addison-Wesley, Reading, MA.
25. Linnainmaa, S., D. Harwood, and L. Davis. 1988. Pose determination of a three-dimensional object using triangle pairs. *IEEE Trans. Pattern Analysis and Machine Intelligence*, v. 10(5):634–647.
26. Ohmura, K., A. Tomono, and A. Kobayashi. 1988. Method of detecting face direction using image processing for human interface. *SPIE Visual Communication and Image Processing*, v. 1001:625–632.
27. Pulli, K., H. Abi-Rached, T. Duchamp, L. G. Shapiro, and W. Stuetzle. 1998. Acquisition and visualization of colored 3D objects. *Proceedings of ICPR '98*, 11–15.
28. Ray, R., J. Birk and R. Kelley. 1983. Error analysis of surface normals determined by radiometry. *IEEE-TPAMI*, v. 5(6):631–644.
29. Shrikhande, N., and G. Stockman. 1989. Surface orientation from a projected grid. *IEEE-TPAMI*, v. 11(4):650–655.
30. Tsai, P.-S., and M. Shah. 1992. A fast linear shape from shading. *Proceedings IEEE Conf. Comput. Vision and Pattern Recognition* (June 1992), 734–736.
31. Tsai, R. 1987. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf cameras and lenses. *IEEE Trans. Robotics and Automation*, v. 3(4).
32. Ullman S. 1979. *The Interpretations of Visual Motion*. MIT Press, Cambridge, MA.
33. Vetterling, W. T. 1992. *Numerical Recipes in C*. Cambridge University Press, New York.
34. Zhang, R., P.-S. Tsai, J. Cryer, and M. Shah. 1999. Shape from shading: a survey. *IEEE-TPAMI*, v. 21(8):690–706.

# Трехмерные модели. Распознавание объектов на изображениях на основе моделей

Модели трехмерных объектов играют важную роль и в компьютерном зрении, и в машинной графике. В машинной графике трехмерные объекты необходимо представлять в виде структур данных, удобных для расчетов графических сцен и вывода на экран. Одной из наиболее известных разновидностей моделей является представление в виде трехмерного каркаса, т. е. набора многоугольников, состоящих из трехмерных точек и соединяющих их ребер. Во многих графических аппаратных устройствах предусмотрена поддержка таких моделей. Для описания гладких и/или простых поверхностей в машинной графике был разработан ряд других моделей: поверхности второго порядка (квадрики), В-сплайновые поверхности и др. Кроме информации о пространственной форме, графические модели могут содержать информацию о цвете и текстуре, накладываемой на поверхность объекта при отображении (возможно, с помощью аппаратных графических ускорителей). На рис. 14.1 показана грубая каркасная модель игрушечной собаки и изображение этой же модели в том же ракурсе с наложенными текстурами.

В области компьютерного зрения требуются способы представления объектов, удобные для решения задач распознавания объектов. Следовательно, должно быть возможно обнаружение некоторого соответствия между описанием модели объекта и характерными признаками, которые можно извлечь из изображений этого объекта. Однако в задачах распознавания трехмерных объектов применяются несколько форматов изображений, в частности: полутоновые изображения, цветные и дальнометрические. В ряде задач приходится иметь дело с полутоновыми или цветными изображениями, сопоставленными с дальнометрическими изображениями. В таком случае алгоритмы распознавания получают большое количество различных характерных признаков. Большинство видов трехмерных моделей объектов не предназначены для учета всего разнообразия характерных признаков, скорее, каждый вид моделей был разработан для решения задач конкретного типа. Поэтому перед обсуждением задачи распознавания трехмерных объектов имеет смысл сначала рассмотреть



**Рис. 14.1.** Трехмерная каркасная модель игрушечной собаки и изображение той же модели с наложением текстур. (С любезного разрешения Kari Pulli.)

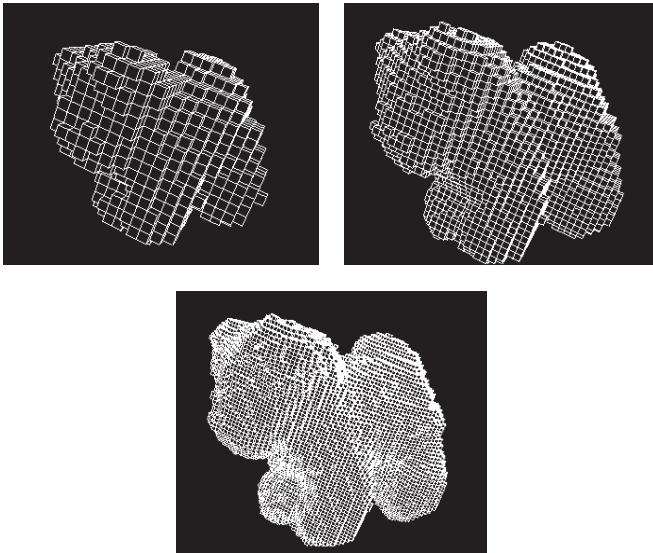
наиболее распространенные разновидности моделей объектов. Выделяются следующие наиболее общие виды моделей: геометрические модели (состоят из точек, отрезков и поверхностей), символические модели (состоят из элементарных компонент и их пространственных взаимосвязей) и функциональные модели (содержат функциональные части и их функциональные взаимосвязи). В данной главе сначала будут рассмотрены наиболее распространенные разновидности моделей трехмерных объектов, а затем модели, необходимые в типичных алгоритмах распознавания объектов.

## 14.1 ОБЗОР РАСПРОСТРАНЕННЫХ РАЗНОВИДНОСТЕЙ МОДЕЛЕЙ

Первой работой в области компьютерного зрения можно считать работу Робертса [Roberts, 1965] о распознавании многогранных объектов. Для решения этой задачи использовались простые каркасные модели и процедура сопоставления по отрезкам, найденным на изображениях. Модели на основе отрезков прямых остаются популярными и сегодня. Был разработан ряд альтернативных представлений для более точного описания объектов с криволинейными и даже произвольными поверхностями. В данном разделе будут рассмотрены каркасные модели, модели типа поверхность-ребро-вершина, воксельные модели и модели на основе октантных деревьев, модели на основе обобщенных цилиндров, модели на основе суперквадрик и деформационные модели. Будет описано различие между истинными трехмерными моделями и моделями характеристического типа, представляющими трехмерный объект с помощью множества двумерных видов.

### 14.1.1 Трехмерные каркасные модели

*Трехмерная каркасная модель* (3D mesh) — это очень простой геометрический способ описания объекта в форме множества вершин и ребер, образующих многоугольники в трехмерном пространстве. Возможны произвольные каркасные модели, содержащие любые многоугольники. *Регулярная каркасная модель* (3D regular mesh) состоит из многоугольников одного типа. Часто используются



**Рис. 14.2.** Три каркасных модели игрушечной собаки с различной точностью представления. (С любезного разрешения Kari Pulli.)

*триангулированные каркасные модели* (triangular mesh), состоящие только из треугольников (например, рис. 14.1). Каркасные модели позволяют описывать объект с различной точностью, от грубой до высокой, когда учитываются мелкие детали. На рис. 14.2 показаны три различные каркасные модели, соответствующие различной степени точности представления игрушечной собаки (модели с различным разрешением). Эти модели могут использоваться и для графической визуализации, и для распознавания объекта по дальнометрическим данным. Для решения задач распознавания операторы выделения характерных признаков должны быть реализованы таким образом, чтобы из дальнометрических данных выделялись признаки, подходящие для сопоставления с каркасной моделью. Типы таких признаков будут рассматриваться в разд. 14.4.1.

### 14.1.2 Модели типа «поверхность-ребро-вершина»

Большинство первых трехмерных зрительных систем работали с многогранными объектами с плоскими гранями, поэтому в качестве основного характерного признака для распознавания и оценки положения использовались ребра. Трехмерная модель объекта, содержащая только вершины и ребра объекта, называется *проволочной каркасной моделью* (wire-frame model). В проволочном представлении предполагается, что поверхность объекта состоит из плоских кусков и что все ребра объекта являются прямолинейными отрезками.

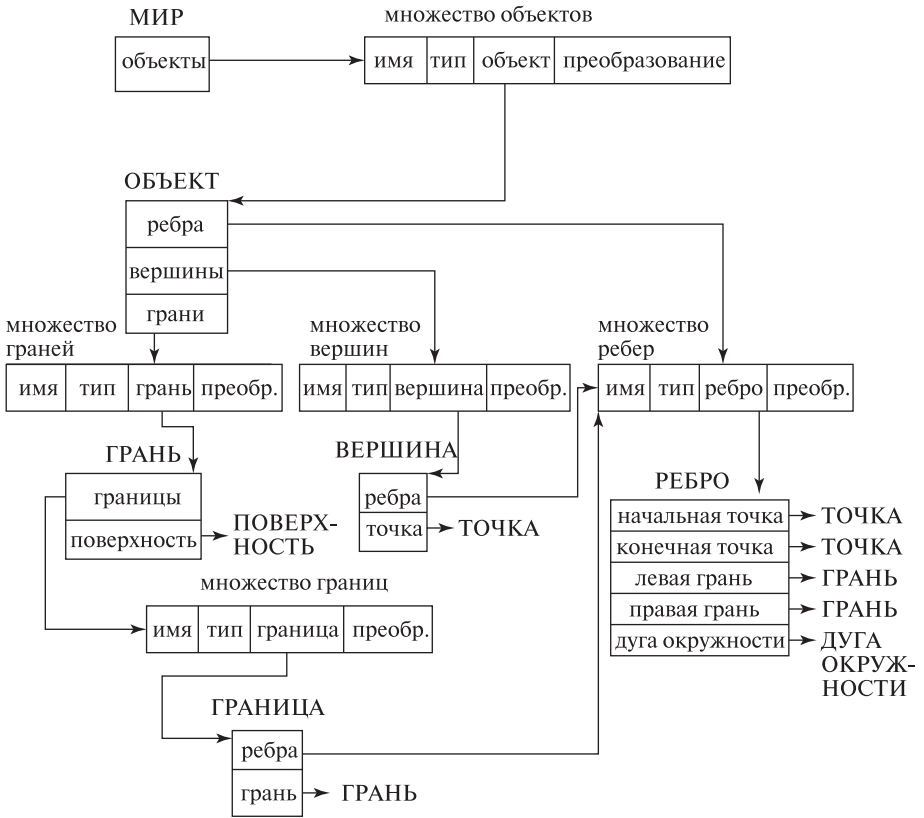
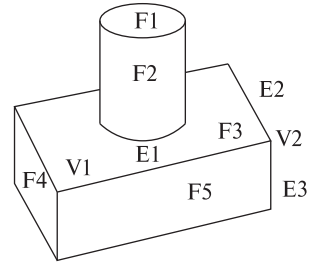


Рис. 14.3. Структура данных в модели типа «поверхность-ребро-вершина».

Полезным обобщением проволочной модели, которая широко используется в компьютерном зрении, является представление типа «поверхность-ребро-вершина» (surface-edge-vertex). Это представление предполагает использование структуры данных, в которой хранятся все вершины, поверхности и ребра объекта, а также сведения об их топологических взаимосвязях. Эти взаимосвязи указывают, какие поверхности располагаются по каждую сторону от ребра и какие вершины являются концами ребра. В случае многогранных объектов поверхности являются плоскими, а ребра — прямолинейными отрезками. Однако данная модель может быть обобщена для учета криволинейных ребер и/или криволинейных поверхностей.

На рис. 14.3 показан пример структуры данных модели «поверхность-ребро-вершина», используемой в качестве базы данных моделей объектов в системе распознавания трехмерных объектов. Это иерархическая структура данных. Наверху иерархии расположен «мир», воспринимаемый системой распознавания. На нижних уровнях расположены поверхности и дуги окружностей.



**Рис. 14.4.** Пример трехмерного объекта с плоскими и цилиндрическими поверхностями.

На рис. 14.3 прямоугольники с метками формата [имя, тип, ⟨сущность⟩, преобразование] используются для обозначения элементов, принадлежащих множеству класса ⟨сущность⟩. У каждого элемента множества есть имя, тип, указатель на ⟨сущность⟩ и трехмерное преобразование, которое применяется к ⟨сущности⟩ для получения экземпляра, нужным образом повернутого и смещенного. Например, «мир» содержит множество объектов. Это множество содержит именованные экземпляры различных моделей трехмерных объектов. Каждая модель объекта задана в собственной системе координат. Преобразование позволяет расположить каждый экземпляр объекта в мировой системе координат.

Модель каждого объекта содержит три множества: ребра, вершины и грани объекта. Вершина содержит соответствующую трехмерную точку и множество ребер, соединяющихся в этой точке. Ребро содержит начальную и конечную точки, грани слева и справа от ребра, и дугу, описывающую форму ребра, если оно не является отрезком. Грань содержит поверхность, описывающую форму грани и множество границ (внешние границы и границы внутренних отверстий). Граница содержит соответствующую грань и множество ребер. На нижнем уровне иерархии в этой структуре данных находятся дуги, поверхности и точки — все эти сущности здесь не определяются. Представление поверхностей и дуг зависит от решаемой задачи и требуемой точности и гладкости. Они могут быть представлены в виде уравнений или с помощью дальнейшего разбиения на куски поверхностей и сегменты дуг. Точки представляются просто в виде векторов координат  $(x, y, z)$ .

На рис. 14.4 показан простой трехмерный объект, который можно представить данным способом. Для упрощения примера будем рассматривать только несколько видимых поверхностей и ребер. Видимыми поверхностями являются F1, F2, F3, F4 и F5. Поверхности F1, F3, F4 и F5 являются плоскими, а F2 — цилиндрической. F1 ограничена одной границей, состоящей из единственного ребра, которое можно представить дугой окружности. У F2 есть две таких границы. У F3 есть внешняя граница из 4 прямолинейных отрезков и граница внутреннего отверстия в виде одной круговой дуги. У поверхностей F4 и F5 есть по одной границе, составленной из четырех отрезков. Ребро E1 разделяет грани F3 и F5. Если в качестве начальной точки E1 взять вершину V1, а конечной точки — вершину V2, то поверхность F3 будет левой гранью, а F5 —



правой гранью ребра E1. Вершина V2 имеет три связанных с нею ребра E1, E2 и E3.

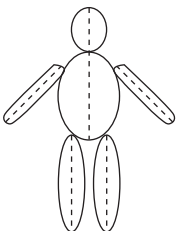
### Упражнение 14.1. Структура данных «поверхность-ребро-вершина»

С учетом представления, показанного на рис. 14.3, сформируйте полную модель объекта, показанного на рис. 14.4. Присвойте имена всем граням, ребрам и вершинам трехмерного объекта и используйте эти имена в структуре данных.

### 14.1.3 Модели на основе обобщенных цилиндров

*Обобщенный цилиндр* (generalized cylinder) — это примитивный объемный элемент, который задается пространственной криволинейной осью и функцией поперечного сечения, определенной в каждой точке этой оси. Поперечное сечение «перемещается» вдоль оси и таким образом формирует тело вращения. Например, обычный круговой цилиндр является обобщенным цилиндром, ось которого является прямолинейным отрезком, а поперечное сечение — окружностью постоянного радиуса. Конус — это обобщенный цилиндр, у которого ось — прямолинейный отрезок, а поперечное сечение — окружность с радиусом, возрастающим от 0 на одном конце оси до максимального значения на другом конце оси. Параллелепипед — это обобщенный цилиндр, ось которого — прямолинейный отрезок, а поперечное сечение — прямоугольник фиксированных размеров. Тор — это обобщенный цилиндр, ось которого — это окружность, а поперечное сечение — окружность постоянного радиуса.

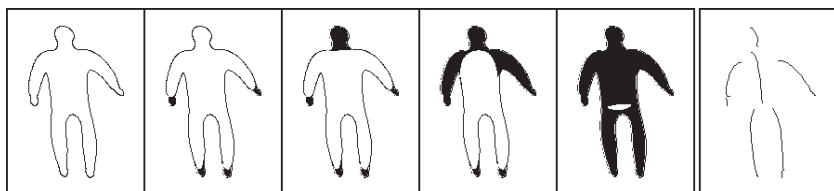
Модель объекта на основе обобщенных цилиндров содержит описание обобщенных цилиндров и их пространственных отношений, а также глобальные свойства объекта. Цилиндры могут быть описаны путем указания длины оси, среднего размера поперечного сечения, отношения двух сечений, и угла конусности. Наиболее распространенным типом пространственных отношений является связность. Кроме связности за счет совпадения конечных точек, цилиндры могут соединяться так, что конечная точка одного цилиндра соединяется с некоторой внутренней точкой второго. В данном случае для описания соединения могут быть использованы такие параметры, как координаты точки касания цилиндров, угол наклона и угол поворота одного из цилиндров относительно другого. Глобальные свойства объекта могут включать количество составных частей (цилиндров) в модели объекта, количество вытянутых



**Рис. 14.5.** Грубая модель тела человека на основе обобщенных цилиндров. Пунктирной линией показаны оси цилиндров.

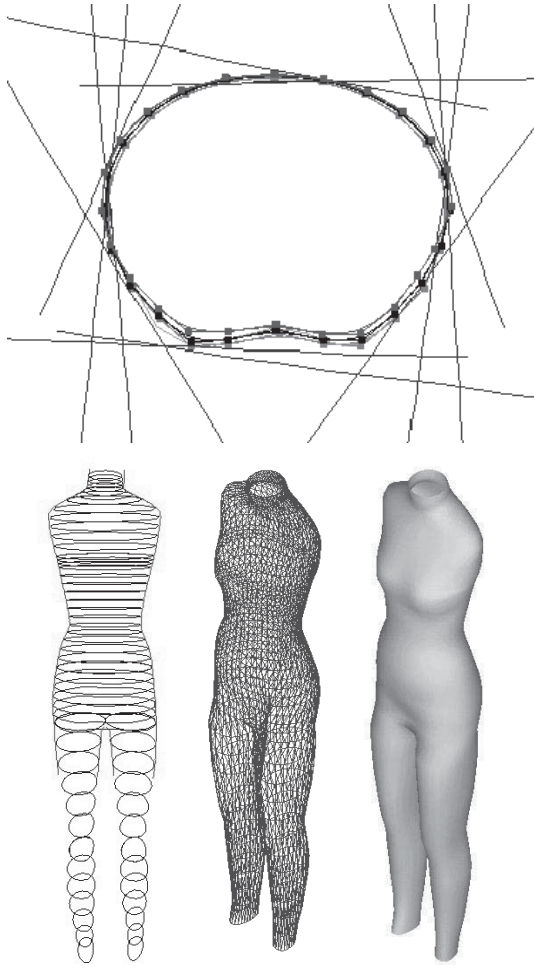
частей, и симметрию соединений. Известны иерархические модели на основе обобщенных цилиндров, в которых различные уровни детализации описываются на разных уровнях иерархии. Например, в очень грубом разрешении тело человека можно смоделировать в виде набора «стержней» (рис. 14.5), т. е. цилиндров, представляющих голову, туловище, руки и ноги. На следующем уровне иерархии туловище можно разделить на шею и нижнюю часть, руки — на три цилиндра для плеча, предплечья и кисти, и аналогичным образом ноги. На следующем уровне кисти можно разделить на ладонь и пять пальцев, а на еще более точном уровне разрешения 4 пальца руки можно разделить на 3 части, а большой палец — на две.

При проектировании трехмерного обобщенного цилиндра на двумерное изображение возникают проекции двух типов: ленты и эллипсы. *Лента* — это проекция «длинной части» цилиндра, а *эллипс* — проекция его поперечного сечения. Конечно, поперечное сечение не всегда является окружностью, так что его проекция не всегда будет эллипсом. Некоторые обобщенные цилиндры являются полностью симметричными, так что у них нет «длинных» или «коротких» частей. Но в расчете на цилиндры, у которых подобные части есть, были разработаны алгоритмы поиска лент на изображении объектов. В целом, эти алгоритмы пытаются обнаружить на изображении продолговатые области, которые по форме могут соответствовать оси цилиндра. На рис. 14.6 показан процесс определения возможных осей обобщенных цилиндров на двумерном изображении.



**Рис. 14.6.** Процесс построения модели на основе обобщенных цилиндров по двумерному изображению. (С любезного разрешения Gerard Medioni.)

На рис. 14.7 показаны этапы построения детализированной модели человеческого тела, предназначенной для системы автоматического подбора одежды. Для решения этой задачи была разработана зрительная система с 12 камерами. 6 камер получают изображения человека на равных расстояниях в цилиндрической комнате: есть набор нижних камер и набор верхних камер, так что можно получать изображения людей ростом до 2 м. Как показано на рис. 14.7, силуэты с 6 камер используются для формирования эллиптических поперечных сечений, которые затем используются для формирования цилиндрической модели. В системе используется сеточный световой шаблон, так что, кроме определения точек силуэта, можно применять триангуляцию для вычисления трехмерных координат точек поверхности. Вогнутости модели



**Рис. 14.7.** Этапы формирования модели человеческого тела для автоматического подбора одежды. (*Сверху*) три поперечных сечения вдоль оси цилиндра, и силуэты поперечных сечений по изображениям с 6 камер (прямые линии проектируют силуэт по направлению к камере). Признаки, получаемые с помощью структурной подсветки, позволяют вычислить трехмерные точки в вогнутых областях модели. (*Внизу*) модель на основе обобщенных цилиндров созданная по эллиптическим поперечным сечениям для 6 силуэтов, результирующая триангулированная каркасная модель и тонированное изображение этой модели. (С любезного разрешения Helen Shen и др. из Dept. of Computer Science, Hong Kong University of Science and Technology: проект выполнен при поддержке гранта AF/183/97, полученного от Industry and Technology Development Council of Hong Kong, SAR of China, 1997.)

формируются с использованием данных структурной подсветки, и в результате система вычисляет триангулированную каркасную модель с высоким разрешением.

---

### Упражнение 14.2. Модели на основе обобщенных цилиндров

Постройте на основе обобщенных цилиндров модель самолета. У модели должны быть фюзеляж, крылья и хвост. На каждом крыле должен располагаться мотор. Попробуйте описать отношения связности между парами обобщенных цилиндров.

#### 14.1.4 Модели на основе октантных деревьев

*Октантное дерево* (octree) — это иерархическая восьмиарная древовидная структура данных. Каждый узел дерева соответствует кубическому объему трехмерного пространства. У каждого узла есть *метка*, которая может принимать одно из трех значений: *заполнено*, если куб полностью лежит внутри трехмерного объекта, или *пусто*, если куб лежит вне объекта, или *частично заполнено*, если куб частично пересекает объект. У узлов с *полными* и *пустыми* метками нет дочерних узлов. Узлы с *частичными* метками имеют по 8 дочерних узлов, представляющих разбиение данного куба на 8 кубов меньшего объема.

Трехмерный объект может быть представлен в виде трехмерного массива размерами  $2^n \times 2^n \times 2^n$ , где  $n$  — некоторое целое число. Элементы массива называются *вокселями* (voxels) и могут принимать значения 1 (полный) или 0 (пустой), соответствующие наличию или отсутствию объекта в данном вокселе. Описание объекта на основе октантного дерева эквивалентно представлению в виде трехмерного массива, но обычно требует меньше памяти. На рис. 14.8 представлен простой пример объекта и его представления в виде октантного дерева (используется схема нумерации октантов, предложенная в работе [Jackins, Tanimoto, 1980]).

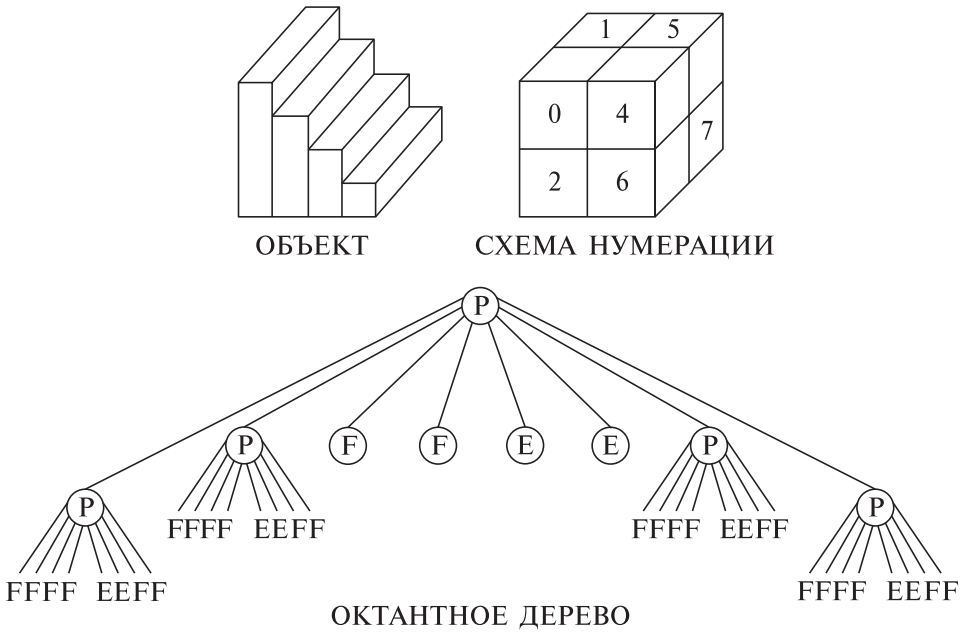
---

### Упражнение 14.3. Октантные деревья

На рис. 14.11 показаны два вида простого объекта — стула. Постройте модель стула на основе октантного дерева. Предположите, что сиденье и спинка занимают объем  $4 \times 4 \times 1$  вокселей, а каждая ножка —  $3 \times 1 \times 1$  воксел.

#### 14.1.5 Модели на основе суперквадрик

Модели на основе суперквадрик первоначально были разработаны для задач машинной графики. Использовать их в области компьютерного зрения предложил Пентланд (Pentland). Качественно суперквадрики можно представлять себе как куски глины, которые для формирования модели объекта можно



**Рис. 14.8.** Простой трехмерный объект и его описание на основе октантного дерева.

деформировать и склеивать вместе. Математически, суперквадрики образуют семейство параметрических поверхностей. Суперквадрика может быть представлена в виде вектора  $\mathbf{S}$ ,  $x$ -,  $y$ - и  $z$ -компоненты которого задаются как функции углов  $\eta$  и  $\omega$ :

$$\mathbf{S}(\eta, \omega) = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_1 \cos^{\epsilon_1}(\eta) \cos^{\epsilon_2}(\omega) \\ a_2 \cos^{\epsilon_1}(\eta) \sin^{\epsilon_2}(\omega) \\ a_3 \sin^{\epsilon_1}(\eta) \end{bmatrix} \quad (14.1)$$

для  $-\frac{\pi}{2} \leq \eta \leq \frac{\pi}{2}$  и  $-\pi \leq \omega < \pi$ . Параметры  $a_1$ ,  $a_2$  и  $a_3$  задают размеры суперквадрики по направлениям  $x$ ,  $y$  и  $z$ . Параметры  $\epsilon_1$  и  $\epsilon_2$  представляют «прямоугольность» в плоскостях широты и долготы.

С помощью суперквадрик можно описывать модели таких полезных «строительных блоков», как сферы, эллипсоиды, цилиндры, параллелепипеды и фигуры «промежуточных» между ними форм. При  $\epsilon_1 = \epsilon_2 = 1$  суперквадрика является эллипсоидом, а если при этом  $a_1 = a_2 = a_3$ , то сферой. При  $\epsilon_1 \ll 1$  и  $\epsilon_2 = 1$  поверхность выглядит похожей на цилиндр.

Значимость представления на основе суперквадрик не в том, что оно позволяет моделировать идеальные геометрические фигуры, а в том, что оно позволяет моделировать деформируемые геометрические фигуры путем деформаций, таких, как *сужение* и *изгиб*. Линейное сужение вдоль оси  $z$  может

быть представлено в виде преобразования:

$$\begin{aligned}x' &= \left( \frac{k_x}{a_3} z + 1 \right) x \\y' &= \left( \frac{k_y}{a_3} z + 1 \right) y \\z' &= z\end{aligned}$$

где  $k_x$  и  $k_y$  ( $-1 \leq k_x$  и  $k_y \leq 1$ ) являются параметрами сужения в плоскостях  $x$  и  $y$  по отношению к оси  $z$ . Деформация изгиба определяется преобразованием

$$\begin{aligned}x' &= x + \cos(\alpha)(\mathbf{R} - r) \\y' &= y + \sin(\alpha)(\mathbf{R} - r) \\z' &= \sin(\gamma)\left(\frac{1}{k} - r\right)\end{aligned}$$

где  $k$  — кривизна, а  $r$  — проекция  $x$ - и  $y$ -компонент на плоскость изгиба  $z - r$ :

$$r = \cos\left(\alpha - \text{tg}^{-1}\left(\frac{y}{x}\right)\right)\sqrt{x^2 + y^2}$$

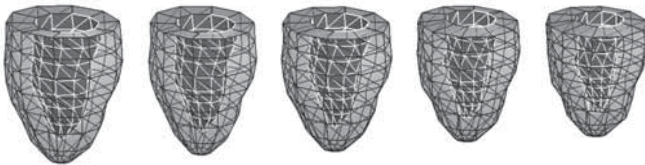
$\mathbf{R}$  является преобразованием  $r$ , которое описывается выражением

$$\mathbf{R} = k^{-1} - \cos(\gamma)(k^{-1} - r)$$

а  $\gamma$  — это угол изгиба

$$\gamma = zk^{-1}$$

Модели на основе суперквадрик в основном применяются для обработки дальнометрических данных. Для этих целей был разработан ряд методов подбора параметров суперквадрики для построения сглаженных поверхностей. На рис. 14.9 показан подбор моделей на основе суперквадрик по трехмерным дальнометрическим данным о положении левого желудочка сердца в пять различных моментов времени. В данном случае использовались расширенные суперквадрики с параметрическими функциями (т. е. параметрами суперквадрики являются функции, а не константы).



**Рис. 14.9.** Сглаженные модели левого желудочка, построенные для пяти различных моментов времени в течение одного сокращения. Модели построены на основе расширенных суперквадрик с использованием параметрических функций. (С любезного разрешения Jinah Park, Dmitris Metaxas.)

## 14.2 МОДЕЛИ НА ОСНОВЕ КЛАССОВ ВИДИМОСТИ КАК АЛЬТЕРНАТИВА ИСТИННЫМ ТРЕХМЕРНЫМ МОДЕЛЯМ

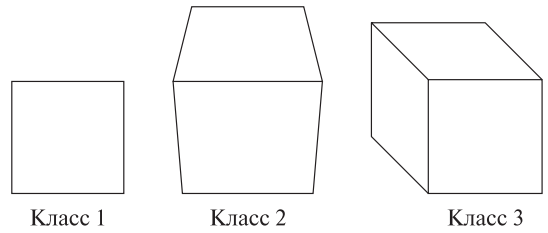
Во всех рассмотренных выше способах описания трехмерных объектов основное внимание уделялось именно пространственной форме объектов, и при этом игнорировалась задача распознавания этих объектов на двумерных изображениях, полученных с произвольной точки наблюдения. Большинство объектов выглядят по-разному при рассмотрении с различных точек наблюдения. Цилиндр с одного множества точек зрения проектируется в ленту (см. выше), а с другого множества точек — в эллипс. В общем, мы можем разбить пространство точек наблюдения на конечное множество *классов видимости* (view classes)<sup>1</sup>, каждый из которых представляет множество точек зрения с одинаковым свойством проекции объекта. Например, таким свойством может быть то, что на изображении будут видны одни и те же поверхности объекта, или одни и те же прямолинейные отрезки, или относительное расстояние (см. гл. 11) между структурами, выделенными среди контуров на изображении, оказывается малым для каждой точки наблюдения из одного класса видимости.

На рис. 14.10 показаны классы видимости для куба, определенные следующим образом: вместе сгруппированы точки наблюдения, для которых получаются топологически изоморфные совокупности прямолинейных отрезков. На рис. 14.11 показаны два вида стула, на которых большинство (но не все) поверхности являются одинаковыми. Эти виды могут быть сгруппированы вместе на основе алгоритма кластеризации, использующего относительное расстояние, заданное на примитивах-областях и на отношении смежности областей как основе для оценки близости видов. Может существовать множество различных похожих видов, которые вместе образуют класс видимости. В принципе, количество видов может быть бесконечным. Главное, что если для объекта был определен корректный класс видимости, то согласование модели и изображения для определения соответствия в процедуре оценки местоположения/ориентации сводится к двумерному согласованию с сильными ограничениями.

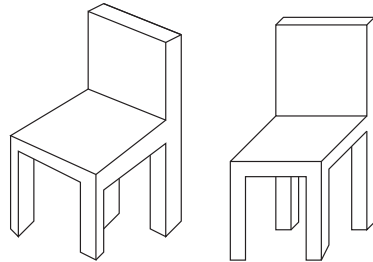
Классы видимости были предложены в работе [Koenderink, van Doorn, 1979] в виде структуры, названной *графом ракурса* (aspect graph). *Ракурс* определяется как качественно отличный от других ракурсов вид объекта, наблюдаемый с некоторого множества связанных точек наблюдения. Узлы графа ракурса представляют ракурсы, а ребра соединяют смежные ракурсы. Изменение вида объекта на границе двух ракурсов называется *визуальным событием*. Алгоритмы для автоматического построения графов ракурсов были разработаны в конце 1980-х гг., но так как для реальных объектов получаемые структуры оказываются слишком большими, то они не получили широкого распространения в системах распознавания объектов. Вместо графов ракурса

<sup>1</sup> Также называются *характеристическими видами*.

**Рис. 14.10.** Три класса видимости куба, заданные путем группировки точек наблюдения, для которых получают топологически изоморфные структуры из прямолинейных отрезков.



**Рис. 14.11.** Два вида стула, принадлежащие одному классу видимости за счет небольшого относительного расстояния между видами.



широкое распространение получили классы видимости (характеристические виды).

#### Упражнение 14.4. Модели на основе классов видимости

На рис. 14.11 показаны два стула, принадлежащие одному классу видимости трехмерного объекта-стула. Нарисуйте изображения для трех наиболее общих классов видимости стула.

### 14.3 ФИЗИЧЕСКИЕ И ДЕФОРМАЦИОННЫЕ МОДЕЛИ

Физические модели можно использовать для моделирования внешнего вида и поведения реального физического объекта, наблюдаемого на изображениях. Ниже приведены примеры моделей человеческого сердца (рис. 14.16) и телефонной трубки (рис. 14.15). Часто физические законы используются не для моделирования реальной физической системы, а в качестве аналогии для построения алгоритмов анализа изображений, поведение которых некоторым образом напоминает физические модели. В задаче моделирования сердца цель состоит в моделировании формы и поведения объекта на протяжении некоторого времени, чтобы по результатам моделирования можно было понять особенности функционирования этого объекта. В моделировании телефонной трубки целью является получение «хорошей» каркасной модели для статических измерений.

С понятием *физической модели* тесно связано понятие *деформационной модели*. Последний термин подчеркивает, что особое внимание уделяется моделированию изменений формы объекта.



В последнее время в области физических и деформационных моделей был достигнут значительный прогресс. По объему и сложности теория и приложения в этой области компьютерного зрения превосходят другие темы, рассматриваемые в данной книге. Ниже приводится краткий обзор, предназначенный только для первоначального ознакомления с темой. Заинтересовавшийся читатель может обратиться к дополнительной литературе по этому быстро развивающемуся разделу компьютерного зрения.

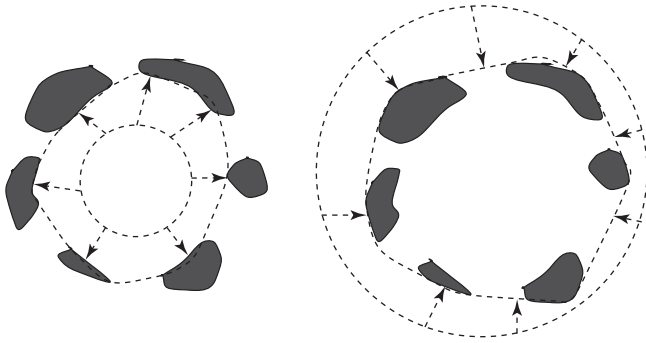
### 14.3.1 Модели на основе активных контуров (snakes-модели)

Представим себе резинку, натянутую на растопыренные пальцы. В данном случае пальцы аналогичны пяти двумерным точкам, а резинка — замкнутому контуру, проходящему через эти пять точек.

По аналогии с поведением резиновой ленты в компьютерном зрении разработаны алгоритмы с использованием *активных контуров*. Активный контур перемещается на изображении, стремясь занять положение с наименьшей энергией. Резиновая лента стягивается для уменьшения запасенной в ней энергии, до тех пор, пока силы упругости, стягивающие резинку, не будут уравновешены некоторыми внешними силами (например, реакцией пальцев). Это поведение поясняется на рис. 14.12 (справа). Небольшие черные области аналогичны нашим пальцам: резинка прекращает стягиваться по достижении этих областей. Еще одно свойство аналогии заключается в том, что резиновая лента не может иметь бесконечно большую кривизну; даже если она задерживается единственной точкой (или проводом), то лента будет гладко охватывать эту точку. В процессе имитации резиновой ленты надо предусмотреть защиту от формирования контуров с большой кривизной. На рис. 14.12 (слева) показано, что происходит при надувании оболочки внутри руки, как будто пальцы руки охватывают надуваемый мяч. Аналогично, можно смоделировать надувание воображаемой оболочки внутри некоторых областей изображения или множества точек.

Рис. 14.12 демонстрирует важнейшее преимущество активных контуров: контур является самодостаточной структурой, даже если данные, по которым он построен, оказываются сильно фрагментированными. Более того, многие характеристики можно ввести в качестве параметров алгоритма, например, гладкость, пределы периметра, или ограничение на форму контура (например, что контур должен иметь форму некоторой простой кривой). Далее кратко будет рассмотрено, как можно смоделировать поведение активного контура с помощью компьютерного алгоритма.

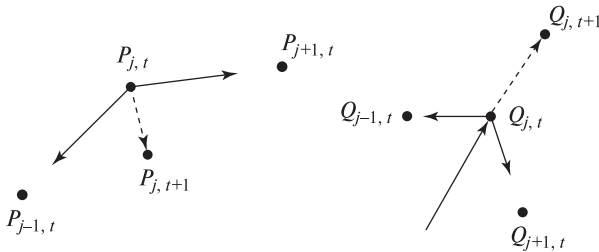
Для имитации поведения активного контура, во-первых, надо выбрать структуру данных для хранения информации о форме и местоположении контура. Рассмотрим простейший случай, когда в момент времени  $t$  имеется  $N$  точек контура. Обозначим их местоположение  $P_{j,t}$ . Каждая точка связана с двумя соседями  $P_{j-1,t}$  и  $P_{j+1,t}$ . В случае воображаемой резиновой ленты, на каждую точку будет действовать сила реакции со стороны двух соседей. В



**Рис. 14.12.** (Слева) двумерная «оболочка» или *активный контур*, расширенный для согласования с множеством двумерных точек. (Справа) двумерная «резиновая лента», натянутая снаружи двумерных точек.

результате точка  $P_{j,t}$  будет ускоренно двигаться в новое положение  $P_{j,t+1}$  (см. рис. 14.13). Обычно считается, что каждая точка имеет единичную массу. Зная ускорение, вычисляем скорость, а затем, по скорости — координаты точки. Следовательно, для момента времени  $t$  в структуре данных для активного контура надо хранить ускорение и скорость каждой точки. В момент, соответствующий началу моделирования, они не обязательно должны быть равны 0. Для каждой точки требуется еще одна логическая переменная: флаг, показывающий, продолжает данная точка двигаться или была остановлена (наткнулась на некоторую точку данных) — т.н. *жесткое ограничение*. Конечно, кроме объекта-активного контура, необходима память для хранения исходных данных: это может быть полутоновое изображение, множество двумерных контурных точек, множество трехмерных точек поверхности и т. п. (исходные данные должны быть представлены одним из способов, описанных в данной главе, в гл. 2 или 10).

Ниже приведен простой алгоритм расчета перемещения точек активного контура. Точки контура двигаются, пока не встретится жесткое ограничение



**Рис. 14.13.** (Слева) силы, действующие на точку растянутой резиновой ленты, стремятся сдвинуть эту точку внутрь. (Справа) сила, растягивающая резиновую оболочку, сдвигает точку наружу, если превышает упругие силы, действующие на точку со стороны соседей.

По текущим координатам точек  $P_{j,t}$  активного контура алгоритм вычисляет новые координаты этих точек  $P_{j,t+1}$ .

**Входные данные:**  $N$  точек контура в момент времени  $t$ ; для каждой точки  $P_{j,t}$  известна скорость  $V_{j,t}$  и ускорение  $A_{j,t}$ .

**Выходные данные:**  $N$  точек контура в момент времени  $t + 1$ ; для каждой точки  $P_{j,t+1}$  вычислены скорость  $V_{j,t+1}$  и ускорение  $A_{j,t+1}$ .

Задан временной шаг  $\Delta t$ . Для каждой точки  $P_{j,t}$ , если она не встретила жесткое ограничение, выполняются следующие действия:

1. Вычислить результирующую силу, действующую на  $P_{j,t}$  со стороны соседних точек контура.
2. Зная силу, вычислить вектор ускорения  $A_{j,t+1}$ .
3. Вычислить скорость  $V_{j,t+1} = V_{j,t} + A_{j,t}\Delta t$ .
4. Вычислить новые координаты точки  $P_{j,t+1} = P_{j,t} + V_{j,t}\Delta t$ .
5. Если точка  $P_{j,t+1}$  лежит в некоторой окрестности одной из точек данных (предел точности задается как параметр алгоритма), то на эту точку накладывается жесткое ограничение и она больше не будет двигаться.

**Алгоритм 14.1.** Один шаг циклического алгоритма перемещения активного контура

или пока действующая на них сила не станет равной 0. Возможно, такой алгоритм никогда не прекратит работу (например, когда активный контур применяется к изображению губ говорящего человека). Отметим, что необходимо задать исходное местоположение контура.

В алгоритме 14.1, в сущности, приведен шаг алгоритма Эйлера, в котором для малого временного промежутка вычисляется сила, затем по силе — ускорение, затем — скорость по ускорению и координаты точки по скорости. Координаты точки «замораживаются» при столкновении с какой-либо точкой данных, ребром или куском поверхности модели. В общем, это может быть достаточно сложная вычислительная процедура, в которой требуется выполнять поиск такой точки данных в некоторой структуре данных или на изображении.

Пружина — часто встречающийся элемент физических моделей. Ее поведение описывается законом Гука. Предположим, что точки  $P_j$  и  $P_k$  соединены пружиной длиной  $L$ . Сила  $F$ , действующая на точку  $P_j$ , пропорциональна растяжению (или сжатию) пружины.

$$F = -k_L(\|P_j - P_k\| - L) \frac{P_j - P_k}{\|P_j - P_k\|} \quad (14.2)$$

Изложенного должно быть достаточно для моделирования нашей резиновой ленты. Чтобы пружина не колебалась бесконечно долго, можно добавить силу

трения. Надо также решить, какой выбрать начальную длину пружины  $L$  — длину в нерастянутом состоянии. Если выполняется моделирование известного объекта, например, губ говорящего человека, то можно экспериментально определить значения  $N$ ,  $L$  и  $k_L$ .  $k_L$  — это жесткость пружины, от нее зависит величина силы, деформирующей резиновую ленту.

**Понятие активного контура и критерий минимума энергии\*** Хотя понятие активного контура было введено достаточно давно, интерес исследователей в области компьютерного зрения к подобным алгоритмам возрос после публикации работы [Kass, Witkin, Terzopoulos, 1987]. Большая часть вышеприведенного изложения основана на рассмотрении snakes-моделей в их работе. Подгонка snake-модели к данным была сведена к задаче оптимизации, в которой требуется определить граничный объект с минимальной энергией при наличии жестких ограничений. Полную энергию контура можно рассматривать как сумму трех компонент: (1) *внутренняя энергия контура* ( $E_{internal}$ ), зависящая от растяжения и изгиба контура, (2) *энергия изображения* ( $E_{image}$ ), характеризующая, насколько хорошо контур соответствует интенсивности и градиенту на изображении, и (3) *внешняя энергия* ( $E_{constraints}$ ), зависящая от ограничивающих сил. Ограничения вводятся для учета информации от пользователя или от высокоуровневых программных компонентов.

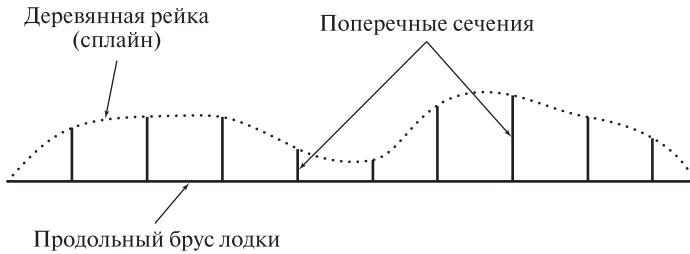
Представим контур в параметрическом виде  $\mathbf{v}(s) = [x(s), y(s)]$ , где параметр  $s$  — вещественная переменная,  $s \in [0, 1]$ . Задача заключается в нахождении функции, которая минимизирует функцию следующего вида:

$$E_{contour} = \int_0^1 (E_{internal} + E_{image} + E_{constraints}) ds \quad (14.3)$$

$$E_{internal} = \alpha(s)|\mathbf{v}'(s)|^2 + \beta(s)|\mathbf{v}''(s)|^2 \quad (14.4)$$

Контуром можно управлять, чтобы он проходил около некоторых заданных точек — для этого надо, чтобы энергия  $E_{constraints}$  зависела от квадрата расстояния между каждой такой точкой и точкой контура. Слагаемое  $E_{image}$  может быть просто суммой квадратов расстояний между точками контура и ближайшими к ним точками данных (например, перепадами яркости). Определение внутренней энергии контура несколько интереснее. Первое слагаемое  $E_{internal}$  учитывает разницу длин малых сегментов контура — чем меньше они отличаются, тем меньше энергия. Второе слагаемое  $E_{internal}$  учитывает кривизну контура. Весовые функции  $\alpha(s)$  и  $\beta(s)$  используются для суммирования двух слагаемых внутренней энергии и позволяют управлять формированием углов контура в тех точках изображения, где их выделил детектор углов или, наоборот, распространять контур вдоль длинных объектов на однородных текстурах.

Подгонка активных контуров к данным изображений напоминает сплайновую интерполяцию, применяемую в САПР. Сплайны — это геометрические кривые, которые ведут себя подобно упругим рейкам из дерева или стали.



**Рис. 14.14.** Деревянная рейка, прикрепленная к поперечным сечениям, образует контур с «минимальной энергией».

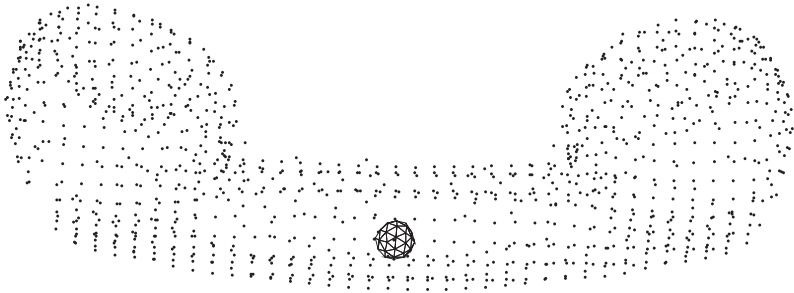
На рис. 14.14 показано, что происходит, когда деревянная рейка закрепляется через равномерные промежутки гвоздями так, чтобы поперечные сечения формировали обвод судна (например, лодки каное). В воздушных промежутках между точками крепления рейка стремится занять такое положение, что ее форма может быть описана гладкой кривой. Но форма этой кривой имеет довольно сложный вид. Точки крепления образуют жесткие ограничения. Высокая кривизна уменьшается, так как дерево распределяет энергию изгиба по некоторому множеству близлежащих точек. Существуют компьютерные алгоритмы для построения подобных кривых — *сплайнов*. Рис. 14.14 был построен с помощью одного из таких алгоритмов, реализованного в графическом редакторе *xfig*.

Подход, основанный на минимизации энергии контура, выходит за рамки материала данной книги. При практической реализации необходимо аккуратное программирование численных методов, чтобы обеспечить хорошие возможности управления активным контуром. Могут быть использованы готовые пакеты конечноэлементного анализа. После 1987 г. появился ряд новых работ, в которых применительно к данной задаче вместо предложенного в [Kass и др., 1987] метода масштабирования использовался метод динамического программирования. Читатель может найти ряд интересных работ по этой теме в списке литературы.

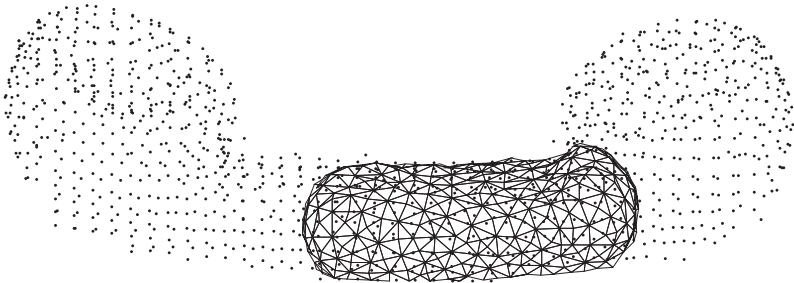
### 14.3.2 Трехмерные оболочечные модели

В качестве примера оболочечной модели можно рассмотреть каркасную модель сферы. Большинство футбольных мячей сделаны из 12 пятиугольных и 20 шестиугольных частей, каждую из которых можно разбить на треугольники. Стороны треугольников можно представить как пружины, так что форма всей модели может деформироваться в результате расширения или сжатия. На рис. 14.15 показана подобная сферическая модель, расширяющаяся внутри облака трехмерных точек данных, полученных по результатам дальнометрических измерений поверхности телефонной трубки. Алгоритм прекращает движение вершины модели после того, как она столкнется с одной из точек

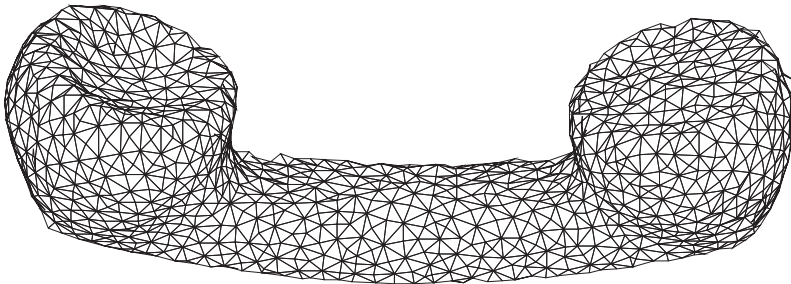
данных. Сила растяжения считается приложенной к каждой вершине модели по направлению нормали к поверхности модели. Направление нормали интерполируется по нескольким вершинам модели. Для обнаружения столкновения с точками данных выполняется поиск этих точек только в направлении нормали к поверхности модели. Когда треугольник оболочки становится слишком большим, алгоритм разбивает его на 4 треугольника меньших размеров. Такой



(a) В начальном положении сферическая оболочка лежит целиком внутри облака трехмерных точек



(b) На некотором шаге оболочка расширяется настолько, что некоторые ее точки сталкиваются с трехмерными точками данных



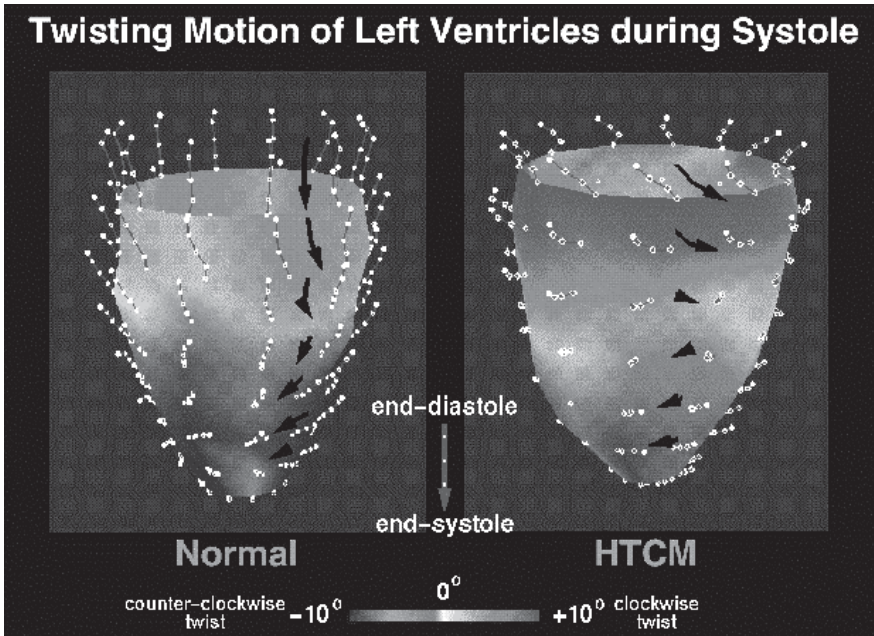
(c) Вид триангулированной каркасной модели после завершения расширения оболочки

**Рис. 14.15.** Начальное, промежуточное и окончательное состояние оболочки в алгоритме построения каркасной триангулированной модели по множеству дальнометрических точек. В алгоритме используется физическая оболочечная модель. (С любезного разрешения Yang Chen, Gerard Medioni.)

подход позволяет сферической оболочке превратиться в оболочку продолговатой формы (рис. 14.15 (b), (c)). Трехмерные точки данных были получены путем совмещения нескольких дальнометрических видов объекта, полученных с помощью трехмерного сканера. Процедура «сшивки» нескольких различных и неполных каркасных моделей поверхности представляется гораздо более сложной, чем описанная процедура с использованием раздувающейся оболочки. Оболочечная модель сохраняет изначально заложенные в нее топологические свойства и несколько сглаживает неоднородности, когда треугольники сталкиваются с точками данных. По сравнению с оболочечной моделью, построить хорошую модель поверхности непосредственно по множеству трехмерных точек оказывается значительно сложнее.

### 14.3.3 Моделирование движения человеческого сердца

По аналогии с использованием треугольников в каркасных моделях поверхностей, для объемных трехмерных моделей можно использовать треугольные пирамиды. Каждая треугольная пирамида имеет 4 вершины, 4 грани и 6 ребер. Ребрам можно присвоить некоторые значения *жесткости*, зависящие



**Рис. 14.16.** Обработка данных ЯМР-томографа о движении двух бьющихся сердец. Датчик может воспринимать отдельные фрагменты живых тканей и их движение в трехмерном пространстве. На рис. показаны векторы перемещений, различные для двух сердец. (С любезного разрешения Jinah Park, Dimitris Metaxas.)



от моделируемого материала. Когда к различным точкам модели будут приложены силы, модель будет деформироваться. На рис. 14.16 показаны два состояния бьющегося человеческого сердца. Эти модели построены по данным, полученным на ЯМР-томографе. Датчик может отмечать отдельные части живых тканей, так что их можно наблюдать в движении и в трехмерном виде. Модель сердца построена по данным измерений для того, чтобы можно было моделировать реальные физические свойства сердца. Движение точек модели может облегчить понимание функционирования сердца. Деформации пирамидальных элементов модели зависят от действующих на них сил и от жесткости моделируемых тканей.

#### 14.4 ОСНОВНЫЕ МЕТОДЫ РАСПОЗНАВАНИЯ ТРЕХМЕРНЫХ ОБЪЕКТОВ

После обзора разновидностей моделей трехмерных объектов перейдем к рассмотрению наиболее общих подходов к распознаванию трехмерных объектов на изображениях. Выделить общие подходы довольно сложно, так как метод распознавания сильно зависит от конкретной прикладной задачи, типа данных и требований к результатам распознавания. Существует много критериев классификации или ограничения задачи распознавания объектов. Перечислим часть из них:

- **К какой области относится задача — к одной из инженерных областей или к психологии?** При разработке систем для решения прикладных технических задач часто оказывается, что у задачи есть ряд специфических особенностей, которые позволяют упростить разработку системы. Примером подобной задачи является разработка зрительной системы для робота, который должен захватить стальной цилиндр из груды однотипных цилиндров. С другой стороны, задача может относиться к области психологии и требовать понимания процессов распознавания объектов человеком. Эти задачи очень сложны, и для их решения необходима разработка общей теории, которая будет объяснять многочисленные психологические факты.
- **Какие объекты встречаются в наблюдаемых сценах — искусственные или естественные?** У искусственных объектов характерные признаки обычно лучше заметны и проще обнаруживаются, чем у естественных. Для многих объектов, изготовленных человеком, можно построить точную компьютерную твердотельную модель. Это сразу открывает возможность применения ряда известных алгоритмов распознавания. Естественные объекты формируются природными процессами (геологическими, биологическими и др.), и в результате даже среди однотипных объектов наблюдается большое разнообразие свойств, что сильно усложняет построение моделей. Кроме того, сцены с естественными объектами обычно имеют меньше ограничений и их содержание сложнее предсказать, чем сцены с искусственными объектами. Например, задача распознавания для управ-



ления мобильным роботом в естественной среде является гораздо более сложной, чем задача распознавания и определения расположения объектов для автоматизации производственной линии.

- **Являются ли поверхности объектов многогранными, квадраками или произвольными?** Во многих известных проектах по распознаванию объектов рассмотрение ограничивалось объектами в форме многогранников. В таком случае моделирование объектов особенно упрощается. В последнее время возрос интерес к объектам с поверхностями-квадриками. Распространено мнение, что к ним относится до 85% промышленно изготовленных объектов. Главное удобство такого подхода в том, что модельные и измерительные данные могут быть описаны одинаковыми примитивами (возможно, с учетом подбора значений параметров). Пока не ясно, какие модели наилучшим образом подходят для представления пространственных поверхностей произвольной формы, даже если это поверхности твердых тел. Произвольные поверхности, например, кузов спортивного автомобиля, лопатка турбины или поверхность айсберга, могут состоять из большого количества гладко сопряженных поверхностных фрагментов, которые оказывается сложно сегментировать на какие-либо элементарные примитивы.
- **В наблюдаемой сцене может присутствовать только один объект или несколько?** В некоторых системах распознавания предполагается, что распознаваемые объекты в сцене всегда отделены от других объектов. Изоляция наблюдаемых объектов возможна в ряде технических задач, но не всегда. Сцены с несколькими объектами обычно сложнее для анализа, так как характерные признаки различных объектов могут перекрываться или смешиваться. Методы, основанные на глобальном выделении характерных признаков, хорошо работают только для сцен с одним объектом. Задача сегментации приобретает особую важность именно для сцен с несколькими объектами.
- **Для чего будут использоваться результаты распознавания?** Может потребоваться распознавать объект для целей инспекционного контроля, для захвата манипуляционным роботом, для объезда препятствия подвижным роботом и др. В задаче контроля можно было бы сосредоточиться на мелких деталях некоторой части объекта — в этой задаче точность моделирования и измерений должна быть высокой. Захват объекта предъявляет другие требования. В этой задаче требуется знать не только геометрические характеристики объекта, но также необходимо учитывать вес, прочность и доступность объекта в рабочем пространстве робота. Для мобильного робота, который распознает объекты вблизи траектории движения и должен избегать препятствий, достаточно получения грубых данных о размерах, форме и местоположении объекта.
- **Исходные данные являются двумерными или трехмерными?** Человек достаточно хорошо воспринимает трехмерные сцены даже одним глазом. Разработано большое количество зрительных систем, в которых

в качестве входных данных используются только двумерные полутонные изображения. Двумерные характерные признаки, найденные на изображении объекта, связаны с трехмерной моделью посредством видового преобразования. Обычно процесс распознавания включает вычисление этого преобразования и проверку правильности идентификации объекта. При наличии трехмерных данных распознавание часто упрощается. Поэтому во многих современных системах используются дальнометрические измерения. При этом считается, что эти измерения непосредственно позволяют оценить форму поверхности объектов и их местоположение. Это, в свою очередь, упрощает выбор моделей возможных объектов и уменьшает степень неоднозначности при вычислении преобразования, совмещающего различные дальнометрические виды.

- **Какие модели объектов используются — геометрические или символические?** Геометрические модели точно описывают пространственную форму объекта, а символические модели — целые классы объектов. Геометрические модели особенно широко используются в промышленных задачах компьютерного зрения, в которых распознаваемые объекты принадлежат небольшому заранее известному множеству. Необходимые геометрические данные часто можно получить из систем САПР. Символические модели нужны в задачах, в которых есть несколько классов объектов и в каждом классе объекты могут иметь какие-то различия. Например, при обработке медицинских изображений, на которых каждый орган связан с отдельным классом объектов, а изображение органа каждого человека является уникальным объектом данного класса. У многих объектов окружающего мира, например, кресел в комнате, есть много индивидуальных различий. Для учета этих различий необходим более гибкий, по сравнению с геометрическим, подход к моделированию.
- **Предусмотрено ли изменение моделей посредством обучения или они жестко запрограммированы?** Модели объектов могут содержать большое количество точных данных, которые сложно подготовить вручную. Данных, импортированных из систем САПР, может оказаться недостаточно. В них может быть необходимо внести некоторые дополнения или изменить их структуру, например, как-то обозначить характерные признаки. В связи с этим представляется привлекательной возможность введения в систему распознавания режима обучения, в котором система будет запоминать геометрические свойства объекта, предъявляемого в удобных для распознавания условиях.

#### **14.4.1 Распознавание по трехмерным моделям с использованием процедуры совмещения**

При распознавании трехмерных объектов посредством совмещения данных модели и данных измерений применяются те же приемы, что и при двумерном совмещении (см. гл. 11). Основные этапы распознавания трехмерного объекта

### Проверка соответствия множества точек изображения и множества точек трехмерной модели объекта.

1. Предположим наличие соответствия между двумя множествами точек.
2. С учетом этого соответствия вычислим преобразование, преобразующее модельные точки в точки изображения.
3. Применим найденное преобразование к модельным точкам для получения множества преобразованных модельных точек.
4. Сравним преобразованные модельные точки с точками изображения, чтобы подтвердить или опровергнуть гипотезу о соответствии модельных точек и точек изображения.

**Алгоритм 14.2.** Основной алгоритм распознавания посредством совмещения.

показаны в алгоритме 14.2. В дальнейшем изложении модельные данные также будут называться «модельными точками», а данные измерений — «точками данных» или «точками изображения».

Далее более подробно будет рассмотрено распознавание посредством совмещения для случаев 3D-3D и 2D-3D.

**Совмещение типа 3D-3D** Предположим, что трехмерные модели представлены (или могут быть преобразованы в такое представление) в виде наборов характерных признаков, связанных с модельными точками (модельные точечные признаки). Если в качестве исходных данных используются дальнометрические измерения, то для распознавания объекта необходимы соответствующие пары трехмерных точек — «точка данных — модельный точечный признак». Процедура совмещения ищет соответствие между тремя выбранными модельными точечными признаками и тремя точечными признаками, вычисленными по данным измерений. Найденное соответствие определяет трехмерное преобразование, включающее в себя трехмерный поворот и перенос. Это преобразование переводит три модельные точки в соответствующие точки данных. Алгоритм выполнения совмещения описан в гл. 13 (алгоритм 13.3).

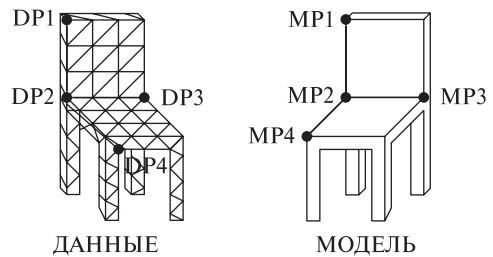
Корректное вычисление трехмерного преобразования по трем точкам возможно в случае правильного нахождения соответствующих точек и при отсутствии ошибок измерений. На практике так бывает редко, поэтому были разработаны более надежные методы, использующие обычно порядка десяти соответствующих точек.

После вычисления преобразования, предположительно выполняющего совмещение, это преобразование применяется ко всем модельным точкам для получения множества преобразованных модельных точек. Их можно сравнить с точками данных. Как и в двумерном случае, в процедуре верификации оценивается, насколько хорошо преобразованные модельные точки согласуются с

точками данных. В результате принимается решение об успешном выполнении совмещения или о необходимости перехода к следующему возможному набору соответствующих точек.

Как и для двумерного случая, были разработаны алгоритмы поиска пар соответствующих точек, учитывающие локальные характерные признаки или использующие специфические для конкретных задач способы группировки модельных точек и точек данных. На рис. 14.17 показан пример соответствия характерных признаков типа 3D-3D. Это соответствие было обнаружено посредством сопоставления трехсегментных трехмерных стыков модели стула и каркасной модели, построенной по трехмерным данным измерений.

**Рис. 14.17.** Соответствующие пары трехмерных модельных точек и трехмерных точек каркасной модели, построенной по данным измерений. Эти соответствующие точки можно использовать для вычисления преобразования, переводящего модельные точки в точки данных, и необходимого для выполнения процедуры совмещения типа 3D-3D.



#### Упражнение 14.5. Совмещение характерных признаков типа 3D-3D

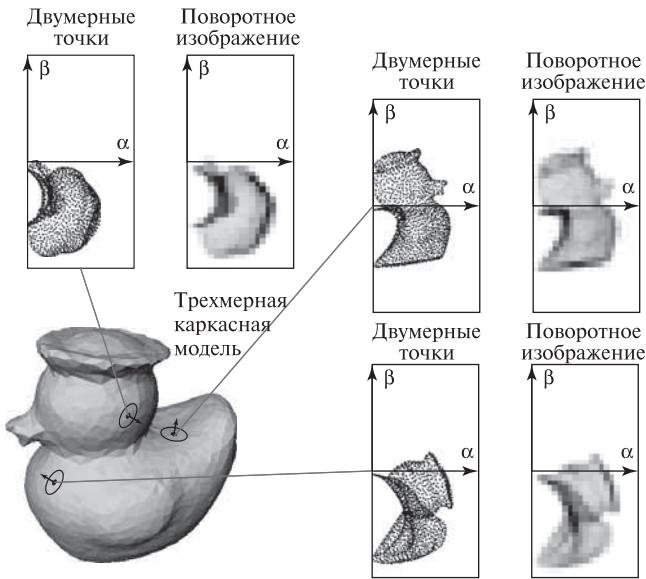
Стыки отрезков часто встречаются на изображениях многогранных объектов. Рассмотрите трехмерный объект — чашку, у которой есть цилиндрическая часть с цилиндрической полостью для жидкости и полукруглая ручка. Какие характерные признаки чашки, пригодные для поиска соответствующих точек при распознавании методом совмещения, можно обнаружить среди трехмерных данных измерений?

Выделение характерных признаков является очень важным этапом обработки. Описанная выше процедура будет хорошо работать с объектами, у которых легко обнаружить какие-либо различимые точки (например, угловые точки, вершины выступов, углубления). Если же поверхности объекта гладкие, а различимых точек нет или они редко встречаются, то необходим более совершенный метод для установления соответствия между моделью и данными измерений. Авторы статьи [Johnson, Hebert, 1998] разработали очень надежный метод именно для данной задачи. Они предложили представление трехмерного объекта, которое состоит из (1) трехмерной каркасной модели объекта (она хранит информацию о локальных характерных признаках формы объекта) и (2) из множества *поворотных изображений* (spin images), построенных на основе каркасной модели.

Если имеется каркасная модель трехмерного объекта, то в каждой вершине модели можно оценить нормаль к поверхности объекта. Между любой трехмерной точкой (вектором) и нормалью к поверхности в данной вершине можно определить отношение, представляемое с помощью двух параметров-расстояний  $\alpha$  и  $\beta$ .  $\alpha$  — это расстояние вдоль перпендикуляра к нормали поверхности, а  $\beta$  — расстояние со знаком вдоль перпендикуляра к касательной плоскости в данной вершине. Углы поворота в данном описании не учитываются, так как их нельзя определить однозначно.

*Поворотное изображение* — это двумерная гистограмма, которая вычисляется в выбранной вершине каркасной модели. С каждым поворотным изображением связано множество *составляющих модельных точек*, используемых при его построении. Объем, в котором заключены составляющие точки, зависит от двух параметров поворотного изображения:  $D$ , максимального расстояния между составляющей точкой и выбранной вершиной, и  $A$ , допустимого угла между нормалью в составляющей точке и нормалью в выбранной вершине. Поворотное изображение строится вокруг выбранной вершины  $o$  по отношению к множеству составляющих точек  $C$ , выбранных на основе указанных параметров поворотного изображения  $A$  и  $D$ . Представлением поворотного изображения служит аккумуляторный массив  $S[\alpha, \beta]$ , изначально заполняемый нулевыми значениями. Затем для каждой точки  $c \in C$  вычисляются ее параметры расстояний  $\alpha$  и  $\beta$  по отношению к выбранной вершине каркасной модели  $o$ , и инкрементируется соответствующий этим значениям  $\alpha$  и  $\beta$  элемент аккумуляторного массива. Размер элемента по порядку равен среднему расстоянию между вершинами трехмерной каркасной модели. Примеры поворотных изображений показаны на рис. 14.18.

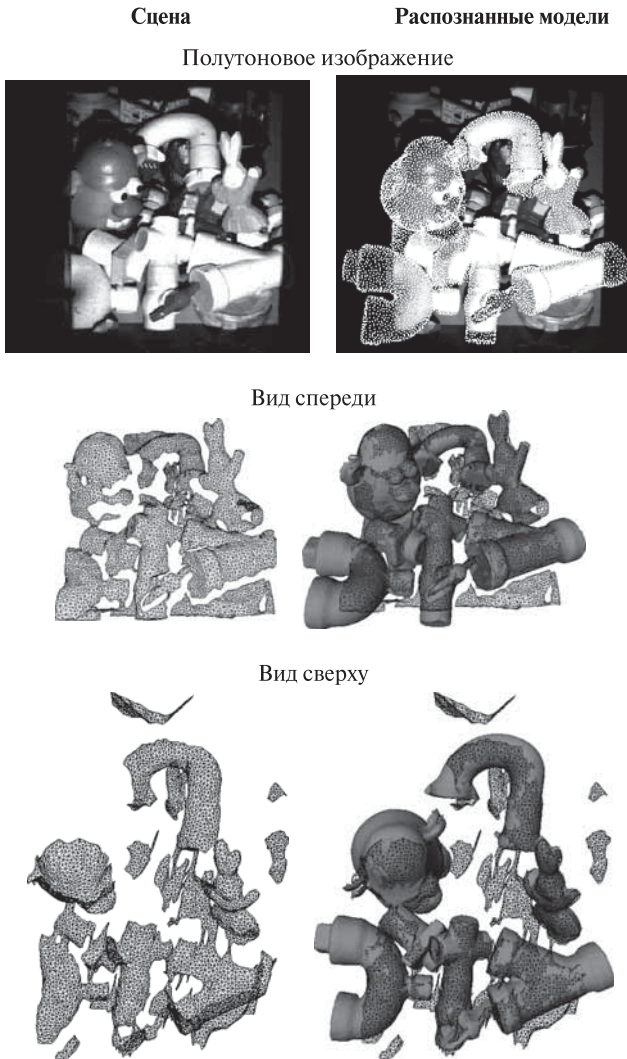
Поворотные изображения строятся для каждой вершины каркасной модели. Они содержат информацию о локальных свойствах формы объекта в окрестности каждой точки каркасной модели. При сопоставлении двух объектов используются два множества поворотных изображений. Путем вычисления коэффициента корреляции между парой поворотных изображений выполняется сравнение поворотного изображения в каждой точке первого объекта с поворотными изображениями всех точек второго объекта. Пары точек с большими значениями коэффициента корреляции дают пары соответствующих трехмерных точек, которые необходимы для сопоставления объектов. Соответствующие пары точек группируются и на основе соображений геометрической целостности объекта из них удаляются выбросы. Затем, как и при совмещении в общем случае, вычисляется трехмерное преобразование, которое используется для подтверждения успешности совмещения. На рис. 14.19 иллюстрируется работа метода распознавания с использованием поворотных изображений на примере сложного, зашумленного изображения. На этом изображении есть шесть различных объектов, модели которых известны системе распознавания (всего в базе данных системы распознавания хранятся модели двадцати различных объектов).



**Рис. 14.18.** Примеры поворотных изображений. (С любезного разрешения Andrew Johnson и IEEE. Воспроизведено по «Efficient Multiply Model Recognition in Cluttered 3-D Scenes», A.E. Johnson and M. Hebert, *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition*, June 1998. © 1998 IEEE.)

**Совмещение типа 2D-3D** Совмещение может применяться для сопоставления типа 2D-3D, когда имеется трехмерная модель объекта, а в качестве исходных данных используются двумерные изображения. В таком случае преобразование модельных точек в точки данных оказывается более сложным. Кроме трехмерного поворота и переноса, оно будет содержать перспективную проекцию. Полное преобразование можно определить на основе множества соответствующих точек, или множества соответствующих отрезков, или эллипса на плоскости, окружности в пространстве и одной пары соответствующих точек. Могут быть использованы комбинации характерных признаков всех трех перечисленных типов.

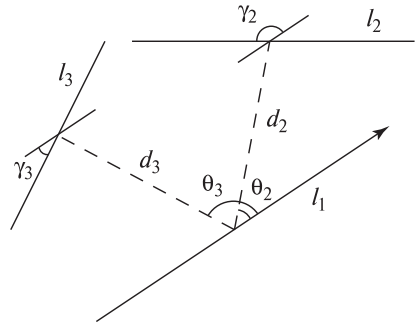
Вычисление преобразования между моделью и данными измерений является мощным средством сопоставления. Соответствующие признаки для расчета предполагаемого преобразования могут быть выбраны не абсолютно достоверно или с помощью реляционного сопоставления (см. разд. 14.4.2). Затем вычисленное преобразование применяется к характерным признакам трехмерной модели для получения двумерных характерных признаков, которые можно обнаружить в данных. Здесь возникает проблема, которая не встречалась при совмещении типа 2D-2D или 3D-3D. На любой двумерной перспективной проекции трехмерного объекта некоторые из преобразованных признаков появляются на поверхностях, которые не видны камере или закрыты



**Рис. 14.19.** Работа системы распознавания с использованием поворотных изображений. (С любезного разрешения Andrew Johnson и IEEE. Воспроизведено по «Efficient Multiply Model Recognition in Cluttered 3-D Scenes», A.E. Johnson and M. Hebert, *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition*, June 1998. © 1998 IEEE.)



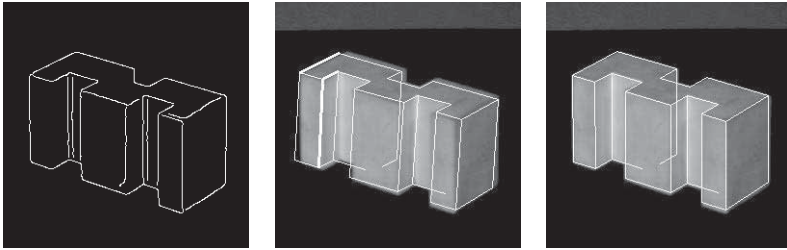
**Рис. 14.20.** Параметризация триплета отрезков в системе TRIBORS. Величины  $d_2$  и  $d_3$  — это расстояния от середины отрезка  $l_1$  до средних точек отрезков  $l_2$  и  $l_3$ , соответственно. Углы  $\gamma_2$  и  $\gamma_3$  — это углы, которые отрезки  $l_2$  и  $l_3$  образуют с отрезком  $l_1$ . Углы  $\theta_2$  и  $\theta_3$  — это углы между отрезком  $l_1$  и прямыми, соединяющими его с отрезками  $l_2$  и  $l_3$  (как на рис.).



другими поверхностями, более близкими к наблюдателю. Следовательно, для определения множества преобразованных признаков, которые можно сравнить с признаками изображения, требуется некоторый алгоритм поиска скрытых характерных признаков. Похожие алгоритмы удаления скрытых поверхностей применяются в машинной графике. При программной реализации они могут оказаться недопустимо медленными. В случае использования подходящих каркасных моделей и графической аппаратуры возможен полный расчет трехмерной сцены. В противном случае проблема скрытых признаков или игнорируется, или применяется приближенный алгоритм, который не гарантирует точности, но позволяет выполнить верификацию.

В системе распознавания объектов TRIBORS [Pulli, Shapiro, 1996] используются модели на основе классов видимости для распознавания многогранных объектов с плоскими гранями. Система выполняет поиск соответствующих пар между триплетами (тройками) модельных отрезков и триплетами отрезков на двумерном изображении. Модельные триплеты упорядочиваются на этапе обучения системы таким образом, чтобы триплеты с высокой вероятностью обнаружения на этапе сопоставления выбирались первыми, а триплеты с низкой вероятностью обнаружения не обрабатывались вообще. Триплет отрезков описывается вектором из девяти параметров, которые идентифицируют триплет в классе видимости. Параметризация триплета поясняется на рис. 14.20. Модельный триплет сопоставляется с триплетом с похожими параметрами на изображении. Если сопоставление считается успешным, то точки стыков отрезков из триплета изображения объединяются в пары с трехмерными вершинами модели. Затем для вычисления преобразования применяется итерационный точный алгоритм (см. гл. 13). Преобразование применяется к проволочной каркасной модели трехмерного объекта и с помощью алгоритма обнаружения скрытых линий определяются видимые ребра. С целью верификации для каждого предсказанного ребра на изображении ищется ближайший отрезок и производится сравнение этого отрезка с предсказанным ребром. Работа системы TRIBORS поясняется на рис. 14.21.





**Рис. 14.21.** (a) Ребра, обнаруженные на изображении трехмерного объекта. (b) Сопоставление триплетов отрезков (они показаны толстыми линиями) и начальная оценка положения объекта. (c) Окончательное сопоставление с моделью и оценка положения. (С любезного разрешения Kari Pulli.)

### Упражнение 14.6. Распознавание объектов в системе TRIBORS

Для поиска предполагаемых соответствий между модельными триплетами и триплетами отрезков на изображении система TRIBORS использует 9 параметров каждого триплета. Постройте несколько различных видов трехмерного многогранного объекта из одного класса видимости (см., например, стул на рис. 14.11). Найдите три главных отрезка, видимых на всех ваших видах, и вычислите 9 параметров этого триплета согласно рис. 14.20. Насколько похожи различные вычисленные вами векторы параметров? Сравните 9 параметров этих трех отрезков с параметрами какого-нибудь другого триплета. Насколько хорошо 9 параметров идентифицируют триплет?

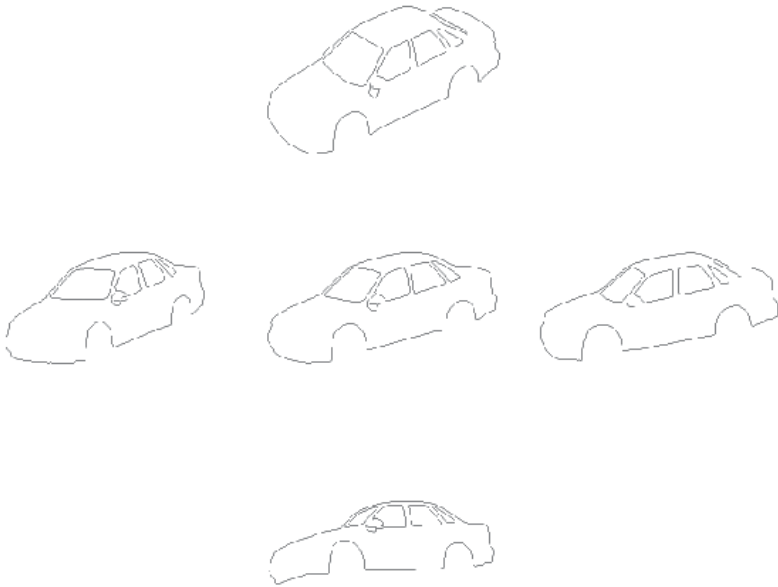
**Совмещение для объектов с гладкими поверхностями** Выше были рассмотрены совмещение трехмерной каркасной модели с трехмерными дальнометрическими данными и совмещение трехмерной каркасной модели многогранного объекта с двумерным полутоновым изображением. Теперь рассмотрим задачу распознавания и оценки положения произвольного трехмерного объекта по данным одного двумерного полутонового изображения. В описываемом решении применяется модель на основе классов видимости, но представление класса видимости отличается от наборов триплетов отрезков, которые используются для сопоставления в системе TRIBORS при низкоуровневой обработке изображений с выделенными перепадами яркости.

Обсуждаемый алгоритм основан на работе [Chen, Stockman, 1996], в которой описывается система для определения положения трехмерных объектов с гладкими поверхностями. В этой системе трехмерный объект моделируется набором « $2\frac{1}{2}$ -мерных» видов (они называются ракурсами модели), каждый из которых строится по 5 изображениям, полученных путем поворота точки наблюдения вверх, вниз, влево и вправо относительно центральной точки наблюдения. На рис. 14.22 показаны входные данные для построения одного ракурса модели автомобиля. Силуэт автомобиля на центральной карте перепадов сформирован путем выделения перепадов яркости и их сегментирования

на криволинейные сегменты. У этих сегментов вычисляются инвариантные характерные признаки, которые в процессе распознавания используются для индексированного доступа к ракурсу модели.

Трехмерные координаты точек  $(x, y, z)$ , соответствующие каждой двумерной точке края силуэта  $(u, v)$  на центральном изображении, вычисляются аналогично трехмерным координатам в стереосистеме. Верхнее и нижнее изображения используются для вычисления кривизны краев объекта в направлении оси  $y$ , а левое и правое изображения — для вычисления кривизны в направлении оси  $x$ . Аналогично, вычисления, похожие на стереоскопические, выполняются для определения пространственного положения сгибов и контрольных точек центральной карты перепадов (центрального изображения). Таким образом,  $2\frac{1}{2}$ -мерный ракурс модели состоит из трехмерных координат краев силуэта объекта, сгибов и контрольных точек, соответствующих центральному изображению с перепадами яркости. В представлении ракурса также хранится  $x$ - и  $y$ - кривизна для каждой из этих точек. С помощью информации о ракурсе впоследствии можно вычислить карту перепадов для любой другой точки наблюдения в данном классе видимости. Класс видимости описывается (1) множеством трехмерных точек и значений кривизны, упомянутых выше, и (2) множества инвариантных характерных признаков, используемых для индексированного доступа к ракурсам модели. Трехмерные точки вычисляются на основе соответствий, как в стереосистеме, между центральной и смежными картами перепадов (см. гл. 13). Инвариантные признаки вычисляются по двумерной карте перепадов центрального изображения (см. гл. 10).

В процессе обработки входного изображения для него строится карта перепадов и формируется множество криволинейных сегментов. Криволинейные сегменты используются для индексирования при работе с базой данных видов модели для формирования гипотез «объект-вид». Эти гипотезы, сгенерированные на основе схемы индексирования, проверяются в процедуре сопоставления. Каждая гипотеза включает идентификационные данные распознаваемого объекта и приблизительное положение. Верификация выполняется сглаживанием  $2\frac{1}{2}$ -мерного ракурса для каждой модели предполагаемого объекта относительно наблюдаемой карты перепадов. Изначально считается, что предполагаемый объект занимает положение, соответствующее центральному ракурсу модели, для которой выполняется сопоставление. Спроектированная карта перепадов этого ракурса модели сравнивается с наблюдаемой картой перепадов. В большинстве случаев они не будут совмещаться достаточно точно. Значит, при сопоставлении надо уточнить параметры положения, чтобы уменьшить двумерное расстояние между спроектированной модельной картой перепадов и наблюдаемой картой перепадов. Этап сопоставления поясняется на рис. 14.23. Карта перепадов вычисляется по исходному изображению (а), затем на этапе индексирования выдвигается гипотеза о положении объекта (b). На рис. 14.23 (с) показаны несколько итераций генерации границы модели, а на рис. 14.23 (d) — первый приемлемый результат сопоставления.



**Рис. 14.22.** Пять исходных изображений для построения одного ракурса модели автомобиля. (С любезного разрешения Jin-Long Chen.)

---

### Упражнение 14.7

Рассмотрите перечисленные далее прикладные задачи и объясните, сколько моделей объектов и сколько ракурсов может потребоваться в системах распознавания для решения этих задач. С какой точностью необходимо оценивать положение объектов? (а) Необходимо произвести инициализацию автоматического оборудования автомобильной мойки в соответствии с моделью поступившего автомобиля. (б) Система мониторинга и учета для автомобильного гаража должна определять модель каждого прибывшего или уехавшего автомобиля и фиксировать время этого события. (в) Система компьютерного зрения для автомобильной свалки должна определять количество и модели имеющихся машин.

---

### Упражнение 14.8

Для какого-либо объекта постройте ракурс модели с учетом кривизны и покажите, что на его основе можно генерировать силуэты объекта при малых поворотах. Например, рассмотрите тор с внешним диаметром 10 единиц и внутренним диаметром 1. Ракурс модели центрируется по виду, перпендикулярному к окружности внешнего диаметра. Определите множество трехмерных точек вдоль силуэта модели и значения  $x$ - и  $y$ - кривизны в этих точках. Затем с



**Рис. 14.23.** Сопоставление карт перепадов с использованием масштабного множителя  $s$  для масштабирования 1:2. (a) Наблюдаемая карта перепадов. (b) Модельная карта перепадов. (c) Сходимость алгоритма совмещения. (d) Сглаженная карта перепадов, показанная с наложением на исходное изображение. (С любезного разрешения Jin-Long Chen.)

помощью синтезированных изображений продемонстрируйте, как изменяется этот силуэт при малых поворотах.

#### 14.4.2 Распознавание на основе сопоставления исходных данных с реляционными моделями

Для перехода от геометрических моделей к символическим при распознавании трехмерных объектов могут использоваться реляционные модели (аналогично двумерному случаю). В алгоритме 14.3 в общем виде представлен метод распознавания на основе оценки реляционного расстояния (см. гл. 11). С целью упрощения этот алгоритм рассчитан на работу с одиночными отношениями. На практике используемые модели и методы выбираются с учетом природы исходных данных — трехмерных или двумерных.

**Трехмерные реляционные модели** Трехмерные реляционные модели состоят из трехмерных примитивов и пространственных отношений. В качестве примитивов могут быть выбраны элементы объемов и поверхностей, характерные признаки типа трехмерных отрезков и криволинейных сегментов. Обычным примером объемных примитивов являются обобщенные цилиндры. Вместе с ними в реляционную модель включаются какие-либо разновидности трехмерного отношения связности. Для распознавания трехмерных объектов

известны попытки использования *геонов* (geons, geometric ions) — объемных примитивов, предположительно используемых в человеческом зрении. Промышленно изготовленные объекты можно описать путем перечисления их плоских и цилиндрических поверхностей и указания отношения поверхностной смежности. В реляционных моделях вместе с пространственными отношениями различных видов (например, стык, параллельность и коллинеарность) могут быть использованы трехмерные отрезки и криволинейные сегменты.

Для построения грубых моделей трехмерных объектов были разработаны модели на основе объемных элементов трех типов — *стержней* (sticks), *пластин* (plates) и *капель* (blobs). Эти модели предназначены для описания и распознавания сложных искусственных объектов, содержащих много составных частей с плоскими или криволинейными поверхностями. Допускается существование разновидностей составных частей. Для грубого распознавания вместо точного описания каждой части (как в моделях типа «поверхность-ребро-вершина») можно классифицировать каждую часть как *стержень*, *пластину* или *каплю*. Стержни — это длинные тонкие части, у которых можно выделить один наиболее значимый размер. Пластины — это плоские, широкие части с двумя близко расположенными плоскими поверхностями, соединенными тонким краем между ними. У пластины есть два значимых размера, а у капли — три. Все три типа частей являются «почти выпуклыми», так что стержень не может изгибаться «слишком сильно», складки на поверхностях пластины не могут быть слишком большими, а у капли не может быть «слишком больших» вогнутостей. Несколько примеров стержней, пластин и капель показаны на рис. 14.24.

Модели типа «стержни-пластины-капли» описывают, как стержни, пластины и капли объединяются вместе для формирования объекта. Эти описания

**Требуется определить, являются ли два реляционных описания достаточно похожими, чтобы можно было считать их одинаковыми.**

**Входные данные:**

$\mathbf{P}$  — множество частей модели;

$\mathbf{L}$  — множество предполагаемых меток частей;

$\mathbf{R}_P$  — отношение на множестве частей;

$\mathbf{R}_L$  — отношение на множестве меток.

Надо найти отображение  $f$ , отображающее  $\mathbf{P}$  в  $\mathbf{L}$  при условии минимума ошибки  $E_S(f) = |\mathbf{R}_P \circ f - \mathbf{R}_L| + |\mathbf{R}_L \circ f^{-1} - \mathbf{R}_P|$ . Для поиска отображения могут использоваться деревья интерпретации, методы дискретной или вероятностной релаксации, или один из методов, описанных в гл. 11.

**Алгоритм 14.3.** Распознавание на основе оценки реляционного расстояния.

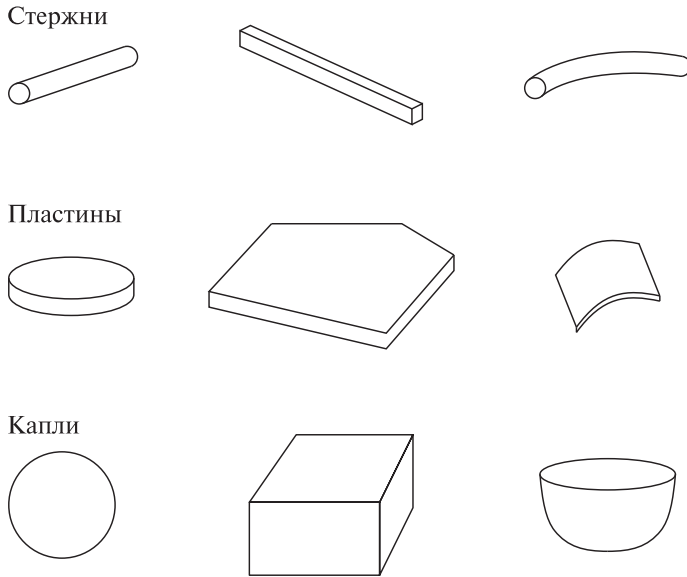
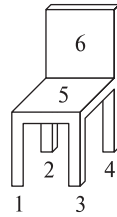


Рис. 14.24. Несколько примеров стержней, пластин и капель.

довольно грубые; в них точно не указывается местоположение физического соединения частей модели. У стержня есть две логические концевые точки, логическое множество внутренних точек и логический центр тяжести. Все эти точки могут быть указаны в качестве точек соединения. У пластины есть множество краевых точек, множество точек поверхности и центр тяжести. У капли есть множество поверхностных точек и центр тяжести. В моделях объектов используется только эта минимальная информация.

Реляционная структура модели типа «стержни-пластины-капли» является хорошим примером ясно детализированной символической модели объекта, которая была успешно использована в системе символического распознавания объектов. Реляционная модель содержит пять отношений. Унарное отношение *SIMPLE PARTS* — это список частей объекта. У каждой части есть несколько дескриптивных атрибутов, в том числе тип части (стержень, пластина или капля) и еще, возможно, числовая информация, относящаяся к размерам и/или форме части. Отношение *CONNECTS/SUPPORTS* содержит часть наиболее важной информации о структуре объекта. Оно состоит из 4 элементных кортежей вида  $(s_1, s_2, SUPPORTS, HOW)$ . Компоненты  $s_1$  и  $s_2$  являются простыми частями, *SUPPORTS* принимает истинное значение если  $s_1$  является основанием для  $s_2$  или ложное в противном случае, а *HOW* описывает тип соединения  $s_1$  и  $s_2$ .

Четыре оставшихся отношения служат для представления ограничений. Отношение *TRIPLE CONSTRAINT* содержит 4-элементные кортежи вида



SIMPLE-PARTS		CONNECTS-SUPPORTS			TRIPLES				
PART#	TYPE	SP1	SP2	SUPPORTS	HOW	SP1	SP2	SP3	SAME
1	Stick	1	5	True	end-edge	1	5	2	True
2	Stick	2	5	True	end-edge	1	5	3	True
3	Stick	3	5	True	end-edge	1	5	4	True
4	Stick	4	5	True	end-edge	1	5	6	False
5	Plate	5	6	True	edge-edge	2	5	3	True
6	Plate					2	5	4	True
						2	5	6	False
						3	5	4	True
						3	5	6	False
						4	5	6	False

PARALLEL		PERPENDICULAR	
SP1	SP2	SP1	SP2
1	2	1	5
1	3	2	5
1	4	3	5
2	3	4	5
2	4	5	6
3	4		

**Рис. 14.25.** Полная реляционная структура модели типа «стержни-пластины-капли» объекта-стула.

$(s_1, s_2, s_3, SAME)$ , где простая часть  $s_2$  касается и  $s_1$ , и  $s_3$ , а SAME равно истинному значению, если  $s_1$  и  $s_3$  касаются  $s_2$  с одного конца (или поверхности)  $s_2$  или ложному значению в противном случае. Отношения PARALLEL и PERPENDICULAR содержат пары вида  $(s_1, s_2)$ , где простые части модели  $s_1$  и  $s_2$  являются параллельными (или перпендикулярными). На рис. 14.25 показана модель типа «стержни-пластины-капли» для объекта-стула. Все стулья с похожими отношениями должны удовлетворять этой модели, независимо от точной формы их составных частей.

#### Упражнение 14.9. Модели типа «стержни-пластины-капли»

Нарисуйте простой многогранный объект-стол и постройте для него полную реляционную модель типа «стержни-пластины-капли».

**Реляционные модели на основе классов видимости** Если в качестве исходных данных для распознавания используются двумерные изображения,

то вместо истинных трехмерных моделей могут применяться модели на основе классов видимости. Обучение системы с целью построения таких моделей объектов проводится либо на основе искусственно сформированных данных, либо на основе набора реальных изображений объекта. Множество выделяемых на изображениях объектов двумерных характерных признаков зависит от класса объектов. Признаки, выделенные на каждом изображении из учебного набора, используются для формирования реляционного описания соответствующего вида объекта. Затем выполняется кластеризация реляционных описаний с целью формирования классов видимости объекта. Каждый класс видимости представляется в виде комбинированного реляционного описания, включающего характерные признаки, обнаруженные на всех видах этого класса видимости. Комбинированное реляционное описание служит реляционной моделью класса видимости. Обычно для каждого объекта строится порядка пяти классов видимости с собственными реляционными описаниями. Модели на основе классов видимости можно использовать для распознавания путем полного реляционного сопоставления. Если в базе данных хранится много различных моделей, то полный перебор всех моделей занимает много времени. Ускорить работу с базой данных позволяет реляционная индексация (см. гл. 11). Она будет рассмотрена в примере в данном параграфе.

Система распознавания объектов RIO (разработана Мауро Коста (Mauro Costa) в Вашингтонском университете) предназначена для распознавания трехмерных объектов на двумерных изображениях сцен с несколькими объектами. Изображения вводятся парами с одной камеры: одно изображение при включенном источнике света слева от сцены и одно — при включенном источнике света справа от сцены. По двум изображениям определяются области теней и бликов, что позволяет получить высококачественное изображение с перепадами яркости, соответствующими только объектам сцены. На изображении с перепадами яркости выделяются отрезки и круговые сегменты, служащие основой для формирования характерных признаков. На рис. 14.26 приведены примеры левого и правого изображений одной пары и изображение с выделенными перепадами яркости (карта перепадов яркости). На рис. 14.27 показаны отрезки и дуги окружностей, обнаруженные на карте перепадов яркости.

Система RIO рассчитана на работу с объектами с плоскими, цилиндрическими и резбовыми поверхностями. Эти ограничения обуславливают выбор высокоуровневых характерных признаков. Система RIO учитывает десять характерных признаков: эллипсы, соосные дуги (две, три или больше), параллельные пары отрезков (близко или далеко расположенных), триплеты отрезков (в форме U и Z), L-стыки, Y-стыки и V-стыки. На рис. 14.28 показаны примеры характерных признаков, найденных среди отрезков и дуг рис. 14.27. Среди отрезковых признаков есть два L-стыка и пара параллельных прямых. В кластере дуг показаны три соосные дуги. Отметим, что не каждый отрезок и не каждая дуга становятся частью характерного признака, который будет

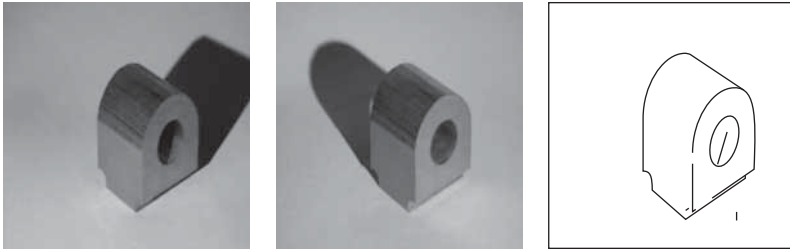


использоваться при распознавании. На рис. 14.29 показано полное множество характерных признаков системы RIO.

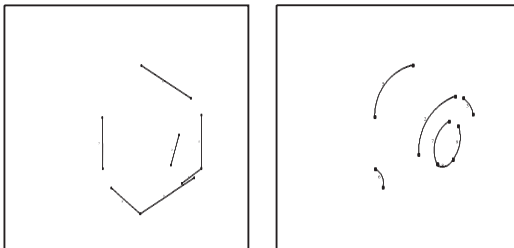
Кроме маркированных признаков, в системе RIO для распознавания объектов используются маркированные бинарные отношения между признаками. Учитываются следующие отношения: совместное использование одной дуги, совместное использование одного отрезка, совместное использование двух отрезков, соосность, близость в концевых точках и отношения «окружать»/«быть окруженным» (см. рис. 14.30).

Для каждого модельного вида строится структурное описание в форме графа. Узлами графа являются характерные признаки различных типов, а ребрами — отношения между признаками. Для применения реляционной индексации этот граф разбивается на множество 2-графов. 2-граф — это граф с двумя узлами, соединенных отношением. На рис. 14.31 показан один вид модели объекта «шестигранная гайка», часть полного графа для трех признаков объекта и их отношений, а также разбиение этой части графа на 2-графы.

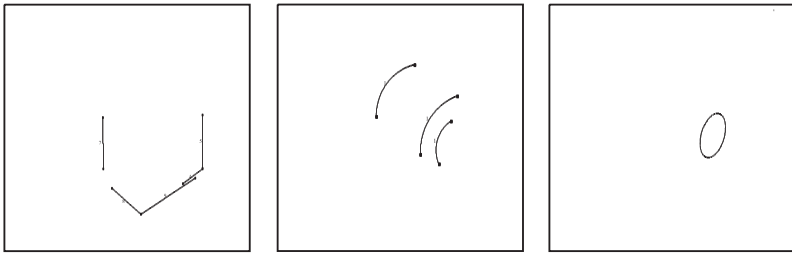
В процедуре распознавания возникает проблема сопоставления неизвестного изображения с потенциально большой базой данных видов моделей объек-



**Рис. 14.26.** «Левое» и «правое» изображения искусственного промышленно изготовленного объекта и перепады яркости, выделенные на этапе обработки изображений после удаления теней и большинства бликов. (С любезного разрешения Mauro Costa.)



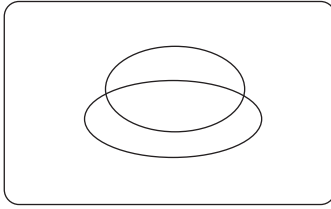
**Рис. 14.27.** Отрезки и дуги окружностей, найденные на карте перепадов яркости с рис. 14.26. (С любезного разрешения Mauro Costa.)



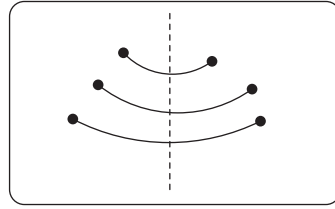
**Рис. 14.28.** Характерные признаки — отрезки, дуги окружностей и эллипсов — найденные среди отрезков и дуг на рис. 14.27. (С любезного разрешения Мауго Costa.)

тов. Реляционная индексация позволяет получить небольшой набор гипотез о присутствующих на изображении объектах. Для применения индексации необходим предварительный этап подготовки структуры данных — хэш-таблицы, которая будет использоваться при обращении к базе данных на этапе распознавания объектов. Индексами для хэш-таблицы служат 4-элементные кортежи, представляющие 2-графы модельных видов объекта. Компонентами кортежа являются типы двух узлов и типы двух отношений. Например, 4-элементный кортеж (эллипс, пара далеко расположенных параллельных отрезков, быть окруженным, окружает) означает, что 2-граф представляет характерные признаки «эллипс» и «параллельная пара далеко расположенных отрезков», где эллипс окружен парой параллельных отрезков, и, значит, эта пара окружает эллипс. Так как в системе RIO большинство отношений симметричны, то два отношения обычно оказываются одинаковыми. Например, 4-элементный кортеж (эллипс, кластер соосных дуг, совместное использование дуги, совместное использование дуги) описывает отношение, в котором у эллипса и кластера соосных дуг есть общий сегмент окружности. Для хэширования символические компоненты 4-элементных кортежей преобразуются в числовое представление. На предварительном этапе производится обработка всех модельных видов в базе данных. В результате каждый 2-граф из базы данных дает индекс в виде 4-элементного кортежа. Имя модельного вида и связанная с ним информация сохраняется в форме списка в указанном кармане хэш-таблицы.

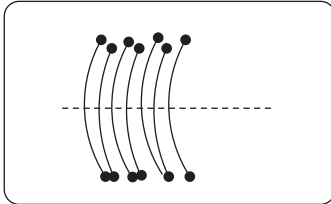
После формирования хэш-таблицы она может быть использована для распознавания в основном режиме работы системы. Также создается набор аккумуляторов голосования, по одному для каждого модельного вида в базе данных. При анализе сцены сначала выделяются ее характерные признаки и строится реляционное описание в форме набора 2-графов. Затем каждый 2-граф этого описания кодируется для получения индекса для обращения к хэш-таблице. Из нее извлекается список, связанный с указанным карманом таблицы. Он содержит все модельные виды, содержащие данный 2-граф. Затем для каждого модельного вида из списка проводится голосование. Описанные действия выполняются для всех 2-графов, найденных на анализируемом изображении сцены. В конце процедуры в качестве гипотез распознавания выбира-



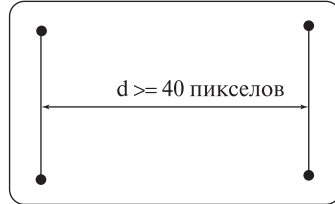
(a) Эллипсы



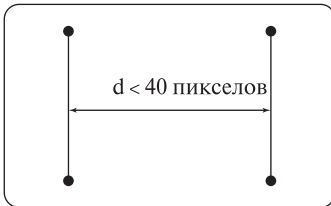
(b) 3 соосные дуги



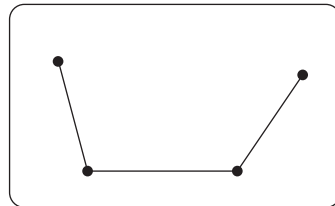
(c) Множественная соосность



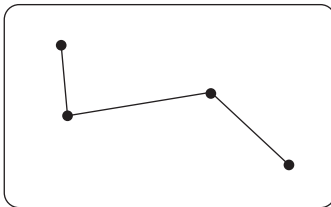
(d) Параллельные далеко расположенные отрезки



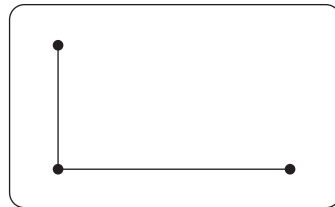
(e) Параллельные близко расположенные отрезки



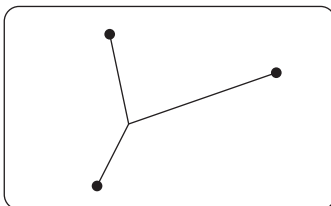
(f) U-триплет



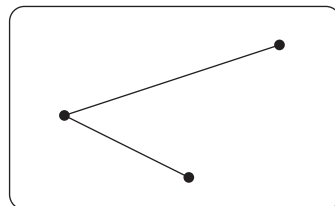
(g) Z-триплет



(h) L-стык

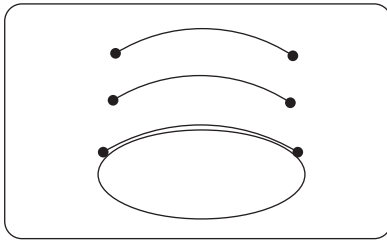


(i) Y-стык

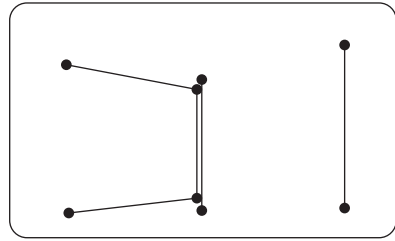


(j) V-стык

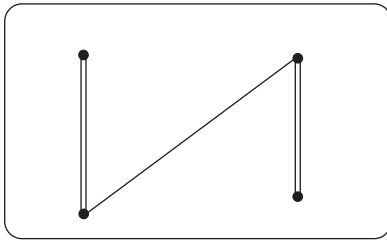
**Рис. 14.29.** Характерные признаки, используемые в системе распознавания объектов RIO. (С любезного разрешения Mauro Costa.)



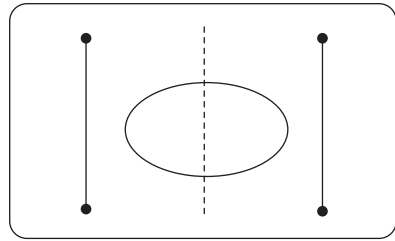
(a) Совместное использование одной дуги



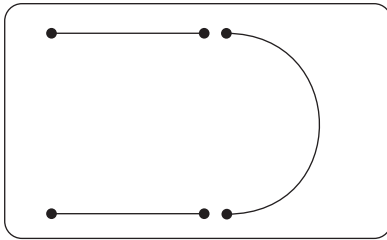
(b) Совместное использование одного отрезка



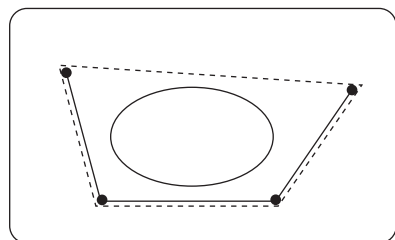
(c) Совместное использование двух отрезков



(d) Соосность



(e) Близость в конечных точках

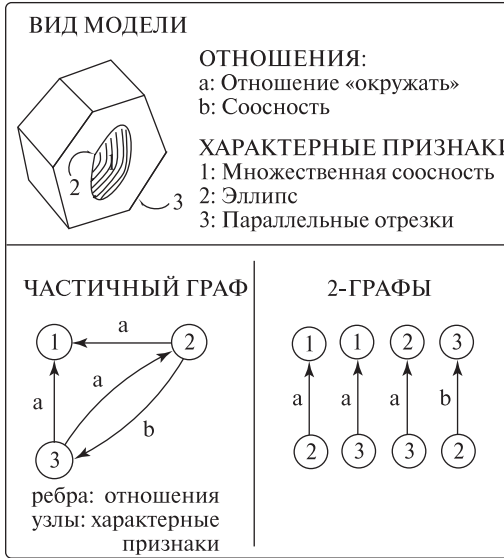


(f) Отношения «окужать»/«быть окуженным»

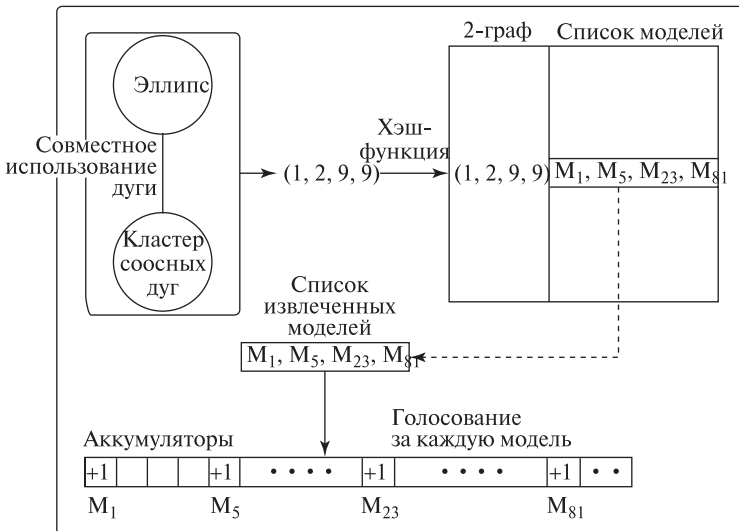
**Рис. 14.30.** Примеры отношений между парами характерных признаков. (С любезного разрешения Mauro Costa.)

ются модельные виды с наибольшими значениями аккумуляторов голосования. Процесс распознавания в системе RIO поясняется на рис. 14.32. Показанный на рисунке 2-граф преобразуется в числовой 4-элементный кортеж (1, 2, 9, 9), который указывает на карман хэш-таблицы. Из него извлекается список из четырех моделей  $M_1$ ,  $M_5$ ,  $M_{23}$  и  $M_{81}$ . Для каждой из них увеличиваются значения аккумуляторов.

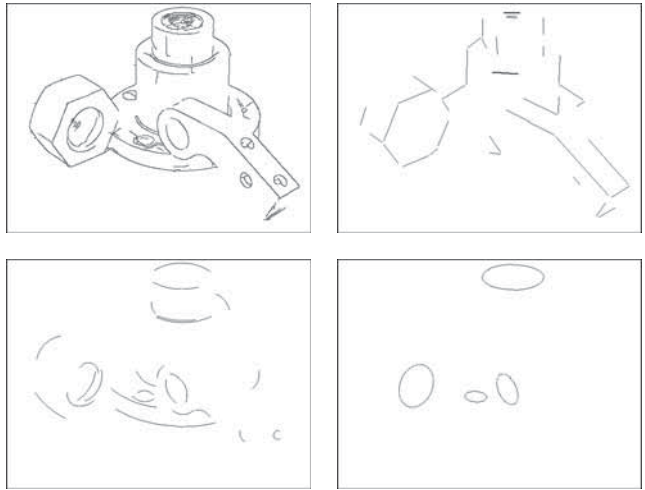
После генерации гипотез выполняется верификация. На этапе реляционной индексации было установлено соответствие между двумерными характерными признаками изображения и двумерными признаками модельного вида. Двумерные модельные признаки связаны с признаками трехмерной модели предполагаемого объекта. В системе RIO для верификации используются соответствующие пары точек типа 2D-3D, пары отрезков 2D-3D и пары эллипс



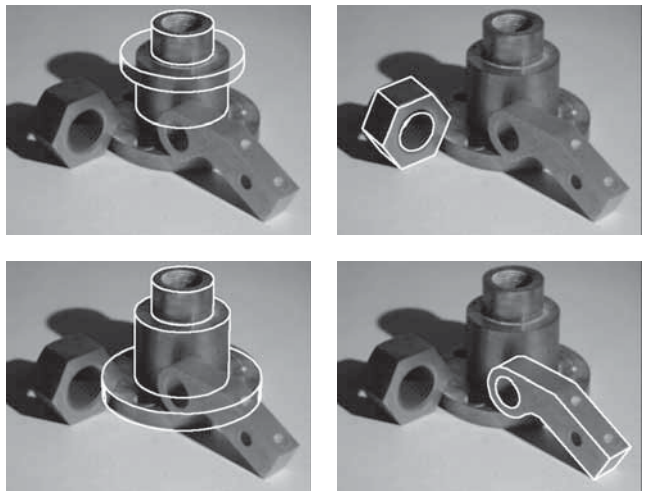
**Рис. 14.31.** Фрагмент графа и соответствующих 2-графов для объекта «шестигранная гайка». (С любезного разрешения Mauro Costa.)



**Рис. 14.32.** Схема голосования для реляционной индексации. (С любезного разрешения Mauro Costa.)

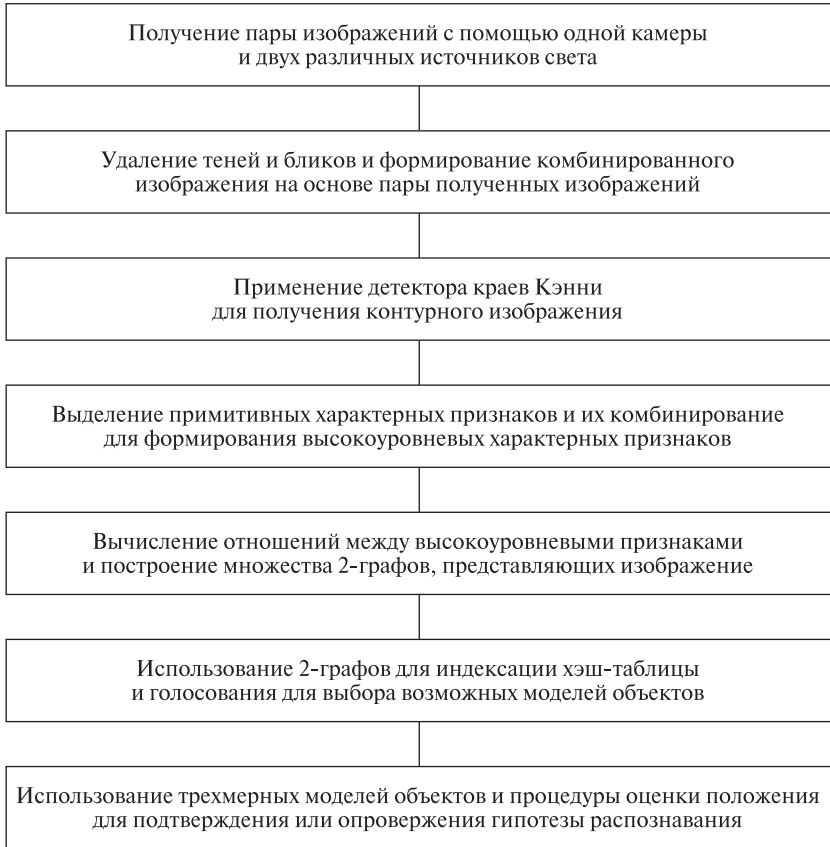


**Рис. 14.33.** Тестовое изображение и его характерные признаки — отрезки, дуги окружностей и эллипсов. (С любезного разрешения Mauro Costa.)



**Рис. 14.34.** Некорректная (слева вверху) и три корректных гипотезы. (С любезного разрешения Mauro Costa.)

2D-окружность 3D. Вычисляется преобразование для перехода от трехмерной модели предполагаемого объекта к изображению. Отрезки и дуги окружностей трехмерной модели проектируются на плоскость изображения и вычисляется расстояние, значение которого определяет успешность верификации или некорректность гипотезы распознавания. Пример работы системы RIO приведен на рис. 14.33 и 14.34. На рис. 14.33 показана карта перепадов яркости для сцены с несколькими объектами. Среди перепадов яркости найдены характерные признаки — отрезки, дуги окружностей и эллипсов. На рис. 14.34 показана некорректная гипотеза, которая была отброшена процедурой верификации, а также есть три корректных гипотезы, подтвержденных при верификации.



**Рис. 14.35.** Схема работы системы распознавания объектов RIO.

В системе RIO реализована процедура оценки местоположения/ориентации, описанная в гл. 13. На рис. 14.35 дана общая блок-схема системы RIO.

### Упражнение 14.10. Реляционная индексация

Напишите программу, реализующую реляционную индексацию для распознавания объектов. Программа должна использовать библиотеку моделей объектов, каждый из которых представляется набором из 2-графов. Входными данными для этапа распознавания является описание изображения сцены с несколькими объектами в форме 2-графов. Программа должна возвращать список со всеми моделями из базы данных, для которых на изображении нашлось не менее 50% 2-графов.

### 14.4.3 Распознавание на основе сопоставления с функциональными моделями

Геометрические модели точно описывают форму конкретных объектов. Модель, построенная с помощью системы САПР, описывает один объект со всеми его особыми точками и размерами. Реляционные модели являются моделями более общего вида, так как они описывают целые классы объектов, имеющих одинаковую реляционную структуру. Например, стул можно описать как объект, у которого есть спинка, сиденье и четыре ножки, присоединенных снизу около углов сиденья. Стул, у которого вместо четырех ножек под сиденьем расположено основание в виде тумбы, такому описанию не соответствует. Функциональный подход к распознаванию объектов идет еще на один шаг дальше. В нем предпринимается попытка определить классы объектов на основе их функционального назначения. Например, стул — это некоторый объект, на котором может сидеть человек. У стула может быть несколько различных реляционных структур, но все они удовлетворяют одному набору функциональных ограничений.

Первой работой по распознаванию объектов на основе функционального подхода можно считать работу [Stark, Bowyer, 1996] о системе GRUFF. Название этой системы распознавания является сокращением от Generic Object Recognition Using Form and Function (Универсальная система распознавания объектов с использованием характеристик формы и функциональных свойств). В системе GRUFF выделены три уровня знаний:

1. иерархия категорий всех объектов в базе знаний;
2. определение каждой категории в терминах функциональных свойств;
3. примитивы знаний, используемые для построения функциональных описаний.

**Примитивы знаний** Каждый примитив знаний является параметризованной процедурой, которая реализует базовое понятие, связанное с геометрическими, физическими или функциональными свойствами (функциональное свойство — это действие, приводящее к заданному результату). Примитив знаний получает в качестве входных данных часть описания пространственной формы и возвращает значение, обозначающее, насколько хорошо входные данные удовлетворяют заданным требованиям. В системе GRUFF определены 6 примитивов знаний, реализующих следующие понятия:

- относительная ориентация;
- размеры;
- близость;
- устойчивость;
- отсутствие препятствий;
- огораживание.



Примитив *относительной ориентации* (relative orientation) используется для определения того, насколько хорошо относительное расположение двух поверхностей соответствует некоторому желаемому взаимному размещению. Например, верхняя поверхность сиденья стула должна быть примерно перпендикулярна соседней поверхности спинки. Примитив *размеров* (dimensions) выполняет проверку размеров шести возможных типов: ширины, глубины, высоты, площади, непрерывной поверхности и объема. У большинства объектов размеры одной части объекта служат ограничением размеров других частей. Примитив *близости* (proximity) проверяет качественные пространственные отношения между элементами формы объектов. Например, ручка кувшина, чтобы его было легче поднимать, должна быть расположена выше центра тяжести объекта.

Примитив *устойчивости* (stability) проверяет, что данная фигура в данной ориентации остается устойчивой в случае расположения на поддерживающей плоскости даже под действием некоторой силы. Примитив *отсутствия препятствий* (clearance) проверяет, является ли указанный объем пространства между частями объекта свободным от преград. Например, чтобы на стул можно было сесть, должен быть свободен прямоугольный объем над сиденьем. И, наконец, примитив *огораживания* (enclosure) проверяет наличие требуемых полостей и вогнутостей объекта. Например, у бокала для вина должна быть полость, в которую можно налить жидкость.

**Функциональные свойства** В определении функционального класса в терминах примитивов знаний задаются функциональные свойства, которыми должны обладать объекты данного класса. В системе GRUFF для объектов классов *мебель*, *посуда* и *ручные инструменты* были выделены функциональные категории, определяемые четырьмя возможными шаблонами:

- обеспечивает устойчивость для  $X$ ;
- обеспечивает поверхность для  $X$ ;
- обеспечивает хранилище для  $X$ ;
- обеспечивает рукоять для  $X$ ;

где  $X$  является параметром шаблона. Например, для сидящего на стуле человека стул должен обеспечивать устойчивую поддержку и иметь пригодную для сидения поверхность. У супницы должно быть устойчивое хранилище для супа, а у чашки — подходящая ручка, позволяющая поднять ее и соответствующая размерам чашки.

**Иерархия категорий** В системе GRUFF все категории объектов объединены в дерево категорий. В нем показаны все категории объектов, которые может распознавать система. На верхнем уровне дерева расположены наиболее общие категории, например, мебель и посуда. Каждый следующий уровень соответствует большей детализации. Например, у узла «мебель» есть дочерние узлы классов объектов «стул», «стол», «скамья», «книжная полка» и «кро-

вать». Эти классы объектов можно разделить дальше. Например, среди стульев можно выделить обычные стулья, шезлонги, кресла-качалки и др. Фрагмент дерева категорий системы GRUFF показан на рис. 14.36.

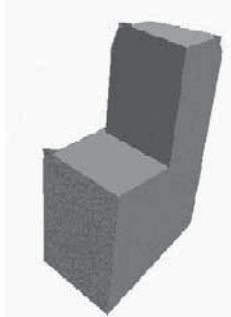
Функциональное определение категорий объектов в системе GRUFF служит скорее не для «распознавания объектов», а для принятия решения, может ли наблюдаемый (в дальнометрических данных) объект действовать как элемент данной категории. В процессе этой функциональной обработки выделены два основных этапа: предварительный этап и этап распознавания. Предварительный этап не зависит от категорий; все объекты обрабатываются одинаково. На этом этапе выполняется анализ трехмерных дальнометрических данных и среди них перебираются все возможные функциональные элементы. На этапе распознавания эти элементы используются для построения индексов, которые используются для упорядочения категорий объектов. Индекс состоит из функционального элемента, а также его площади и объема. Те категории, которые невозможно использовать для сопоставления на основе данных индекса, из поиска исключаются. Остальные упорядочиваются для дальнейшей оценки. Для каждой гипотезы о классе объекта сначала вызываются примитивы знаний этого класса. Они позволяют вычислить, насколько хорошо функциональный элемент, найденный в исходных данных, удовлетворяет требованиям данного класса. Каждый примитив знаний возвращает вычисленную численную оценку. Они комбинируются так, чтобы получить



Рис. 14.36. Фрагмент дерева категорий системы GRUFF. (С любезного разрешения Louise Stark, Kevin Bowyer.)



Дальнометрическое изображение



Сегментированное дальнометрическое изображение

**Рис. 14.37.** Входные данные для системы GRUFF. (С любезного разрешения Louise Stark, Kevin Bowyer.)

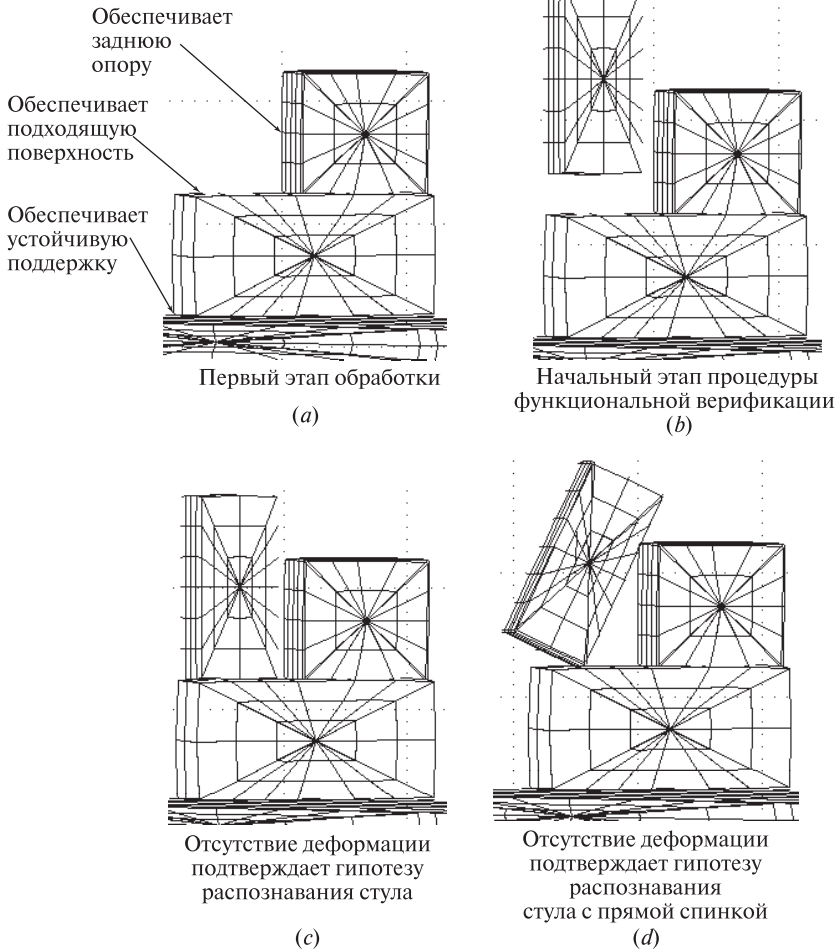
окончательную ассоциативную оценку, описывающую, насколько хорошо полный набор функциональных элементов из исходных данных соответствует предполагаемой категории объектов. На рис. 14.37 показан пример входных данных для системы GRUFF, а на рис. 14.38 показана часть функциональных рассуждений, возникших при обработке этих данных.

#### **Упражнение 14.11.** Функциональный метод распознавания объектов

Допустим, есть два объекта-стола: у одного из них 4 ножки около углов крышки, а у другого вместо ножек тумба. Какие общие свойства этих объектов могут быть использованы в системе функционального распознавания объектов, чтобы оба этих объекта можно было отнести к одному классу?

#### **14.4.4 Распознавание по внешнему виду объектов**

В большинстве схем систем распознавания трехмерных объектов модель является самостоятельной сущностью, существующей независимо от двумерных изображений объекта. В данном параграфе будет рассмотрен подход, в котором объект запоминается в виде набора своих двумерных изображений. Тогда распознавание можно произвести, сопоставляя входное изображение неизвестного объекта с изображениями, хранящимися в памяти системы. Представление объекта хранится на *сигнальном уровне* (т. е. на уровне исходных данных датчика), а сопоставление выполняется путем непосредственного сравнения полутоновых изображений. Высокоуровневые характерные признаки (например, поиск частей объекта) при этом не используются. Следовательно, при таком подходе возможно сокращение времени на разработку сложных для тестирования программных алгоритмов. Некоторые проблемы, присущие подходу на основе сопоставления на сигнальном уровне, будут рассмотрены ниже. Простота методов распознавания по внешнему виду позволили опробовать их на больших наборах изображений, и был получен ряд значительных ре-



**Рис. 14.38.** Этапы обработки данных в системе GRUFF. (С любезного разрешения Louise Stark, Kevin Bowyer.)

зультатов. Возможно, наиболее важные результаты были получены в задаче распознавания человеческих лиц. Эту задачу мы будем использовать в данном параграфе в качестве поясняющего примера.

В общих чертах распознавание по внешнему виду включает два этапа обработки:

- На этапе обучения системы создается база данных маркированных (снабженных метками) изображений. Обозначим ее  $\mathbf{DB} = \{\langle I_j[\ ], L_j \rangle_{j=1,k}\}$ , где  $I_j$  — это  $j$ -е эталонное изображение, а  $L_j$  — метка этого изображения.

- На этапе распознавания неизвестный объект распознается путем сравнения его изображения  $I_u$  с изображениями из базы данных. Объекту присваивается метка  $L_j$  от наиболее похожего эталонного изображения  $I_j$ . Наиболее похожее эталонное изображение  $I_j$  можно найти из условия минимума евклидова расстояния  $\|I_u[\ ] - I_j[\ ]\|$  или из условия максимума скалярного произведения  $I_u \circ I_j$  (см. гл. 5).

Конечно, для каждого этапа можно отметить ряд существенных уточнений.

- Эталонные изображения должны быть представительными изображениями распознаваемых объектов. В случае человеческих лиц (и многих других объектов) на эталонных изображениях должны быть учтены различные выражения лица, изменение освещения, малые повороты головы вокруг различных осей.
- Объект должен быть отчетливо ограничен. Положение и размеры всех лиц должны быть примерно одинаковыми. В противном случае надо будет вычислять параметры положения и размера.
- Так как в данном методе объект не отделяется от фона, то фон должен быть включен в принятие решения при распознавании и его надо учесть при обучении системы.
- Даже при малых размерах изображений, например,  $100 \times 100$ , которых достаточно для распознавания лиц, размерность пространства всех изображений составляет 10000. Вероятно, количество эталонных образцов намного меньше этого числа. Следовательно, потребуется использовать некоторый метод *уменьшения размерности*.

Прежде чем продолжить чтение, читатель может подумать о вопросе различения лиц двух классов — с очками и без очков; или, автомобилей с радиантеннами и без них. Можно ли обнаружить эти различия при совпадении всех остальных свойств объектов?

Теперь остановимся на важной проблеме уменьшения количества сигнальных признаков, используемых для представления наших объектов. Известно, что при распознавании лиц размерность можно уменьшить со  $100 \times 100$  до 15, при этом обеспечивается распознавание с точностью 97%. В гл. 5 обсуждались различные базисы пространства изображений размеров  $R \times C$  и было показано, как изображение можно представить в виде суммы особых изображений, таких, как элементарные края, волнообразная рябь и др. Также было показано, что при представлении изображения в виде линейной комбинации изображений ортонормированного базиса энергия этого изображения равна сумме квадратов коэффициентов.

**Базисные изображения для набора эталонных изображений** Будем полагать, что можно найти ортонормированное базисное множество изображений  $\mathbf{B}$ , обладающее следующими свойствами:

1.  $\mathbf{B} = \{F_1, F_2, \dots, F_m\}$ , где  $m$  намного меньше  $N = R \times C$ .

2. Среднее качество представления набора изображений с использованием этого базиса является удовлетворительным в следующем смысле. Для всех  $M$  изображений  $I_j$  из эталонного множества имеем, we have

$$I_j^m = a_{j1}F_1 + a_{j2}F_2 + \dots + a_{jm}F_m$$

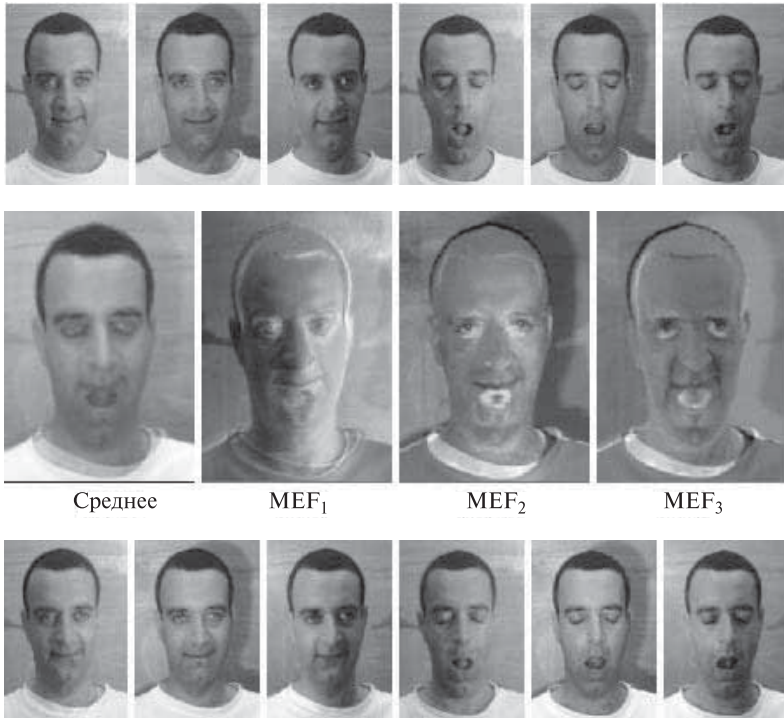
и

$$\sum_{j=1}^m (\|I_j^m - I_j\|^2 / \|I_j\|^2) < P\%.$$

$I_j^m$  — это аппроксимация исходного изображения  $I_j$  в виде линейной комбинации  $m$  базисных изображений.

В верхней строке на рис. 14.39 показаны 6 эталонных изображений лица одного из многих людей из базы данных, подготовленной в Институте Вейцмана. В средней строке на этом рисунке приведены четыре базисных изображения, которые были получены для представления набора эталонных изображений; самое левое из базисных изображений получено усреднением эталонных образцов. В нижней строке рисунка показано, как 6 исходных изображений лиц будут выглядеть в случае представления в форме линейной комбинации четырех базисных векторов. В нескольких исследовательских проектах было продемонстрировано, что  $m = 15$  или  $m = 20$  базисных изображений достаточно для представления базы данных с изображениями лиц (например, 3000 изображений лиц в одной из работ [Pentland, 1986]) таким образом, что в среднем аппроксимированное изображение  $I_j^m$  отличается от  $I_j$  менее, чем на 5%. Следовательно, сопоставление с использованием такой аппроксимации будет давать почти те же результаты, что и сопоставление с использованием исходного изображения. Важно отметить, что для базы данных, показанной на рис. 14.39, **каждое эталонное изображение может быть представлено в памяти в виде всего четырех чисел**, что позволяет организовать эффективное сравнение с неизвестными изображениями. Если предусмотреть хранение четырех базисных векторов в памяти, то при необходимости можно сгенерировать аппроксимированное представление исходного изображения лица. (Отметим, что первый базисный вектор является осреднением изображений эталонного набора и в действительности не является элементом ортонормированного множества.)

**Вычисление базисных изображений** Работа с базисным множеством  $V$  позволяет сильно уменьшить требования к объему памяти и ускорить вычисления, так как  $m$  намного меньше  $N$ , количества пикселей в исходном изображении. Базисные изображения  $F_i$  называются *главными компонентами* (principal components) множества эталонных образцов. Ниже приведен алгоритм 14.4, в котором дается схема распознавания по внешнему виду с использованием главных компонент. В алгоритме есть две части: предварительный этап обучения и этап распознавания. На этапе обучения сначала вычисляется среднее по эталонным изображениям. Затем они используются



**Рис. 14.39.** (Верхняя строка) шесть эталонных изображений лица одного из многих людей, лица которых хранятся в базе данных. (Средняя строка) усредненное эталонное изображение и три наиболее значимых собственных вектора, полученных из разреженной матрицы. (Нижняя строка) изображения верхней строки, представленные в виде линейной комбинации всего четырех изображений из средней строки. (С любезного разрешения Yael Moses и John Weng.)

для генерации множества разностных изображений  $\Phi$ , каждое из которых представляет разность между эталонным и усредненным изображением. Если представлять себе каждое разностное изображение  $\Phi_i$  как вектор из  $N$  элементов, то  $\Phi$  является массивом из  $R$  строк и  $C$  столбцов. На следующем этапе вычисляется ковариационная матрица эталонных изображений  $\Sigma_\Phi$ . По определению, элемент этой матрицы  $\Sigma_\Phi[i, i]$  равен дисперсии  $i$ -го пиксела, а  $\Sigma_\Phi[i, j]$  — ковариации  $i$ -го и  $j$ -го пиксела по всем эталонным изображениям. Так как усредненное и разностное изображения уже были вычислены ранее, то ковариационная матрица определяется так:

$$\Sigma_\Phi = \Phi^T \Phi \quad (14.5)$$

Размеры ковариационной матрицы очень велики,  $N \times N$ , где  $N$  — количество пикселей изображения. Это обычно составляет  $256 \times 256$  или даже  $512 \times 512$ . Поэтому вычисление собственных векторов и собственных значений на сле-

**Предварительный этап обучения:**

**Входные данные:** множество  $\mathbf{I}$  из  $M$  маркированных эталонных изображений

**Выходные данные:** базисное множество  $\mathbf{B}$  и вектор коэффициентов для каждого изображения.

Обозначения:

$\mathbf{I} = \{I_1, I_2, \dots, I_M\}$  — множество эталонных изображений (вход)

$\mathbf{B} = \{F_1, F_2, \dots, F_m\}$  — множество базисных векторов (выход)

$\mathbf{A}_j = [a_{j1}, a_{j2}, \dots, a_{jm}]$  — вектор коэффициентов для изображения  $I_j$  (выход)

1. Вычисление усредненного изображения  $\mathbf{I}_{\text{mean}} = \text{mean}(\mathbf{I})$ .
2. Вычисление множества разностных изображений  $\Phi = \{\Phi_i | \Phi_i = I_i - \mathbf{I}_{\text{mean}}\}$ .
3. На основе  $\Phi$  вычисление ковариационной матрицы  $\Sigma_\Phi$ .
4. Использование метода главных компонент для вычисления собственных векторов и собственных значений матрицы  $\Sigma_\Phi$  (см. описание в тексте).
5. Построение вектора  $\mathbf{B}$  — базисного множества — путем выбора  $m$  наиболее значимых собственных векторов. Выбор начинается с наибольшего собственного значения и продолжается по уменьшению собственных значений, при этом выбираются соответствующие собственные векторы.
6. Представление каждого эталонного изображения  $I_j$  в виде линейной комбинации базисных векторов:  $I_j^m = a_{j1}F_1 + a_{j2}F_2 + \dots + a_{jm}F_m$ .

**Этап распознавания:**

**Входные данные:** множество базисных векторов  $\mathbf{B}$ , база данных наборов коэффициентов  $\{\mathbf{A}_j\}$ ,

изображение для распознавания  $\mathbf{I}_u$ . **Выходные данные:** метка класса для  $\mathbf{I}_u$ .

1. Вычисление для изображения  $\mathbf{I}_u$  вектора коэффициентов  $\mathbf{A}_u = [a_{u1}, a_{u2}, \dots, a_{um}]$ ;
2. Поиск в множестве  $\{\mathbf{A}_j\}$   $h$  ближайших соседей вектора  $\mathbf{A}_u$ ;
3. Принятие решение относительно класса  $\mathbf{I}_u$  путем выбора среди меток  $h$  ближайших соседей (возможно, отмена распознавания в случае, если соседи слишком далеки или их метки противоречивы).

**Алгоритм 14.4.** Распознавание по внешнему виду объектов с использованием базиса главных компонент.



дующем этапе алгоритма оказывается очень длительной процедурой, если непосредственно пользоваться матрицей  $\Sigma_{\Phi}$  (алгоритм вычисления главных компонент см. в книге *Numerical Recipes in C* [Vetterling, 1992]). Вместо этого мы можем вычислить связанную матрицу  $\Sigma'_{\Phi}$ :

$$\Sigma'_{\Phi} = \Phi\Phi^T \quad (14.6)$$

размеры которой намного меньше ( $m \times m$ ). Собственные вектора и собственные значения матрицы  $\Sigma'_{\Phi}$  связаны с собственными векторами и значениями матрицы  $\Sigma_{\Phi}$  следующим образом:

$$\Sigma_{\Phi}F = \lambda F \quad (14.7)$$

$$\Sigma'_{\Phi}F' = \lambda F' \quad (14.8)$$

$$F = \Phi^T F' \quad (14.9)$$

где  $\lambda$  — это вектор собственных значений матрицы  $\Sigma_{\Phi}$ ,  $F$  — вектор собственных векторов матрицы  $\Sigma_{\Phi}$ , а  $F'$  — вектор собственных векторов матрицы  $\Sigma'_{\Phi}$ .

Описанные здесь методы анализа главных компонент позволили получить ряд впечатляющих результатов в области распознавания лиц (см. в списке литературы работы [Kirby, Sirovitch, 1990], [Turk, Pentland, 1991], [Swets, Weng, 1996]). В качестве критического замечания можно отметить, что маловероятно, что этот метод будет работать с изображениями со значительными высокочастотными изменениями, так как автокорреляция будет быстро уменьшаться при малых сдвигах изображения, таким образом влияя на требование явного ограничения объекта. Функции изображения для лиц не решают этой проблемы. В работе [Swets, Weng, 1996] продемонстрировали хорошие результаты для многих (нетекстурированных) объектов, как и в статье [Murase, Nayar, 1995], где практически показана возможность интерполяции положения трехмерного объекта с точностью до двух градусов, при помощи базы данных эталонных изображений, сформированных с шагом 10 градусов.

В работе [Turk, Pentland, 1991] предложено решение двух из перечисленных выше проблем. Во-первых, авторы этой работы использовали методы анализа движения (см. гл. 9) для сегментации для выделения образа головы на видеопоследовательности — это дало им возможность ограничить область лица на изображении и также нормировать размеры изображения. Во-вторых, они произвели взвешивание пикселей изображения путем фильтрации широким гауссовым фильтром, что позволило уменьшить периферийные пиксели примерно до 0, но при этом сохранить важные значения интенсивности в окрестности центра лица.

---

### Упражнение 14.12

Получите набор из 10 изображений лиц и 10 пейзажей с одинаковыми размерами  $R \times C$ . Вычислите евклидово расстояние между всеми парами изображений и отобразите эти расстояния в виде верхней треугольной матрицы размера

$20 \times 20$ . Достаточно ли изображения лиц близки для выделения кластера? А пейзажи? Чему равно отношение наименьшего и наибольшего расстояний? Является ли евклидово расстояние перспективным способом для поиска в базе данных изображений? Объясните, почему.

---

### Упражнение 14.13

Пусть  $I_u$  — это изображение неизвестного объекта и пусть  $\mathbf{V} = \{ \langle I_j, L_j \rangle \}$  — множество маркированных эталонных изображений. Предположите, что все изображения нормированы, так что  $\|I_j\| = 1$ . (а) Покажите, что  $\|I_u - I_j\|$  минимально при максимуме  $I_u \circ I_j$ . (б) Объясните, почему это неверно без предположения о единичных размерах изображения.

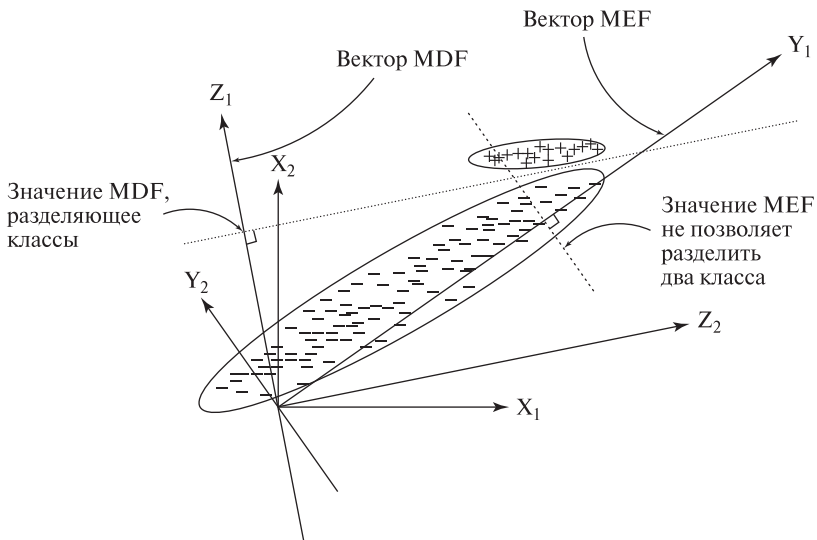
**Улучшение различения и ускорение поиска в памяти** Методы анализа главных компонент позволяют представить подпространство эталонных образцов в компактной форме. Вычисленный по алгоритму 14.4 базис для наилучшего представления эталонных данных был назван множеством *наиболее выразительных признаков* (most expressive features, MEFs). В работе [John Weng и др., 1996] было показано, что, хотя наиболее выразительные признаки оптимальным образом представляют подпространство эталонных изображений, но они не позволяют хорошо представить *различия между изображениями* различных классов. В этой работе было предложено использовать *наиболее различимые признаки* (most discriminating features, MDFs), которые можно получить методами дискриминантного анализа. Признаки MDF сосредотачиваются на вариациях изображения, которые могут служить для различения объектов различных классов. Разница между MEF и MDF иллюстрируется на рис. 14.40. Элемент исходных данных имеет координаты  $(x_1, x_2)$ . Направление  $y_1$  является направлением максимальной дисперсии, а  $y_2$  ортогонально  $y_1$ . Следовательно, координаты  $y_1, y_2$  являются признаками MEF. Исходные классы векторов представлены в виде эллипсов, полуоси которых параллельны  $y_1, y_2$ . (Напомним, что алгоритм для нахождения этих осей в двумерном случае первый раз был представлен в гл. 3.) Пороговые значения, выбранные по осям  $y_1$  или  $y_2$ , не позволяют легко отличить два класса друг от друга. Оси MDF,  $z_1$  и  $z_2$ , вычисленные путем дискриминантного анализа, позволяют лучше разделить эталонные образцы на основе порогового значения по оси  $z_1$ .

Второе усовершенствование, которое в работе [John Weng и др., 1996] выполнено в рамках подхода «распознавание по внешнему виду с использованием пространства собственных векторов», заключается в том, что авторы предусмотрели процедуру построения дерева поиска. Для базы данных из  $S$  эталонных образцов это дерево обеспечивает время поиска ближайших соседей порядка  $O(\log_2 S)$ . Напомним, что деревья решений для классификации объектов рассматривались в гл. 4. В каждой точке решения дерева, неизвестное изображение проектируется в подпространство наиболее различимых

признаков. Это необходимо для принятия решения, какую ветвь или ветви следует обрабатывать далее. Признаки MDF, используемые в различных узлах дерева решений, варьируются в зависимости от того, для каких эталонных образцов они были получены и настраиваются с учетом конкретных необходимых разбиений решений. Заинтересовавшийся читатель может найти в списке литературы ссылки на работы с более детальными сведениями по этой недавно разработанной теории.

### Упражнение 14.14

(а) Получите набор из 300 изображений человеческих лиц и рассматривайте каждое из них как вектор из  $R \times C$  координат. (б) По этим 300 образцам вычислите разреженную матрицу и среднее изображение. (в) По результатам (б) вычислите  $m$  наибольших собственных значений разреженной матрицы и соответствующие  $m$  собственных векторов таких, чтобы они представляли 95% энергии разреженной матрицы. (г) Случайным образом выберите 5 исходных изображений лиц. Представьте каждое из них в виде линейной комбинации из  $m$  наилучших векторов из (в). (д) Выведите на экран каждое из 5 аппроксимирующих изображений, полученных в (г), и сравните их с исходными изображениями.



**Рис. 14.40.** Наиболее выразительные признаки, определенные собственными векторами разреженной матрицы, хорошо представляют данные, но не всегда также хорошо представляют различия между классами. Для поиска подпространств, выявляющих различия между классами, могут применяться методы дискриминантного анализа. (С любезного разрешения J. Swets, J. Weng.)

## 14.5 ЛИТЕРАТУРА

Каркасные модели впервые появились в машинной графике, где они обычно называются полигональными каркасными моделями. Хорошим справочником по данному вопросу является учебник по машинной графике [Foley и др., 1996]. Модели вида «поверхность-ребро-вершина» были введены в системе VISIONS в Массачусетском Университете в 1970-х гг. Структура, рассмотренная в данной главе, заимствована из более поздней работы [Camps, 1992]. Модели на основе обобщенных цилиндров были использованы для работы с дальнометрическими данными [Nevatia, Binford, 1977]. В более поздних работах [Rom, Medioni, 1993] обсуждается вычисление цилиндров по двумерным данным. Октантные деревья впервые были предложены в работе [Hunter, 1978] и в дальнейшем рассмотрены в статье [Jackins, Tanimoto, 1980]. Они более подробно описаны в книге [Samet, 1990]. Наше описание суперквадрик в основном следует работе [Gupta, Bogoni, Bajcsy, 1989]. Иллюстрации с моделями сердца, наряду с материалом для раздела о деформационных моделях, взяты из недавней работы [Park, Metaxas, Axel, 1996].

Введение понятия класса видимости обычно приписывается авторам статьи [Koenderink, Van Doorn, 1979]. В работах [Camps и др., 1992], [Pulli, 1996] и [Costa, 1995] модели на основе классов видимости использовались для распознавания трехмерных объектов. Распознавание на основе совмещения было в основе [Lowe, 1987] и всесторонне проанализировано в статье [Huttenlocher, Ullmann, 1990]. В данной главе рассмотрение совмещения типа 3D-3D следует работе [Johnson, Hebert, 1998], а типа 2D-3D — работе [Pulli, Shapiro, 1996]. Обсуждение распознавания совмещением для объектов с гладкими поверхностями было взято из работы [Chen, Stockman, 1996], и оно связано с оригинальной работой [Basri, Ullman, 1988]. Распознавание по моделям «стержни-пластины-капли» было описано в работе [Shapiro и др., 1984]. Распознавание путем сопоставления с реляционными моделями в общем было рассмотрено в работах [Shapiro, Haralick, 1981, 1985]. Реляционная индексация описана в работе [Costa, Shapiro, 1995]. Наше рассмотрение функционального распознавания объектов следует работе [Stark, Bowyer, 1996].

В статье [Kirby, Sirovich, 1990] был предложен подход к решению задачи сжатия изображений лиц, который в работе [Turk, Pentland, 1991] адаптирован для более эффективного распознавания лиц. Авторы работы [Swets, Weng, 1996] разработали обучающуюся систему SHOSLIF, которая была улучшена на основе метода главных компонент путем использования признаков MDF и конструирования древовидной базы данных для ускорения поиска ближайших соседей за время  $\log_2 N$ . В статье [Murase, Nayar, 1994] также предложен эффективный метод поиска и показано, что положение и ориентацию трехмерного объекта можно оценить с точностью до  $2^\circ$  путем интерполяции эталонных видов, полученных с интервалом  $10^\circ$ . Более того, при работе с несколькими объектами (не лицами), было обнаружено, что для хорошей производительности достаточно использовать пространство собственных векторов размерности

20 или менее. Обзор методов распознавания по внешнему виду в данной главе в основном обязан работе [Swets, Weng, 1996] и часто цитируемой работе [Turk, Pentland, 1991].

Минимизация энергии использовалась в 1970-х гг. для сглаживания контуров. Однако в 1987 г. появилась работа Касса, Виткина и Терзопулоса [Kass, Witkin, Terzopoulos], в которой был введен термин *snake*. Эта работа снова пробудила интерес многих исследователей к данной области. Быстро появились новые алгоритмы для сглаживания и прослеживания поверхностей и объемов. Амини с соавторами [Amini и др., 1988] предложили метод динамического программирования для сглаживания активных контуров на изображениях. Одним из многих примеров его использования является применение для медицинских изображений в работе [Yue и др., 1995]. Работы Чена и Медioni [Chen, Medioni, 1995] и Парк, Метаксаса и Акселя [Park, Metaxas, Axel, 1996] служат хорошими примерами работ в быстро развивающейся области исследований и приложений физических и деформационных моделей.

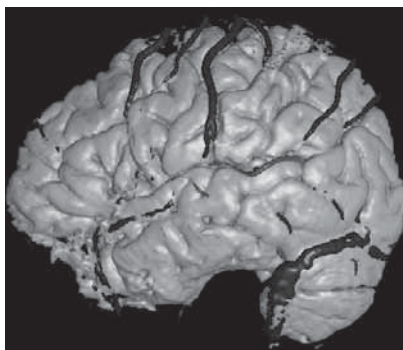
1. Amini, A., S. Tehrani, and T. Weymouth. 1988. Using dynamic programming for minimizing the energy of active contours in the presence of hard constraints. *Proc. IEEE Int. Conf. Comput. Vision*, 95–99.
2. Basri, R., and S. Ullman. 1988. The alignment of objects with smooth surfaces. *Proc. 2nd Intern. Conf. Comput. Vision*, 482–488.
3. Biederman, I. 1985. Human image understanding: recent research and theory. *Comput. Vision, Graphics, and Image Proc.*, v. 32(1):29–73.
4. Camps, O. I., L. G. Shapiro, and R. M. Haralick. 1992. Image prediction for computer vision. In *Three-dimensional Object Recognition Systems*, A. Jain and P. Flynn, eds. Elsevier Science Publishers BV, Amsterdam.
5. Chen, J. L., and G. Stockman. 1996. Determining pose of 3D objects with curved surfaces. *IEEE Trans. Pattern Analysis and Machine Intelligence*, v. 18(1):57–62.
6. Chen, Y., and G. Medioni. 1995. Description of complex objects from multiple range images using an inflating balloon model. *Comput. Vision and Image Understanding*, v. 61(3):325–334.
7. Costa, M. S., and L. G. Shapiro. 1995. Scene analysis using appearance-based models and relational indexing. *IEEE Symp. Comput. Vision* (Nov. 1995), 103–108.
8. Foley, J., A. van Dam, S. Feiner, and J. Hughes. 1996. *Computer Graphics: Principles and Practice*. Addison-Wesley, Reading, MA.
9. Gupta, A., L. Bogoni, and R. Bajcsy. 1989. Quantitative and qualitative measures for the evaluation of the superquadric model. *Proc. IEEE Workshop on Interpretation of 3D Scenes*, 162–169.
10. Hunter, G. M. 1978. *Efficient Computation and Data Structures for Graphics*. Ph.D. Dissertation, Princeton University, Princeton, NJ.

11. Huttenlocher, D. P., and S. Ullman. 1990. Recognizing solid objects by alignment with an image. *Int. J. Comput. Vision*, v. 5(2):195–212.
12. Jackins, C. L., and S. L. Tanimoto. 1980. Oct-trees and their use in representing three-dimensional objects. *Comput. Graphics and Image Proc.*, v. 14:249–270.
13. Johnson, A. E., and M. Hebert. 1998. Efficient multiple model recognition in cluttered 3-D scenes. *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, 671–677.
14. Kass, M., A. Witkin, and D. Terzopoulos. 1987. Snakes: active contour models. *Proc. First Int. Conf. Comput. Vision*, London, UK, 259–269.
15. Kirby, M., and L. Sirovich. 1990. Application of the Karhunen-Loeve procedure for the characterization of human faces. *IEEE Trans. Pattern Anal. and Machine Intelligence*, v. 12(1):103–108.
16. Koenderink, J. J., and A. J. van Doorn. 1979. The internal representation of solid shape with respect to vision. *Biological Cybernetics*, v. 32:211–216.
17. Lowe, D. G. 1987. The viewpoint consistency constraint. *Int. J. Comput. Vision*, v. 1:57–72.
18. Murase, H., and S. Nayar. 1995. Parametric appearance representation. In *3D Object Representations in Computer Vision*, J. Ponce and M. Herbert, eds. Springer-Verlag.
19. Nevatia, R., and T. O. Binford. 1977. Description and recognition of curved objects. *Artificial Intelligence*, v. 8:77–98.
20. Park, J., D. Metaxas, and L. Axel. 1996. Analysis of left ventricular wall motion based on volumetric deformable models and MRI-SPAMM. *Medical Image Anal. J.*, v. 1(1):53–71.
21. Pentland, N. P. 1986. Perceptual organization and the representation of natural form. *Artificial Intelligence*, v. 28:29–73.
22. Pulli, K., and L. G. Shapiro. 1996. Triplet-based object recognition using synthetic and real probability models. *Proc. ICPR96*, v. IV:75–79.
23. Roberts, L. G. 1977. Machine perception of three-dimensional solids. In *Computer Methods in Image Analysis*, J. K. Aggarwal, R. O. Duda, and A. Rosenfeld, eds. IEEE Computer Society Press, Los Alamitos, CA.
24. Rom, H., and G. Medioni. 1993. Hierarchical decomposition and axial shape description. *IEEE Trans. Pattern Anal. and Machine Intelligence*, v. 15(10):973–981.
25. Samet, H. 1990. *Design and Analysis of Spatial Data Structures*. Addison-Wesley, Reading, MA.
26. Shapiro, L. G., J. D. Moriarty, R. M. Haralick, and P. G. Mulgaonkar. 1984. Matching three-dimensional objects using a relational paradigm. *Pattern Recog.*, v. 17(4):385–405.

27. Shapiro, L. G., and R. M. Haralick. 1981. Structural descriptions and inexact matching. *IEEE Trans. Pattern Anal. and Machine Intelligence*, v. PAMI-3(5):504–519.
28. Shapiro, L. G., and R. M. Haralick. 1985. A metric for comparing relational descriptions. *IEEE Trans. Pattern Anal. and Machine Intelligence*, v. PAMI-7(1):90–94.
29. Stark, L., and K. Bowyer. 1996. *Generic Object Recognition Using Form and Function*. World Scientific Publishing Co. Pte. Ltd., Singapore.
30. Swets, D., and J. Weng. 1996. Using discriminant eigenfeatures for image retrieval. *IEEE Trans. Pattern Anal. and Machine Intelligence*, v. 18:831–836.
31. Turk, M., and A. Pentland. 1991. Eigenfaces for recognition. *J. Cognitive Neuroscience*, v. 3(1):71–86.
32. Yue, Z., A. Goshtasby, and L. Ackerman. 1995. Automatic detection of rib borders in chest radiographs. *IEEE Trans. Medical Imaging*, v. 14(3):525, 536.

# Системы виртуальной реальности

Предположим, что хирургу необходимо разработать план операции по удалению опухоли головного мозга. В процессе диагностики были получены трехмерные изображения черепа и мозга пациента. С помощью технологий *виртуальной реальности* хирург может попробовать выполнить операцию сначала на трехмерной модели, а не на реальном пациенте. При этом хирург может проверить различные варианты разрезов и хирургических приемов, чтобы в результате выбрать наилучший способ оперирования реального пациента. Кроме того, можно совместить изображения из атласа головного мозга и трехмерные изображения мозга пациента так, чтобы хирург видел их в режиме оверлейного наложения и ему было легче определять последствия различных хирургических манипуляций и избегать повреждения важных структур мозга. Для поддержки подобных виртуальных операций были разработаны системы виртуальной реальности. На рис. 15.1 показано компьютерное изображение модели мозга, которое могло бы использоваться в системе виртуальной реальности.



**Рис. 15.1.** Искусственно сформированное изображение трехмерной модели, построенной на основе реальных данных ЯМР-томографии. (С любезного разрешения Вашингтонского университета. Получено в рамках программы исследований человеческого мозга.)



Системы виртуальной реальности составляют новую область исследований, хотя она часто рассматривается как один из разделов компьютерной графики. Это объясняется тем, что в системах виртуальной реальности важное место занимают компоненты для формирования компьютерных изображений. Тем не менее приложения виртуальной реальности представляют самостоятельную ценность для изучения. Системы виртуальной реальности имеют отношение к материалу данной книги по нескольким причинам. (а) В этих системах часто необходимы реальные изображения и методы обработки изображений. (б) Для *погружения* пользователя-человека в виртуальную среду требуется генерировать качественные стереоскопические изображения. (в) Для сопоставления трехмерных точек моделей и реального мира применяются такие же математические модели, как и в компьютерном зрении. (г) Компьютерное зрение иногда используется для определения положения пользователя или других реальных объектов.

Системы виртуальной реальности появились благодаря работам в различных прикладных областях, например, это разработка тренажеров (особенно авиационных), дистанционно управляемых устройств и компьютерных игр.

## 15.1 ОСНОВНЫЕ ПРИЗНАКИ СИСТЕМ ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ

Перечислим основные признаки, присущие системам виртуальной реальности и виртуальным средам, после чего рассмотрим несколько интересных приложений.

- Пользователь-человек воздействует на **модель** реальности. При этом обеспечивается имитация операций, возможных с реальными объектами.
- Применяются методы высококачественного отображения информации с достаточным разрешением и скоростью, чтобы пользователь был *погружен* в данные и у него возникали ощущения, похожие на ощущения восприятия реального мира.
- Пользователь должен быть способен взаимодействовать с модельной средой и влиять на нее в режиме реального времени.
- Первостепенное значение имеет трехмерная зрительная обратная связь. В системе виртуальной реальности пользователь обычно может менять точку наблюдения или поворачивать и перемещать объекты для удобства наблюдения. Хотя зрительная обратная связь наиболее важна, также должна присутствовать тактильная, двигательная, силовая и звуковая обратная связь. Например, чтобы пользователь мог почувствовать прикосновение к объекту, услышать соударение объектов и т. п.

На рис. 15.2 иллюстрируется традиционный сценарий действий человека в реальной среде, а на рис. 15.3 — новый сценарий действий в виртуальной среде.

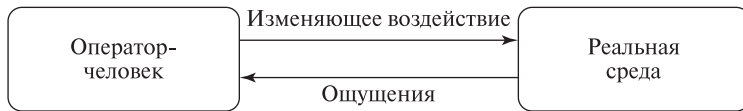


Рис. 15.2. Действия оператора-человека в реальной (естественной) среде.

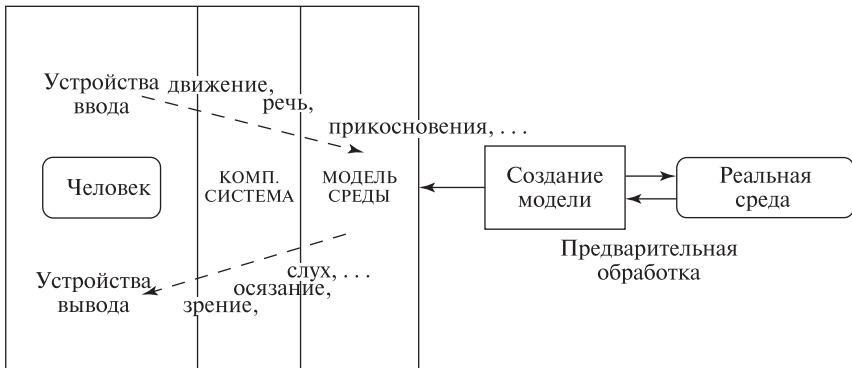


Рис. 15.3. Оператор-человек, погруженный в виртуальную среду.

### Упражнение 15.1. Книги и кинофильмы

- Люди, несомненно, способны погружаться в воображаемую среду во время чтения книги. Какие из вышеперечисленных четырех признаков систем виртуальной реальности наблюдаются при этом, а какие нет? (Отметим, что существуют книги, которые позволяют читателю выбрать один из нескольких вариантов развития сюжета.)
- Люди также могут погружаться в воображаемую среду во время просмотра кинофильмов, особенно с применением технологий широкоэкранный или трехмерного кино. Какие из вышеперечисленных четырех признаков систем виртуальной реальности присутствуют в данном случае, а какие нет?
- Знаете ли вы или, может быть, играли сами в видеоигры, которые обладают всеми четырьмя признаками, перечисленными выше?

## 15.2 ПРИЛОЖЕНИЯ СИСТЕМ ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ

Системы виртуальной реальности обычно ассоциируются с экзотическими новыми приложениями, для которых требуется дорогое современное оборудование. Однако прообразы систем виртуальной реальности были разработаны и на некоторых известных старых компьютерных системах.

**Осмотр архитектурных сооружений.** Пользователь может взаимодействовать с архитектурной моделью дома в процессе виртуальной прогулки. Возможно, обеспечивается даже вид на виртуальный пейзаж из виртуальных

окон. Например, при осмотре усадьбы Монтичелло, принадлежавшей президенту Томасу Джефферсону, пользователь мог бы увидеть его коллекцию редкостей, уникальную кровать и часы. В настоящее время выполняется формирование цифровых представлений многих исторических памятников и *виртуальных музеев*. В простейшем случае пользователи могут исследовать эти виртуальные среды с помощью World Wide Web и обычного плоского дисплея. Пользователям, вероятно, будет запрещено модифицировать модель усадьбы Монтичелло, но они могут брать и рассматривать предметы обстановки. Если же пользователь намеревается построить настоящий дом, то он должен иметь возможность изменения архитектурной модели, возможно, экспериментируя с различными вариантами внутренней отделки, размещением деталей фурнитуры и т. п.

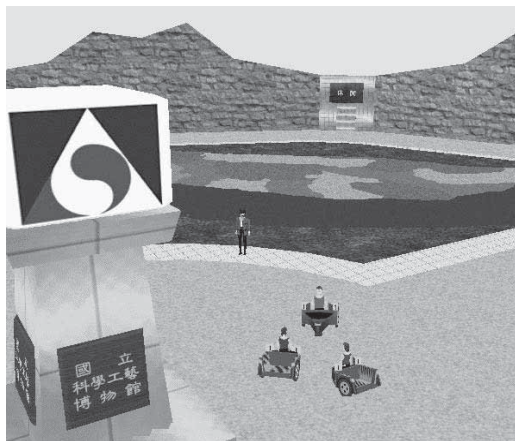
В качестве разновидности *осмотра сцен* может быть предусмотрен их *облет*. Например, пользователь мог бы осматривать Большой Каньон, как будто облетая его сверху. В простейших реализациях облета вид не может выбираться пользователем, который чувствует себя пассажиром самолета. В более сложных реализациях пользователь может чувствовать себя пилотом, который способен изменять курс и рассматривать сцену с различных направлений.

**Моделирование полетов.** С помощью авиационных тренажеров пользователь может попрактиковаться в управлении самолетом, выполняя полеты в различных уголках земного шара и пользуясь разными аэропортами. Эти тренажеры настолько реалистичны, что у людей учащается пульс и дыхание, и это подтверждает глубину погружения в виртуальную среду.

**Интерактивная сегментация анатомических структур.** Системы виртуальной реальности способны помочь медикам в идентификации трехмерных данных измерений и построении по ним анатомических моделей. Другими словами, системы виртуальной реальности могут применяться для обеспечения интерактивной сегментации данных. Например, предположим, что трехмерные данные ЯМР-томографии отображаются оператору в виде стереоизображений. Взаимодействуя с данными, оператор мог бы отметить последовательность точек в качестве центра кровеносного сосуда или границы сердечной камеры. В этой задаче необходимы устройства вывода стереоскопических изображений и трехмерные устройства ввода (см. разд. 15.5).

В настоящее время известно много приложений систем виртуальной реальности, в том числе проектирование, помощь слабовидящим, автомобильные и авиационные тренажеры, визуализация научных данных и виртуальное обучение. На рис. 15.4 приведена иллюстрация по проекту Вашингтонского университета в области систем виртуальной реальности: Динамическая виртуальная игровая площадка (Dynamic Virtual Playground). Виртуальная игровая площадка разрабатывается в качестве прототипа системы для исследования множественных одновременных взаимодействий агентов в виртуальной среде. Например, она могла бы использоваться для моделирования учебной лабо-

**Рис. 15.4.** Динамическая виртуальная игровая площадка — экспериментальная среда, в которой пользователи могут взаимодействовать друг с другом. (С любезного разрешения Лаборатории человеко-машинных интерфейсов Вашингтонского университета.)



ратории, в которой несколько групп студентов работают над различными проектами.

### 15.3 СИСТЕМЫ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ

Рассмотрим задачу реконструкции существующего здания, для решения которой инженеру-строителю необходимо определить расположение электропроводки и водопроводных и газовых труб. Эти данные могут содержаться на бумажных картах, чертежах или даже в файлах САПР. Представленный далее сценарий может служить примером применения системы *дополненной реальности* (иногда они называются системами *смешанной реальности*). Инженер надевает головной дисплей, который нужен для оверлейного наложения компьютерной графики на наблюдаемую реальную сцену. Когда инженер смотрит вниз, то видит синие прямые, отмечающие положение водопроводных труб, а когда смотрит на стену, то видит еще и красные линии, обозначающие электрические провода. В этой задаче, в некотором смысле, дополненная реальность позволяет видеть сквозь стены!

Для разработки систем дополненной реальности требуется обеспечить следующие компоненты и возможности.

- Трехмерные модели объектов, применяемые для дополнения реальных сцен.
- Установление соответствия реального пространства пользователя с данными трехмерных моделей на основе калибровки.
- Слежение за положением пользователя для определения его точки наблюдения в реальном пространстве.
- Отображение в реальном времени, сочетающее изображения реальных сцен и компьютерную графику, сгенерированную на основе моделей.

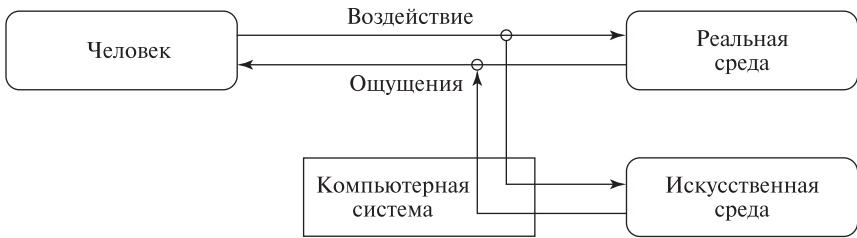


Рис. 15.5. Оператор-человек, действующий в среде дополненной реальности.

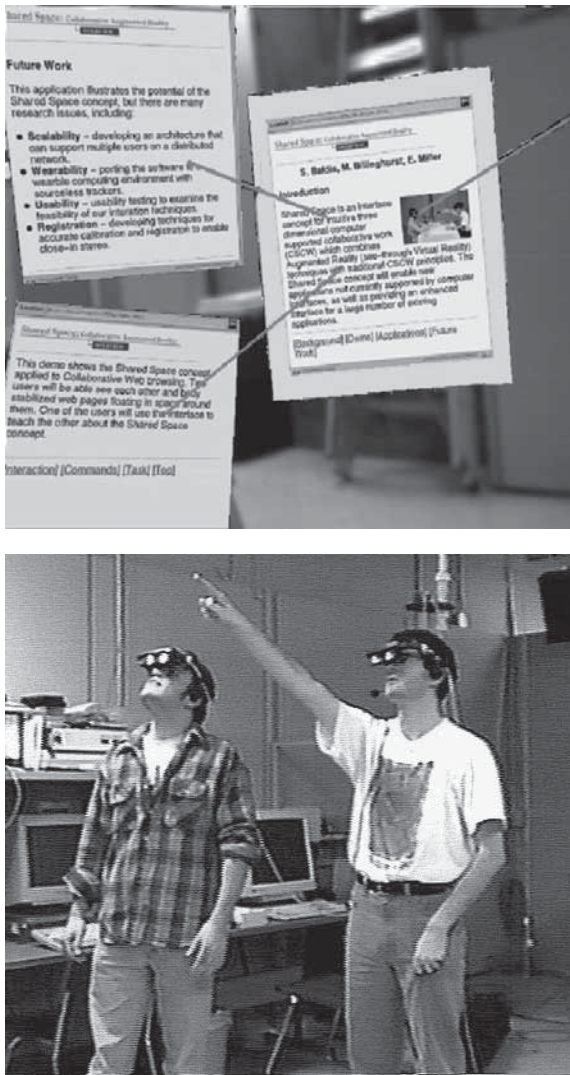
- Малое время отклика на движения головы и точность совмещения изображений и графики. Эти характеристики крайне важны с точки зрения эффективности системы.

Схема среды дополненной реальности приведена на рис. 15.5. Ее можно сравнить с другими схемами данной главы. Существует много приложений дополненной реальности. Далее перечислены некоторые из них.

- Системы дополненной реальности могут применяться в хирургии. Например, хирург, оперирующий реального пациента, может видеть изображение пациента с наложенными на него данными компьютерного томографа и траекторией планируемого разреза. (Траектория разреза может быть получена с помощью системы виртуальной реальности, как описывалось выше.)
- В задаче контроля печатных плат оператор-человек сравнивает новую печатную плату с моделью САПР и проверяет наличие на плате всех необходимых компонент и проводников. Плата точно позиционируется на стенде так, чтобы можно было совместить изображение с камеры и модель



Рис. 15.6. Использование методов дополненной реальности в телеконференциях. (С любезного разрешения Лаборатории человеко-машинных интерфейсов Вашингтонского университета.)



**Рис. 15.7.** Два человека в полупрозрачных очках системы дополненной реальности (внизу). Они видят как реальный мир, так и компьютерные изображения. Оба человека смотрят на Web-страницы (вверху). Эти страницы кажутся людям висящими в воздухе. (С любезного разрешения Лаборатории человеко-машинных интерфейсов Вашингтонского университета.)

САПР. Совмещенное изображение демонстрируется оператору на большом компьютерном дисплее.

- В автомобиле установлен дисплей, на котором компьютерная система показывает расположенные вдоль маршрута географические ориентиры и

достопримечательности. На приборной панели расположены проекторы, которые проектируют названия зданий и улиц на лобовое стекло.

- Группа разработчиков назначает встречу для обсуждения совместно разрабатываемой компьютерной модели. Им требуется посмотреть на модель, выбрать и обсудить ее особенности. При этом разработчики хотят видеть друг друга и окружающую среду. Этот сценарий может быть расширен на случай телеконференции, в процессе которой собеседники могут находиться в географически удаленных местах. При этом они могут захотеть увидеть не только общую компьютерную модель, но и других участников.

Применение системы дополненной реальности для телеконференций иллюстрируется на рис. 15.6. Пользователь этой системы помещает на стол две белые карты. На каждой карте есть черная квадратная область, на которой нанесен некоторый образ (например, напечатано слово). Для обнаружения карт применяются методы компьютерного зрения, а для распознавания образов на них — методы статистического распознавания образов. В результате система определяет назначение каждой карты. С помощью методов дополненной реальности на одну из этих карт помещается изображение удаленного собеседника, а на другую — изображение обсуждаемой графической модели.

На рис. 15.7 представлен случай, когда два человека в одной комнате с помощью полупрозрачных очков видят набор совместно обсуждаемых Web-страниц.

## 15.4 ДИСТАНЦИОННОЕ УПРАВЛЕНИЕ

*Дистанционное управление*, или *телеуправление*, является установившейся инженерной дисциплиной, существенно повлиявшей на системы виртуальной реальности. В частности, из систем дистанционного управления были заимствованы датчики и эффекторы (датчики, воспринимающие активные воздействия окружающей среды), необходимые для организации взаимодействия человека со средой. Схему на рис. 15.8 можно сравнить с другими схемами в этой главе на предмет поиска похожих и различающихся признаков. Оператор-человек с помощью дистанционного управления способен выполнять реальные действия в реальной среде, которая может быть расположена удаленно от оператора. Например, по командам оператора действия в реальной среде могут выполнять различные робототехнические системы. Приведем несколько примеров успешного использования методов дистанционного управления.

1. Управление подвижным роботом Pathfinder. Оператор, находящийся на Земле, с помощью компьютера отдает команду управления роботу, который передвигается по поверхности Марса, чтобы робот переместился на 10 см и взял пробу грунта. Оператор получает обратную связь посредством изображений от камер, установленных на посадочном модуле, который доставил робот на Марс и находится недалеко от робота. Пере-



дача изображений и команд по радиоканалу занимает примерно 11 минут из-за большого расстояния между Марсом и Землей.

2. Управление роботом для опасных работ. Посредством радиосвязи оператор-человек управляет удаленным роботом, который пылесосом очищает радиоактивное загрязнение после незначительной аварии на атомной электростанции. С помощью головного дисплея человек видит зараженную область так, как ее видят камеры робота. Для управления пылесосом робота применяется специальная ручка управления. На ней установлены датчики положения и движения. По данным от этих датчиков формируются управляющие команды, которые передаются роботу и используются для управления установленным на нем пылесосом.
3. Применение в медицине. Для удаленного выполнения операции хирург имитирует наложение шва, зашивая искусственный объект, похожий на футбольный мяч. Датчики записывают параметры движений хирурга и пересылают их удаленному роботу, который зашивает разрез на реальной живой собаке. Этот действительно выполненный эксперимент может рассматриваться в качестве первого шага к разработке способов выполнения хирургических операций в географически труднодоступных местах.
4. Сценарий из будущего. Предположим, что пациент направлен в больницу для очистки артерий от атеросклеротических бляшек. Пациента помещают в ЯМР-томограф, который позволяет наблюдать тело пациента в трехмерном виде в реальном времени. Микророботы вводятся в поток крови и рассредоточиваются по телу. Врач-оператор, с учетом данных предыдущих диагнозов, с помощью трехмерного устройства ввода обозначает на стенках сосудов области для очистки. ЯМР-томограф работает в двух режимах. В одном режиме выполняется просмотр изображений, как обычно, а в другом производится формирование заданий микророботам для очистки заданных областей стенок сосудов.

На рис. 15.9 показан пример реально работающей робототехнической системы с дистанционным управлением, разработанной в компании Kyushu Electric

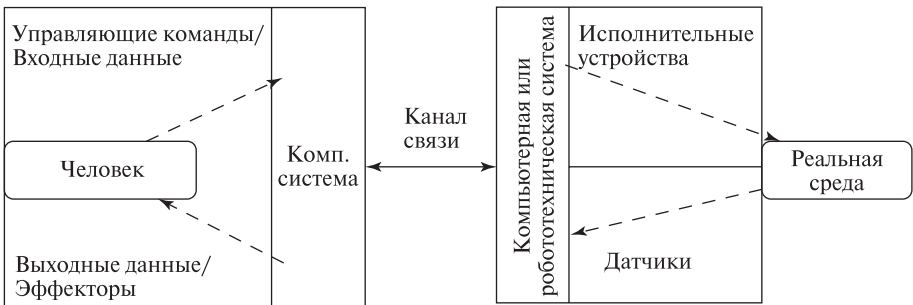


Рис. 15.8. Оператор-человек, использующий средства дистанционного управления.





**Рис. 15.9.** Система для ремонта линий электропередачи *Hot-Line Telerobot System* компании Kyushu Electric Power. (Слева) дистанционно управляемый робот ремонтирует высоковольтную линию электропередачи. (Справа) интерфейс оператора системы. (С любезного разрешения Blake Hannaford и MIT Press. Воспроизведено по изданию К. Goldberg, *The Robot in the Garden*, Cambridge, MA: The MIT Press, 2000.)

Power. Эти роботы используются для безопасного ремонта высоковольтных линий электропередачи.

Перед тем как перейти к обсуждению устройств и математических моделей, необходимых для реализации технологий виртуальных сред, приведем показательный анекдотический случай. Дистанционно управляемый экскаватор был разработан для того, чтобы удаленный оператор мог перемещать землю, уголь и другие материалы. Оператор должен был надевать *информационную перчатку*, которая воспринимает положение руки и пальцев, а компьютерная система преобразует их в сигналы управления экскаватором. На экскаваторе установлены две камеры, дающие левое и правое изображение для головного дисплея оператора. Оператор рукой в перчатке может захватить грудку опилок на столе, а удаленный экскаватор подражает этим движениям и поднимает грудку угля. Легко представить, что у оператора зачесался нос и он непроизвольно поднес руку в перчатке к носу. Перемещение руки в информационной перчатке к носу заставило ковш экскаватора двигаться к камерам в реальной среде. После передачи соответствующих изображений оператору ему показалось, что к лицу приближается массивный ковш! Это был воображаемый удар, и оператор физически не был поврежден. Однако он был обескуражен настолько,

что потребовался перерыв в работе. (Если бы система не была тщательно разработана, то в данной ситуации могло бы произойти повреждение камер на экскаваторе.)

- 110 Определение.** Средства дистанционного управления (телеуправления, teleoperation) позволяют оператору-человеку управлять реальным устройством в удаленной реальной среде. Обратная связь от удаленной среды и возможности управления устройством обеспечивают некоторую иллюзию присутствия оператора в реальной среде.
- 111 Определение.** Виртуальная реальность (virtual reality) — это искусственная реальность, сформированная компьютерной системой для пользователя-человека на основе подробных моделей реального мира с помощью специальных устройств ввода/вывода. У оператора-человека создается иллюзия работы с реальными объектами, которые в действительности не существуют. Иллюзорная среда, которую воспринимает и изменяет оператор, называется виртуальной средой (virtual environment, VE).
- 112 Определение.** Дополненная реальность (augmented reality) или смешанная реальность (mixed reality) формируется посредством комбинирования реальной и виртуальной среды. Искусственные данные, сгенерированные компьютерной системой, комбинируются с воспринимаемыми данными из реальной среды для дополнения у пользователя ощущений восприятия реальности.
- 113 Определение.** Искусственная среда (synthetic environment, SE) — это среда, которая предоставляется человеку-оператору с помощью компьютерной системы и специальных устройств ввода вывода с использованием средств дистанционного управления, дополненной и виртуальной реальности. (Иногда термин «виртуальная среда» используется для обозначения всех трех перечисленных вариантов.)

## 15.5 УСТРОЙСТВА ДЛЯ СИСТЕМ ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ

Для организации взаимодействия оператора-человека с искусственной средой часто используются несколько устройств. На рис. 15.10 представлена иллюстрация для пояснения дальнейшего материала. Этот рисунок относится к описанной выше задаче реконструкции здания. Используемые в этой задаче устройства применяются в приложениях искусственных сред всех трех типов. Инженер обходит перестраиваемое здание и отмечает на стенах расположение электрических проводов и водопроводных труб. Инженер видит реальные стены сквозь очки головного дисплея, который обеспечивает оверлейное наложение компьютерных изображений труб и проводов.

В этой задаче оператор (а) взаимодействует с реальной средой (b), дополненной с помощью компьютерной системы (с). Эта система с помощью

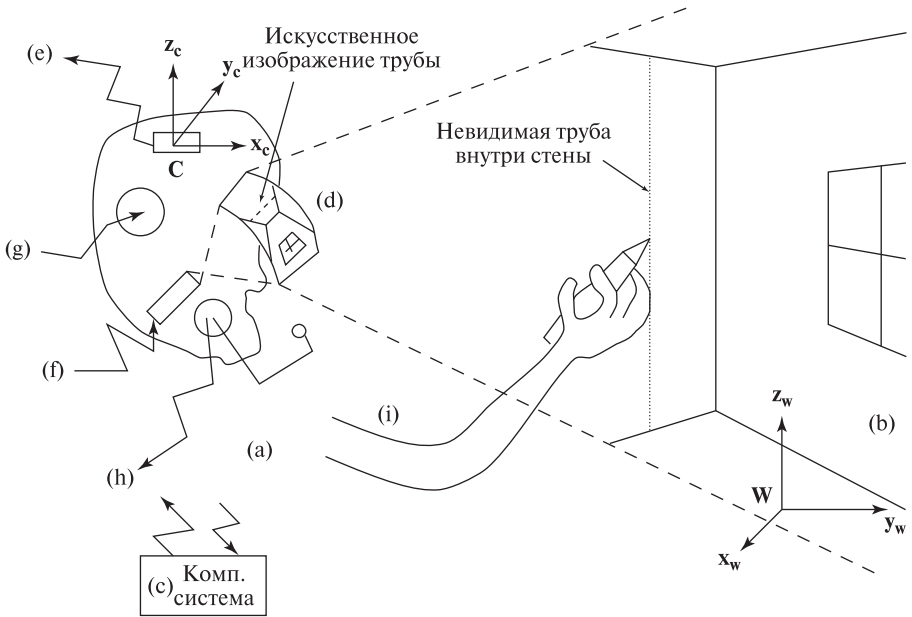
устройств (d, e, f) добавляет к наблюдаемым сценам обычно невидимые детали и обеспечивает голосовой интерфейс (g, h) с оператором, который может свободно перемещаться по зданию и наносить обозначения на стены (i). Оператор смотрит на реальные сцены сквозь полупрозрачное зеркало (d), установленное в головном дисплее. Датчик положения (e) передает ориентацию и положение головы оператора в компьютер, который использует эти параметры для генерации компьютерного изображения труб и проводов на основе трехмерной модели САПР. Для дополнения реальной сцены сгенерированное изображение проектируется на внутреннюю сторону полупрозрачного зеркала, сквозь которое оператор смотрит (f) на реальные объекты.

**Головной дисплей (head-mounted display, HMD).** Схема *полупрозрачного* головного дисплея приведена слева на рис. 15.11. Полупрозрачное одностороннее зеркало пропускает свет, поступающий снаружи из реального мира, и отражает к оператору свет от компьютерного дисплея. В результате оператор видит дополненное изображение. Миниатюрный графический дисплей вместе с оптической системой расположен внутри головного дисплея или шлема. Схема другого варианта устройства, непрозрачного головного дисплея, показана на рис. 15.11, справа. Отметим, что все составные элементы установлены внутри головного дисплея и перемещаются вместе с ним — зеркало, камера и графический дисплей. В этом устройстве искусственное изображение представляет собой комбинацию цифрового изображения, полученного с помощью камеры, и изображения, сгенерированного компьютером. Полупрозрачные головные дисплеи обеспечивают более высокое разрешение изображений, а непрозрачные предоставляют больше возможностей контроля наблюдаемых пользователем сцен.

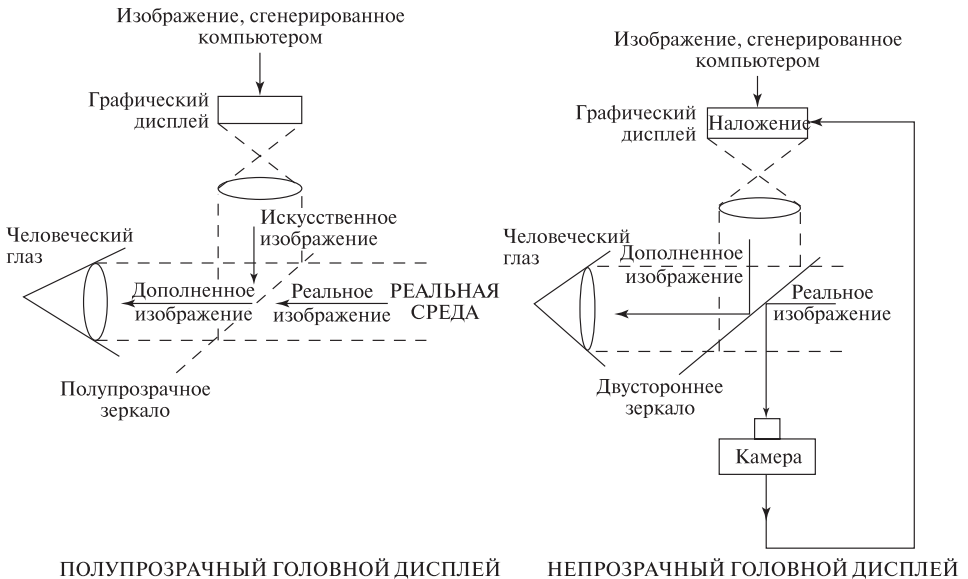
---

### Упражнение 15.2. Точность совмещения изображений в системах дополненной реальности

Рассмотрим рис. 15.10. (а) Допустим, что ошибка определения угла поворота головы наблюдателя с помощью датчика положения составляет два градуса. Какое при этом в реальном пространстве будет расхождение (в см) между местом, где оператор отметил вертикально проходящую трубу, и ее истинным местоположением? (б) Предположим, что угол поля зрения головного дисплея равен 120 градусов, а в строке компьютерного изображения содержится 500 пикселей. (При необходимости можете считать фокусное расстояние системы формирования изображений равным 2.0 см. Также можете полагать, что стена расположена в 3 м от оператора.) Чему будет равно (в пикселях) горизонтальное расстояние между проекцией трубы на дополненном изображении и ее истинным положением на изображении?



**Рис. 15.10.** Аппаратные устройства, используемые оператором-человеком в системе дополненной реальности. Оператор отмечает расположение труб в стенах здания. Местоположение труб описано в модели дома, разработанной в САПР.



**Рис. 15.11.** Устройство головных дисплеев: (слева) полупрозрачного и (справа) непрозрачного.

---

**Упражнение 15.3.** Точность совмещения изображений в системах дополненной реальности

Это упражнение по смыслу связано с предыдущим. Рассмотрим задачу контроля автомобильных приборных панелей, в которой оператор-человек с помощью системы дополненной реальности сравнивает изготовленную панель с моделью, построенной в САПР. В отличие от предыдущего упражнения, на реальном изображении оператор должен видеть все признаки модели САПР. Оператор проверяет наличие и работоспособность всех приборов, в том числе счетчика пройденного пути, манометра, радиоприемника и т. п. Компьютерная система дополненной реальности также управляет тестирующим оборудованием и подает оператору сигналы, обращающие его внимание на обнаруженные дефекты. Будем считать, что поле зрения имеет угол 60 градусов. Панель расположена примерно в 60 см от оператора. Как и в предыдущем упражнении, предположим, что датчик положения при определении поворота головы оператора допускает ошибку в два градуса.

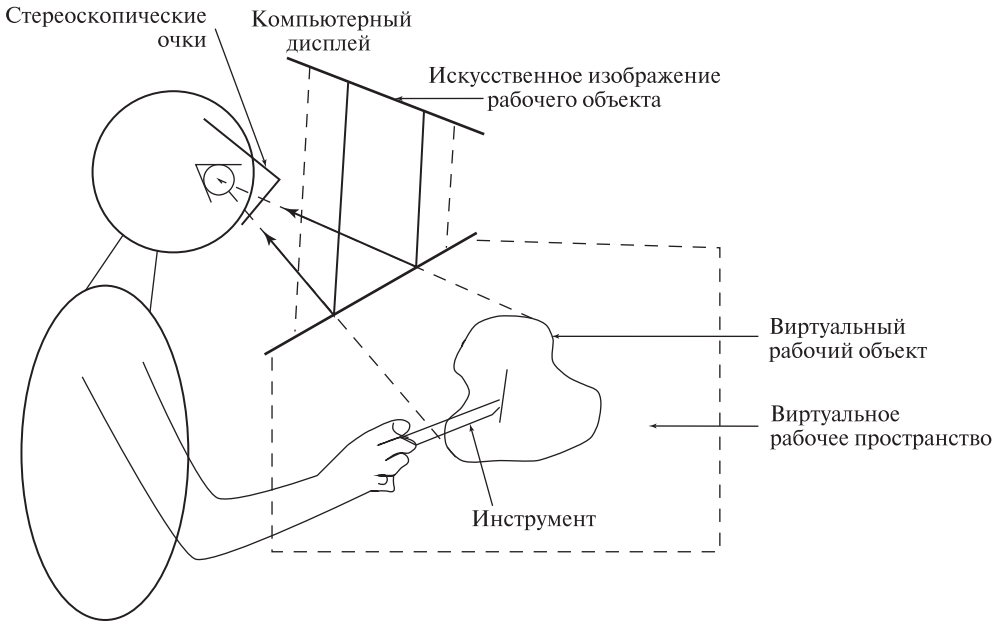
(а) Если на дополненном изображении содержится красная окружность вокруг кнопки радиоприемника, то какая горизонтальная ошибка совмещения будет соответствовать ошибке определения угла поворота? (б) Полагая, что ошибка рассогласования недопустимо велика, предложите метод для ее автоматического уменьшения с помощью компьютерной системы. Допускает ли ваш метод использование головных дисплеев обоих типов? (в) Считая, что оператор мог бы лучше выполнять контроль, если бы обнаруживал небольшие рассогласования реальных и искусственных объектов, предложите метод автоматического внесения небольших рассогласований с помощью компьютерной системы. Допускает ли ваш метод использование головных дисплеев обоих типов? (г) Необходима ли в данной задаче система дополненной реальности или задача может быть решена с помощью полностью автоматической системы? Объясните свой ответ.

---

**Упражнение 15.4.** Система дополненной реальности с несколькими операторами

Предположим, что хирургическая операция выполняется группой из нескольких хирургов. Можно ли обеспечить, чтобы каждый из них, надев головной дисплей, видел схему операции и анатомические структуры, наложенные на сцену с реальным пациентом? Поясните, как можно было бы это сделать или почему это невозможно.

**Виртуальное выполнение сложных действий.** Схема системы виртуальной реальности для выполнения сложных ручных операций на моделях объектов представлена на рис. 15.12. Оператор-человек видит искусственное рабочее пространство на стереоскопических изображениях, проектируемых с расположенного сверху дисплея на зеркало. При этом оператор может



**Рис. 15.12.** Схема рабочего места для виртуального выполнения сложных ручных операций.

выполнять действия руками в реальном рабочем пространстве, расположенном ниже зеркала. Оператор манипулирует инструментом в пространстве под зеркалом. Положение инструмента тщательно отслеживается. Его изображение проектируется на зеркало для обеспечения зрительной обратной связи.

На рис. 15.12 показано, как оператор тренируется выполнять надрез на виртуальном органе. Различные технологии для реализации трехмерных инструментов описаны ниже. Очевидно, что точность и скорость определения положения инструмента и частота обновления изображений являются критически важными характеристиками данной системы. Кроме очевидных приложений обучения или планирования хирургических операций, подобные системы могут использоваться дизайнерами и скульпторами для «цифровой лепки» трехмерных моделей.

**Устройства вывода стереоскопических изображений.** Стереоскопическое зрение, пожалуй, обеспечивает наиболее важные зрительные признаки для восприятия трехмерных объектов на расстояниях в пределах 10 м и является доминирующим способом организации обратной связи в системах виртуальной реальности. Стереоскопические дисплеи обычно разрабатываются на основе двух известных схем. В полупрозрачных головных дисплеях каждому глазу оператора можно демонстрировать отдельные изображения. С помощью математических методов, описанных в гл. 12, на основе модели

объекта генерируются левое и правое изображения с соответствующим рассогласованием. Полупрозрачные головные дисплеи обеспечивают широкоугольное поле зрения бесконечно протяженного виртуального мира и поэтому дают реалистичные ощущения восприятия виртуального мира. Однако ряд ограничений затрудняют разработку устройств, которые будут удобно подходить различным пользователям. Отметим, что в одну виртуальную среду могут быть погружены несколько пользователей. У каждого из них есть собственный головной дисплей с датчиком положения, который обеспечивает данные для генерации индивидуального вида виртуальной среды, соответствующего выбранной пользователем точке наблюдения.

Альтернативный подход предполагает использование обычного графического дисплея (рис. 15.12). На этом дисплее по очереди выводятся левое и правое изображения. Оператор пользуется очками со шторками, которые закрываются и открываются синхронно с кадровой частотой обновления дисплея, так что левый глаз видит только нечетные кадры, а правый — только четные (или наоборот). Зрительная система человека сочетает эти чередующиеся изображения таким образом, что человек видит трехмерную сцену. Описанная схема позволяет разрабатывать недорогие и простые в использовании системы. Однако они обеспечивают ограниченную степень погружения, так как все изображения ограничены областью дисплея (по очевидным причинам он называется *аквариумом виртуальной реальности*, *fishtank virtual reality*): вид виртуального мира ограничен за счет отсечения по границам экрана дисплея (*аквариума*). Поскольку имеется только один физический дисплей, отображение на котором синхронизировано с одним пользователем, то другие пользователи на этом дисплее не смогут увидеть изображения, соответствующие их точкам наблюдения и движениям. Эффект аквариума может быть уменьшен, если окружить пользователя несколькими дисплеями, как предполагается в технологии *CAVE*. Но полностью стереоскопическое восприятие для нескольких пользователей все равно будет недоступно.

## 15.6 КРАТКИЙ ОБЗОР ДАТЧИКОВ ДЛЯ СИСТЕМ ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ

**Зрительная информация.** Как описывалось в предыдущем разделе, для вывода зрительной информации в системах виртуальной реальности обычно применяются устройства вывода стереоскопических изображений на основе головного дисплея или на основе обычного компьютерного дисплея, наблюдаемого с помощью очков со шторками. Представляемые пользователю изображения генерируются с учетом данных от датчиков, которые воспринимают положение пользователя в виртуальной среде, так что изображения соответствуют выбранной точке наблюдения.

Если в системе виртуальной реальности предусмотрен ввод зрительной информации, то при этом обычно выполняется слежение за глазами, голо-

вой или другими частями тела пользователя. В результате определяются параметры положения, которые используются в качестве входных данных для моделирования активных действий пользователя. Слежение за глазами может применяться для определения направления взгляда. Устройства для слежения за глазами могут устанавливаться в головном дисплее или отдельно от пользователя. Прослеживание положения головы можно организовать с помощью камеры, следящей за специальными признаковыми точками, которые нанесены на головной дисплей пользователя. В последних работах по виртуальным средам продемонстрированы возможности слежения за руками, головой и ногами пользователя с использованием наилучших изображений, полученных с нескольких камер. Одно из преимуществ устройств ввода зрительной информации заключается в том, что они не являются ограничивающими свободу движений носимыми устройствами. Однако, хотя перемещения пользователя могут не ограничиваться, но сами устройства наблюдают ограниченный объем пространства. Некоторые устройства слежения за глазами требуют специальной инфракрасной подсветки, а устройства слежения за телом пользователя иногда требуют обеспечения однородного цветного фона.

**Звуковая информация.** Устройства голосового ввода данных в компьютер доступны уже более пятнадцати лет. В настоящее время они используются во многих интерфейсах, например, в телефонных системах и программах для домашних персональных компьютеров. Преимущество голосового ввода данных состоит в том, что он является естественным для человека, т. е. не требует специального обучения. Голосовой ввод может быть необходим в случаях, когда глаза и руки пользователя заняты выполнением других действий. Голосовой вывод данных является удобным коммуникационным каналом, не зависящим от дисплея.

Звуковой вывод способен улучшить многие интерфейсы (например, звук удара по металлу может подтверждать удаление файла после перемещения его пиктограммы на пиктограмму мусорной корзины). Звуковой вывод также повышает степень погружения в виртуальную среду. Водитель виртуального автомобиля может слышать звук двигателя и визг колес при торможении. Или телеоператор может воспринимать уровень радиоактивности по частоте звуковых щелчков.

**114 Определение.** Звуковое кодирование информации, не имеющей звукового происхождения, называется **сонификацией** (sonification).

**Информация о расположении.** Датчики пространственного местоположения воспринимают данные о положении и ориентации частей человеческого тела или инструментов. Датчики с шестью степенями свободы, в том числе датчик Polyhemus, часто используются в головных дисплеях, джойстиках и новых устройствах, таких, как «летучая мышь», описанная в работе [Green, Halliday, 1996]. Среди трехкоординатных ( $x-y-z$ ) датчиков



можно назвать чувствительные перья и различные механические устройства. Менее распространены устройства вывода данных о положении посредством позиционирования механических узлов, закрепленных на теле человека. Этот вопрос относится к силовой обратной связи, которая будет описана далее.

**Осязательная информация.** В человеческом восприятии окружающего мира важную роль играют ощущения прикосновений, силовых воздействий и движения. Осязательные ощущения формируются нервными окончаниями, расположенными в кожном покрове. Эти окончания чувствительны к температуре, плотности и гладкости поверхностей. Нервы в конечностях и мускулах способны воспринимать положение конечности и мускульное усилие и/или длину конечности и ее изменение. Нервы в вестибулярном аппарате позволяют ощущать движение тела.

Для силового ввода/вывода разработаны различные электромеханические устройства.

**115 Определение.** К осязательным ощущениям человека относятся ощущения прикосновений (тактильные), а также ощущения положения тела, силовые и двигательные (кинестетические ощущения).

**Информация о движении.** На человеческое ощущение движения интегрально влияют многие системы восприятия. Визуальные картины, как в кинофильмах или тренажерах, способны обеспечивать достаточно стимулов для вызова двигательных реакций. В некоторых системах виртуальных сред кинестетические стимулы формируются посредством роботизированных манипуляторов, платформ или центрифуг, позволяющих перемещать оператора. Эти средства повышают степень погружения по сравнению с использованием только зрительной информации.

На вестибулярную систему также можно искусственно воздействовать посредством потока прохладного воздуха или воды. Компьютерное зрение способно играть важную роль в процессе восприятия движения. В идеале, оператор-человек мог бы свободно двигаться в реальной среде и обработка изображений от нескольких следящих камер позволила бы получить всю информацию, необходимую для интерпретации движения. Параметры движения могли бы учитываться в компьютерной модели человеческого тела.

Для слежения за движением человеческого тела разработано несколько коммерческих систем. Обычно они требуют закрепления на человеческом теле специальных отражающих мишеней. Эти мишени упрощают сегментацию изображений и выделение признаков. Подобные системы используются при обучении спортсменов и при подборе ортопедических приспособлений. В системах исследовательского характера были продемонстрированы возможности слежения за головой, руками и ногами человека без использования закрепленных на теле мишеней.

## 15.7 ГЕНЕРАЦИЯ ПРОСТЫХ ТРЕХМЕРНЫХ МОДЕЛЕЙ

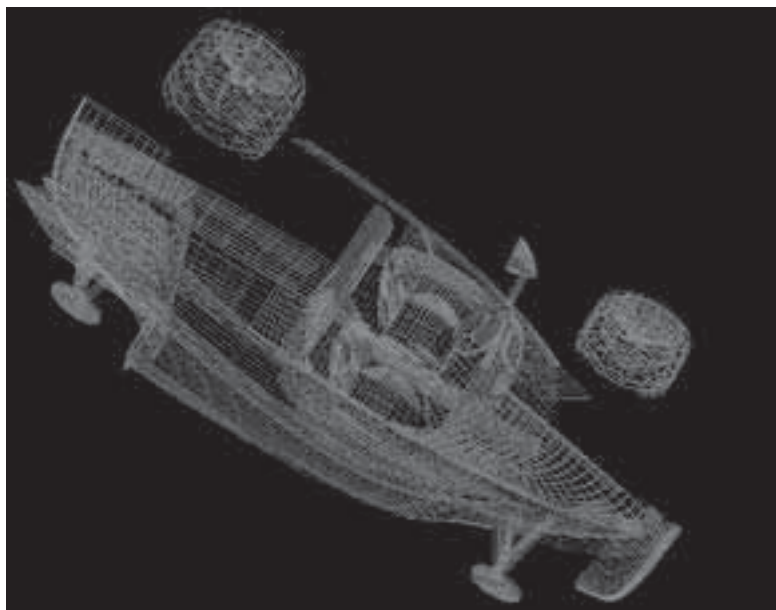
Для формирования виртуальных сцен необходимы средства для построения моделей объектов и генерации их изображений. Используемые модели объектов могут быть как сложными моделями, рассматривавшимися в гл. 13 и 14, так и простыми каркасными моделями, которые легко построить с помощью систем автоматизированного проектирования (САПР), например, AutoCAD. На рис. 15.13 показана каркасная модель перспективного автомобиля Форд, созданная в интерактивном режиме в одном из пакетов САПР.

После получения трехмерной модели требуются средства для построения ее изображений, соответствующих произвольным точкам наблюдения и условиям освещения.

**116 Определение.** Процесс создания изображений на основе моделей объектов называется **рендерингом** (rendering).

В рендеринге можно выделить два этапа:

1. Определение поверхностей модели, видимых с выбранной точки наблюдения.
2. Вычисление значений пикселей формируемого изображения.



**Рис. 15.13.** Каркасная модель перспективного автомобиля Форд. (С любезного разрешения компании Ford Motor.)



**Рис. 15.14.** Результаты рендеринга модели перспективного автомобиля Форд. (С любезного разрешения компании Ford Motor.)

На первом этапе обработка может выполняться посредством построения лучей, исходящих из точки наблюдения в заданном направлении на объект. Точка первого пересечения луча с поверхностью объекта будет точкой поверхности, видимой вдоль данного луча. Этот подход называется *трассировкой лучей* (ray tracing). Для его реализации разработано большое количество алгоритмов. В настоящее время в компьютерах применяется аппаратное средство, называемое *z-буфером*, ускоряющее определение видимых поверхностей объектов.

Обработка на втором этапе может быть как простой, так и очень сложной. В простом случае считается, что объект окрашен в определенный цвет и сделан из материала с известными отражающими свойствами. Свет распространяется от точечного источника света в определенном направлении. Также присутствует некоторое количество рассеянного света. Для определения цвета пиксела, соответствующего малому поверхностному элементу объекта, используются различные математические модели, например, модель освещения Фонга (см. гл. 6). На рис. 15.14 показаны результаты простого рендеринга каркасной модели автомобиля.

Для генерации более реалистичных изображений (за счет увеличения времени обработки) учитываются различные дополнительные факторы, такие, как несколько источников света, протяженные источники света, тени, прозрачные поверхности, взаимные отражения.

На рис. 15.15 показаны два изображения, полученные в результате рендеринга с использованием нескольких различных моделей автомобилей. Хотя рендеринг некоторых объектов на этих изображениях был выполнен согласно описанной выше процедуре, часть объектов имеют текстурированные поверхности со сложными образами. (На левом изображении обратите внимание на здание далеко слева и на тротуар рядом с ним.) Рендеринг подобных объектов мог бы занять очень много времени. Но для получения изображений этих



**Рис. 15.15.** Сцены с использованием моделей нескольких объектов. (С любезного разрешения Habib Abi-Rached.)

объектов вместо рендеринга применялся метод *текстурного отображения*. Этот метод рассматривается в следующем разделе.

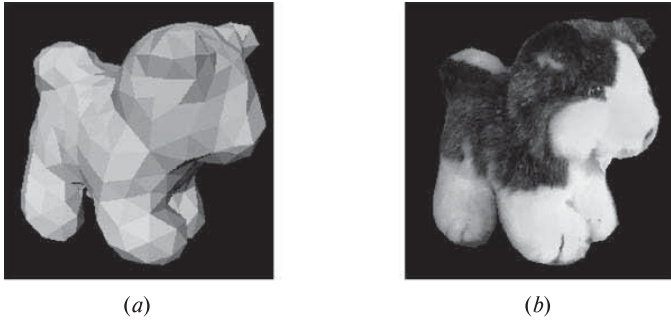
## 15.8 СОЧЕТАНИЕ РЕАЛЬНЫХ И ИСКУССТВЕННЫХ ИЗОБРАЖЕНИЙ

Выше рассматривался рендеринг моделей объектов для получения полностью искусственных изображений. Часто они оказываются недостаточно реалистичными. Для повышения степени реалистичности необходимы подробные геометрические модели, рендеринг которых требует значительных вычислительных затрат. Для повышения реалистичности и для ускорения рендеринга вместо подробных моделей можно пользоваться готовыми изображениями сложных текстур. *Текстуры* могут генерироваться искусственным образом или извлекаться из реальных изображений. В процессе рендеринга вместо закраски поверхности заданным цветом можно было бы извлекать значения цветов пикселей из изображения текстуры. В этом заключается метод *текстурного отображения*, в котором окончательные значения видимых пикселей выбираются из пикселей заданной текстуры.

**117 Определение.** Процесс нанесения текстуры на гладкую поверхность для формирования текстурированного изображения этой поверхности называется **текстурным отображением** (texture mapping).

На рис. 15.15 показаны поверхности, обработанные методом текстурного отображения. Эти поверхности состоят из плоских многоугольников. Текстурование подобных поверхностей выполняется наиболее просто. Текстурное отображение может также выполняться и для более сложных криволинейных поверхностей (например, при нанесении текстуры кожуры на модель апельсина). Если объект произвольной формы представлен в виде кусков поверхности, то его текстурование можно выполнять по частям, по очереди обрабатывая каждый кусок поверхности. Для сложных объектов необходимы более совершенные методы.

В компьютерной графике недавно был разработан метод, в котором текстуры извлекаются из реальных изображений объектов. На рис. 15.16 (a) показана



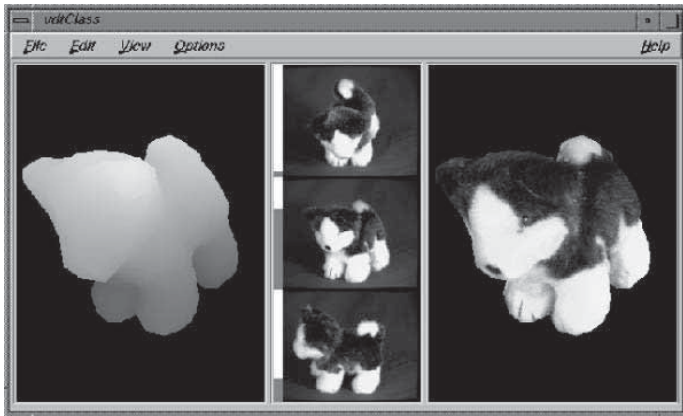
**Рис. 15.16.** Грубая полигональная модель игрушечной собаки и изображение, полученное методом текстурированного отображения. (С любезного разрешения Kari Pulli.)

грубая модель игрушечной собаки (эта модель уже встречалась в гл. 13), а на рис. 15.16 (b) представлено текстурированное изображение этой модели. В данном примере текстура извлекается из реального изображения игрушечной собаки, соответствующего точке наблюдения, используемой при отображении модели. На практике, как и при рендеринге, необходимы средства отображения текстурированных моделей для произвольных точек наблюдения.

**118 Определение. Рендерингом на основе изображений** (image-based rendering) называется метод, в котором для получения искусственного изображения объекта, видимого с произвольной точки наблюдения, используется набор реальных изображений того же самого объекта.

Рендеринг на основе изображений может выполняться без геометрической модели объекта. Если имеется большое количество изображений объекта с различных точек наблюдения, то с помощью интерполяции между этими изображениями можно получить изображение объекта для произвольной точки наблюдения. Реалистичные результаты рендеринга также можно получить на основе «грубой» полигональной модели объекта и небольшого количества реальных изображений для ограниченного множества точек наблюдения.

Система формирования моделей, описанная в гл. 13, которая предназначалась для построения грубых полигональных моделей объектов по набору дальнометрических изображений и соответствующих им цветных изображений, также может применяться для рендеринга объектов с учетом указанной пользователем точки наблюдения. Для этого используется метод, называемый *текстурированием на основе видов* (view-based texturing). Основная идея этого метода поясняется на рис. 15.17. Слева представлено псевдоцветовое изображение объекта. С помощью мыши пользователь может поворачивать этот объект для указания требуемой ориентации. В центре показаны три изображения, извлеченные из базы данных. Виды на этих изображениях наиболее близки к выбранной пользователем точке наблюдения. Справа показано



**Рис. 15.17.** (Слева) дальнометрические данные модели игрушечной собаки. (В центре) три реальных цветных изображения, на которых вид игрушечной собаки наиболее похож на вид, соответствующий указанной пользователем точке наблюдения. (Справа) изображение с результатами рендеринга с использованием взвешенной комбинации пикселей выбранных видов. (С любезного разрешения Kari Pulli.) См. цветную вставку.

результатирующее текстурированное изображение игрушечной собаки, соответствующее заданной пользователем точке наблюдения. При создании каждого не фонового пиксела изображения производится построение луча, исходящего из пиксела изображения и пересекающего поверхность трехмерной модели. После определения точки на поверхности модели из этой точки строятся лучи к соответствующим пикселям каждого из трех близких видов. Цвета этих трех пикселей комбинируются для вычисления значения текущего пиксела формируемого изображения. Комбинирующий алгоритм не просто усредняет значения трех пикселей, а учитывает близость сохраненного и требуемого видов, направление луча от модели объекта к пикселу сохраненного вида, и расстояние пиксела сохраненного вида от его границы. При формировании изображения также применяется программно реализованная  $z$ -буферизация, которая позволяет исключать из расчетов слишком далекие пиксели.

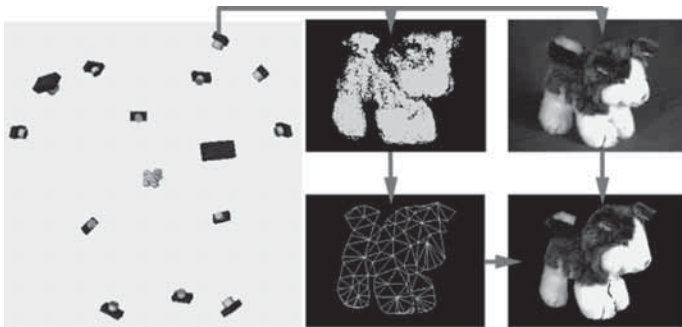
Построение полных трехмерных полигональных моделей реальных объектов занимает много времени. Текстурирование на основе видов фактически не требует полной геометрической модели. Этот метод позволяет работать с небольшим набором исходных совмещенных дальнометрических и цветных изображений. Процесс обработки модели игрушечной собаки поясняется на рис. 15.18. Для каждого выборочного вида вместо полной полигональной модели строится частичная полигональная модель. Эти частичные модели могут использоваться вместе с цветными изображениями объекта для выполнения рендеринга, результаты которого по качеству близки к случаю рендеринга полных полигональных моделей.

Для некоторых реальных объектов затруднительно построить твердотельные модели. Если у объекта есть тонкие детали (например, паруса яхты или листья растений), то для удовлетворительного представления геометрических свойств объекта требуется полигональная модель слишком высокого разрешения. Хранение дальнометрических и цветовых данных для небольшого количества видов обычно не представляет сложностей, поэтому текстурирование на основе видов может применяться для получения реалистичных изображений объектов с тонкими деталями, видимых с различных точек наблюдения. Процесс обработки для получения изображения корзины с цветами поясняется на рис. 15.19.

### Упражнение 15.5. Дополнение реального изображения куба

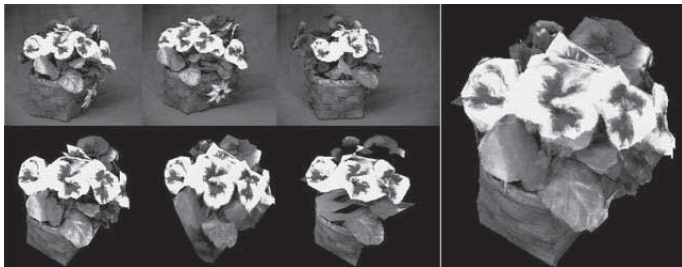
Выполните калибровку камеры с помощью метода, описанного в гл. 13. В качестве *калибровочного стенда* используйте куб или небольшую коробку (управляющими точками будут семь видимых углов). Сформируйте двумерное изображение грани игральной кости с тремя точками, представляющими число 3. Для верхней грани трехмерного куба постройте отображающую функцию  $g$  (см. гл. 11), которая отображает точку  $[x_m, y_m, z_m]$  с верхней грани трехмерного куба на двумерную точку  $[x_t, y_t]$  внутри квадрата, соответствующего изображению грани игральной кости.

Сгенерируйте и напечатайте изображение, содержащее пиксели из реальной калибровочной сцены везде, кроме верхней грани куба, в области



**Рис. 15.18.** Совмещенные дальнометрические и цветные изображения для небольшого количества видов объекта могут использоваться для высококачественного рендеринга изображения. При этом не требуется построение полной модели объекта. (Слева) возможные точки наблюдения. (Сверху в центре) дальнометрические данные для одной из точек наблюдения. (Сверху справа) цветные данные для той же точки наблюдения. (Снизу в центре) полигональная модель, построенная по дальнометрическим данным. (Снизу справа) результаты рендеринга с применением текстурного отображения цветных данных на полигональную модель. (С любезного разрешения Kari Pulli.) См. цветную вклейку.





**Рис. 15.19.** Применение текстурирования на основе видов для получения изображения объекта с большим количеством тонких деталей. Для этого объекта сложно построить точную трехмерную модель. (Сверху слева) три различных цветных изображения объекта. (Снизу слева) три изображения с различных выбранных точек зрения построены посредством отображения пикселей трех исходных изображений на изображение, соответствующее новой точке наблюдения. (Справа) окончательные результаты рендеринга. Значения пикселей этого изображения вычислены как взвешенные комбинации трех построенных ранее изображений. (С любезного разрешения Kari Pulli.) См. цветную вклейку.

которой должны находиться пиксели из модели игральной кости. Повторите эту процедуру с использованием той же самой отображающей функции  $g$ , но с другим квадратным сканированным изображением (например, собственной фотографией).

### Упражнение 15.6. Искусственное изображение куба

При выполнении этого упражнения пользуйтесь матрицей камеры, полученной в упражнении 15.5, или возьмите матрицу камеры из гл. 13. (а) Сформируйте искусственное изображение куба, расположенного в произвольной области поля зрения камеры. (б) Сформируйте изображение, как в задании (а), но на одной из граней куба постройте изображение грани игральной кости, как в предыдущем упражнении. (\*в) С помощью текстурированного отображения выполните наложение двух различных фотографий лиц людей на две грани куба. Для этой цели воспользуйтесь преобразованиями из гл. 11.

## 15.9 ПСИХОФИЗИОЛОГИЧЕСКИЕ АСПЕКТЫ ЧЕЛОВЕКО-МАШИННОГО ИНТЕРФЕЙСА

С помощью устройств, описанных в данной главе, можно различными способами улучшить качество и производительность интерфейса между человеком и компьютером. Основное назначение сред виртуальной реальности заключается в обеспечении качественного погружения пользователя в виртуальную среду. Интерфейс с использованием зрения, осязания и силовой связи необходим, например, в задачах виртуальной хирургии. Индивидуальные различия между



людьми создают сложности при разработке систем виртуальной реальности. Например, разработка головных дисплеев усложняется из-за необходимости обеспечения настройки на размеры и форму головы пользователя. Индивидуальные особенности зрительных систем усложняют управление дисплеями для вывода стереоизображений. У людей также имеются индивидуальные особенности восприятия одних и тех же цветов, рельефа поверхностей, звуков и т. п.

Системы виртуального времени могут приводить к нежелательным последствиям, например, к морской болезни. Другими побочными эффектами могут быть зрительное перенапряжение, усталость или чувство неудовлетворенности. Все эти эффекты могут возникать при неточной настройке подсистемы вывода стереоизображений. Перечисленные вопросы требуют дополнительного изучения разработчиками систем виртуальной реальности.

## 15.10 ЛИТЕРАТУРА

Обзор работ в области систем виртуальной реальности по состоянию на 1995 г. представлен в отчете [Durlach, Mavor, 1995]. В сборник Бафилда и Фурнесса [Barfield, Furness, 1995] включен представительный набор статей с результатами последних исследований по различным аспектам систем виртуальной реальности. Обе книги содержат значительный объем базовых сведений по данной теме, в том числе описание устройств, основные определения, отличные примеры и иллюстрации, а также классифицированные ссылки на литературу.

Системы виртуальной реальности находят разнообразное применение в медицине. В работе [Poston, Serra, 1996] обсуждается их использование при планировании хирургических операций и для интерактивной сегментации трехмерных сцен. Наше описание выполнения сложных ручных операций во многом основано на материалах данной работы. Применение систем виртуальной реальности для реабилитации и лечения различных фобий рассмотрено в статьях сборника [Strickland, 1997].

Создание моделей объектов и определение их поведения и движения является сложной проблемой и темой современных исследований. Описание этой проблемы и некоторых методов ее решения можно найти в работах [Green, Halliday, 1996] и [Deering, 1996]. Возможности систем виртуальной реальности по погружению пользователя в виртуальную среду могут не только помогать людям, но и делать их раздражительными и вызывать болезненные состояния. Нежелательные побочные эффекты особенно вероятны при плохой настройке подсистемы вывода стереоизображений. По этому поводу можно обратиться к статье [Viire, 1997].

В книге [Stuart, 1996] приведено много полезных таблиц с перечислением свойств человеческого восприятия различных ощущений, а также характеристик устройств ввода/вывода для систем виртуальной реальности. Текстурирование на основе видов описано в работе [Pulli и др., 1997]. Ряд работ по

системам *смешанной реальности* включен в недавно вышедший сборник [Ohta, Tamura, 1999].

1. Barfield, W., and T. Furness III, eds. 1995. *Virtual Environments and Advanced Interface Design*. Oxford University Press.
2. Biocca, F., and F. Levy. 1995. *Communication in the Age of Virtual Reality*, F. Biocca and M. R. Levy, eds. L. Erlbaum Associates, Hillsdale, NJ.
3. Deering, M. 1996. The Holosketch VR Sketching System, *Communications of the ACM*, v. 39(5):54–61.
4. Durlach, N., and A. Mavor, eds. 1995. *Virtual Reality: Scientific and Technological Challenges*. National Research Council, National Academy Press, Washington, D.C.
5. Green, M., and S. Halliday. 1996. A geometric modeling and animation system for virtual reality. *Communications of the ACM*, v. 39(5):46–53.
6. Ohta, Y., and H. Tamura, eds. 1999. *Mixed Reality: Merging Real and Virtual Worlds*. Ohmsha, Ltd., Tokyo, Japan; also distributed by Springer-Verlag, New York.
7. Poston, T., and L. Serra. 1996. Dextrous virtual work. *Communications of the ACM*, v. 39(5):37–45.
8. Pulli, K., M. Cohen, T. Duchamp, H. Hoppe, L. Shapiro, and W. Stuetzle. 1997. View-based rendering: visualizing real objects from scanned range and color data. *Proc. 8th Eurographics Workshop on Rendering* (June 1997).
9. Strickland, D., ed. 1997. Special issue on VR and health care. *Communications of the ACM*, v. 40(8).
10. Stuart, R. 1996. *The Design of Virtual Environments*. McGraw-Hill, New York.
11. Viire, E. 1997. Health and safety issues for VR. *Communications of the ACM*, v. 40(8):40–41.

# Примеры прикладных задач

В этой главе описываются две коммерческие системы, в которых для решения реальных прикладных задач применяются методы компьютерного зрения и распознавания образов. Рассмотрение способов решения этих прикладных задач позволит ознакомиться с общей структурой систем, в которых интегрированы различные аппаратные устройства и программные алгоритмы. Большинство (но не все) методов, используемых в этих системах, были рассмотрены в предыдущих главах данной книги. В качестве первого примера описывается разработанная фирмой IBM система *VeggieVision*, которая предназначена для распознавания продуктов на контрольных весах в супермаркете. Вторая рассматриваемая система представляет собой систему идентификации личности по радужной оболочке глаза. Эта система может использоваться в банкоматах или каких-либо системах безопасности для подтверждения или распознавания личности пользователя.

## **16.1 VEGGIE VISION: СИСТЕМА ДЛЯ РАСПОЗНАВАНИЯ ОВОЩЕЙ И ФРУКТОВ**

Применение штрих-кодов позволило значительно сократить трудозатраты на организацию продажи товаров в супермаркетах. Однако обработка товаров по-прежнему требует значительных человеческих усилий. Иногда некоторые продукты, например, картофель или яблоки, упаковываются и помечаются штрих-кодом, чтобы с ними можно было обращаться так же, как и с продуктами в готовой упаковке. Однако подобные товары из-за этого могут оказаться менее привлекательными для покупателя, например, потому, что он не сможет выбирать отдельные экземпляры овощей и фруктов. Покупатель может захотеть выбрать, например, помидоры или даже зеленые бобы и положить их в свой полиэтиленовый пакет. В большинстве магазинов стоимость подобных продуктов обычно определяется с помощью весов.

Тип продуктов и их идентификационные коды могут вводиться кассиром. В такой ситуации становится актуальной задача автоматической обработки — почему бы не выполнять автоматическую идентификацию типа продуктов с помощью камеры? Подобная система смогла бы существенно упростить работу человека-контролера и позволила бы усовершенствовать контроль за запасами товаров. Для решения этой задачи в исследовательском центре фирмы IBM была разработана система **Veggie Vision**. Она продемонстрировала эффективную работу в процессе лабораторных испытаний, и в настоящее время находится в стадии опытной эксплуатации. Автоматическая система обладает еще несколькими преимуществами, например, она позволяет более точно назначать цену с учетом размеров и качества овощей и фруктов. В следующих разделах будет описана задача распознавания типа продуктов в супермаркете и ее решение, предложенной фирмой IBM. Мы выражаем особую благодарность Боллу (Bolle), Коннелу (Connell) и др. (1996) за предоставленную документацию по системе **Veggie Vision**. В их статье, ссылка на которую приведена в конце данной главы, читатель сможет найти более подробное описание системы **Veggie Vision**.

### 16.1.1 Прикладная область и требования к системе

В США в розничной торговле продается примерно  $m = 350$  наименований овощной продукции, но в одном магазине обычно имеется не более 150 наименований. Ни одно из этих чисел не представляет серьезной проблемы для организации автоматической идентификации продуктов. Чтобы применение автоматической системы было экономически оправданным, эта система должна выполнять идентификацию типа продуктов в пределах 1 секунды и использовать вычислительную аппаратуру, стоимость которой не превышает стоимости сканеров и компьютеров, используемых в торговле в настоящее время. Желательно, чтобы новое оборудование допускало установку в пределах пространства, занятого существующим торговым оборудованием, и чтобы при этом в магазине не требовалось изменять сложившуюся обстановку.

Система должна быть настраиваемой для использования в конкретных магазинах посредством учета нескольких факторов. Во-первых, не во всех магазинах продаются одни и те же продукты. Во-вторых, ассортимент и стоимость продуктов в одном и том же магазине может меняться в зависимости от сезона или даже в течение дня (рассмотрим, например, продажу бананов, которые сначала поступили слегка зелеными и затем стали понемногу желтеть). Эффективная система должна быть спроектирована так, чтобы она могла легко адаптироваться к подобным изменениям и обладала возможностями расширения для обработки продуктов новых типов.

В заключение отметим, что в работе всей системы человек должен выполнять приемлемое количество действий. К ним относятся начальное обучение работе с системой, включение системы для работы в автоматическом режиме, и принятие решений в случае, если процесс автоматической обработки

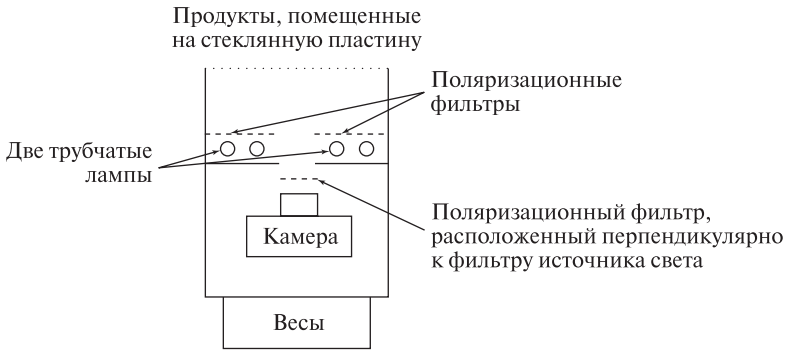


**Рис. 16.1.** На эскизе, выполненном дизайнером, показан внешний вид системы для распознавания овощей и фруктов в супермаркете. (Возможные типы продуктов отображаются на экране в случаях, когда требуется взаимодействие системы с оператором.) (С любезного разрешения R. Bolle и J. Connell.)

по каким-либо причинам была остановлен. Система в целом, включающая машину и оператора-человека, должна быть намного более эффективной по сравнению с существующими методами ручного контроля, применяемыми в большинстве магазинов. На рис. 16.1 показан общий вид разрабатываемой системы. Входящий в нее сенсорный экран, кроме отображения результатов обработки, позволяет оператору выполнить идентификацию продуктов в случаях, когда системе не удается автоматически получить надежный результат.

### 16.1.2 Устройство системы

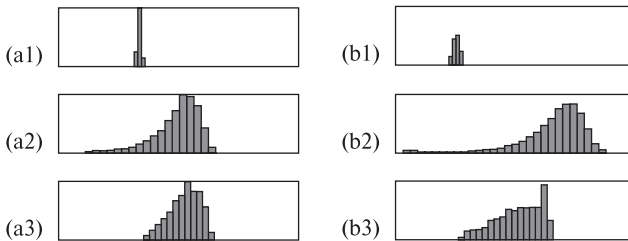
**Аппаратные компоненты** Объем сканирующей аппаратуры не должен превышать объема, занимаемого существующими устройствами сканирования штрих-кодов и весами. Кроме того, аппаратура должна работать в обстановке различных магазинов без каких-либо специальных изменений. Схема разработанного сканера показана на рис. 16.2. Перед источником света и перед камерой расположены поляризационные светофильтры. Плоскости поляризации этих светофильтров перпендикулярны. Это позволяет избавиться от зеркальных бликов на изображениях продуктов. Чтобы обеспечить экономически оправданную реализацию системы с временными затратами не более 1 секунды на одну итерацию цикла обработки, в системе для обработки изображений применяется цифровой сигнальный процессор (digital signal processing chip, DSP). Цветная камера и цифровой сигнальный процессор играют роль низкоскоростного устройства ввода для кассового аппарата, которому требуется передать только идентификатор и вес продуктов каждого типа, расположенных на весах.



**Рис. 16.2.** Схема устройства сканера. Цветная камера и источник поляризованного света размещены в габаритах типичных весов и считывателей штрих-кодов.

**Представление и распознавание** В предыдущих главах было показано, что в ряде задач в качестве эффективных характерных признаков могут быть выбраны цветовые гистограммы. Это подтверждается исследованиями и результатами решения прикладных задач. Классические текстурные признаки не подошли для решения рассматриваемой задачи, поэтому в ней были выбраны несколько специфических признаков (они будут описаны далее), а также простой признак формы. Для изображений продуктов строились цветовая гистограмма, текстурная гистограмма, а также гистограммы формы и размеров. Эти гистограммы объединялись для формирования вектора признаков  $Q$  с размерностью  $d > 100$ . Вектор признаков рассматривался в качестве описания продуктов неизвестного типа, расположенных на весах. На рис. 16.3 показаны цветовые гистограммы, построенные для изображений одного из сортов яблок (слева) и апельсинов (справа).

Для обеспечения адаптации системы к изменяющимся условиям был выбран метод классификации по ближайшему соседу. В виде простого массива в памяти системы хранятся маркированные эталонные образцы векторов



**Рис. 16.3.** (Слева) цветовые гистограммы для яблок и (справа) апельсинов. В направлении сверху-вниз расположены гистограммы, построенные по значениям тона, насыщенности и интенсивности. (С любезного разрешения R. Volle и J. Connell.)

признаков для продуктов известных типов. В расчете на максимальное количество классов  $m = 350$  и в предположении, что для каждого класса допускается наличие до 10 эталонных образцов, в памяти может храниться до 3500 эталонных образцов. Цифровой сигнальный процессор позволяет быстро выполнять сравнение распознаваемого вектора признаков  $Q$  со всеми 3500 маркированными образцами. В результате сравнения менее чем за 1 секунду вычисляется множество  $k$  ближайших соседей вектора  $Q$ . Для хранения эталонных образцов не применяется каких-либо специальных структур данных. Это упрощает обновление базы данных эталонных образцов. Каждый вектор хранится в базе данных вместе с сопутствующей информацией. Например, для векторов признаков можно хранить время создания и время последнего использования, чтобы, при необходимости, удалять из памяти давно не использовавшиеся эталонные образцы.

Расстояние между неизвестным вектором запроса  $Q$  и  $j$ -м эталонным образцом класса  $L$ , обозначаемым как  ${}^L P_j$ , вычисляется с помощью методов, описанных в гл. 8. Так как все отдельные признаки представляются в виде значений счетчиков на гистограммах, то расстояние  $d(Q, {}^L P_j)$  можно определить как модуль суммы разностей счетчиков, соответствующих векторам  $Q$  и  ${}^L P_j$ .

$$d^j = d(Q, {}^L P_j) = \sum_{f \in F} w_f d(Q, {}^L P_{j,f}) \quad (16.1)$$

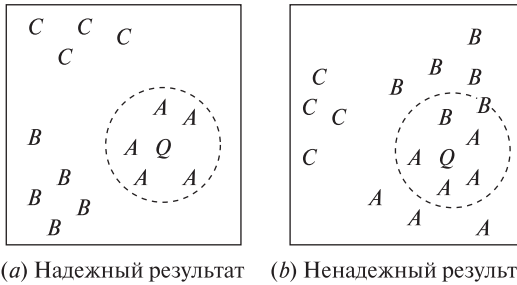
Для управления идентификацией и для регулировки точности распознавания применяются пороговые значения. Используется порог по расстоянию  $t$ , определяющий, является ли вектор  $Q$  достаточно близким к некоторому образцу или образцам  $P_j$ . Введем обозначение  $k_t$  для количества соседей, близких к вектору  $Q$ , которые были выбраны из базы данных в памяти системы. Процедура идентификации описывается в следующем разделе.

### 16.1.3 Процедура идентификации

Общий алгоритм идентификации продуктов на весах приведен в виде алгоритма 16.1. Некоторые шаги этого алгоритма более подробно описаны в разделе 16.1.4. Использование ближайших соседей на множестве эталонных образцов для классификации вектора, представляющего продукт неизвестного типа, поясняется на рис. 16.4.

### 16.1.4 Более подробное описание процесса обработки

**Получение изображений продуктов** В процессе получения изображений не имеется возможностей существенного воздействия на обстановку в магазине. В частности, на установленную внутри весов камеру будет попадать отраженный от потолка свет и свет от расположенных в поле зрения камеры источников света. В системе *VeggieVision* выполняется ввод двух изображений продуктов, расположенных на весах. Одно изображение вводится



**Рис. 16.4.** Схема, поясняющая процедуру принятия решения в пространстве характерных признаков. Показано двумерное пространство признаков, хотя в действительности оно является  $d$ -мерным. (a) Результат идентификации считается *надежным*, если все эталонные образцы, лежащие на расстоянии не более  $t$  от вектора  $Q$ , принадлежат только одному классу. (b) В противном случае результат считается *ненадежным*. Тогда продукты подвергаются повторному сканированию, или система выдает запрос оператору, чтобы он выбрал один из двух близких классов  $A$  и  $B$ .

при включенном источнике света, установленном внутри весов (изображение с подсветкой), а другое — при выключенном внутреннем источнике света (изображение без подсветки). В процессе сегментации этих изображений требуется выделить три области: (1) область продуктов, которая должна быть темной на изображении без подсветки и яркой на изображении с подсветкой; (2) область фона, имеющую одинаковую яркость на обоих изображениях (с подсветкой и без подсветки), и (3) если продукты находятся в полиэтиленовом пакете, то его части будут не слишком темными на изображении без подсветки и не слишком яркими на изображении с подсветкой. Для изображений подобной структуры можно подобрать такие пороговые значения, которые позволят в процессе сегментации отделить область продуктов от областей полиэтиленового пакета и фона. Чтобы получить качественное цветное изображение, для подавления зеркально отраженного света, не несущего полезной информации о поверхности продуктов, в системе используется поляризованный свет. Кроме того, поверхности продуктов, видимые на изображениях, не освещаются какими-либо посторонними источниками света, кроме источника, установленного внутри весов. Следовательно, воспринимаемые цвета будут соответствовать цветам поверхностей продуктов даже при изменении освещения в помещении.

**Вычисление признаков** Признаки вычисляются только для области (или областей) образов продуктов. Используемые признаки должны быть инвариантны относительно поворота, но не относительно масштаба. Для признаков четырех типов строятся гистограммы, которые затем объединяются для формирования одного вектора  $Q$ , который используется в качестве описания неизвестного продукта. Этими четырьмя признаками являются цвет, текстура, форма и размер.



### Идентификация продуктов, помещенных на контрольные веса в магазине.

1. По сигналу оператора вводятся два изображения: без подсветки и с подсветкой. На этих изображениях выделяются области переднего плана, соответствующие образам продуктов.
2. Выполняется построение гистограмм для признаков цвета, текстуры, формы и размера. Затем эти гистограммы объединяются для формирования вектора признаков  $Q$ .
3. Вектор  $Q$  сравнивается с каждым эталонным образцом  $L P_j$  из базы данных системы. Игнорируются все образцы, расстояние которых от вектора  $Q$  превышает пороговое значение  $t$ . Все остальные образцы сортируются по возрастанию расстояния между ними и вектором  $Q$ .
4. Если все  $K$  ближайших соседей вектора  $Q$  имеют одинаковые метки  $L$ , то эта метка  $L$  возвращается в качестве идентификатора  $Q$  и результат идентификации считается *надежным*. В данном случае система способна принять решение автоматически.
5. Если при первой попытке обработки результат идентификации оказался ненадежным, то оператору выдается сообщение, чтобы он по-другому расположил продукты на весах. После этого повторяются перечисленные выше этапы обработки 1–4.
6. Если при повторной обработке снова был получен ненадежный результат идентификации, то на экране отображаются не более  $N$  возможных вариантов меток, взятых из отсортированного списка. Оператору предлагается выбрать нужный вариант среди представленных.
7. При необходимости вектор  $Q$  помещается в множество эталонных образцов, а некоторые другие образцы удаляются.

**Алгоритм 16.1.** Общая схема процесса идентификации продуктов в системе VeggieVision

**Цвет** каждого пиксела преобразуется из цветовой системы RGB в систему HSI с компонентами  $(h, s, i)$ . Все три компоненты  $h$ ,  $s$  и  $i$  используются для построения гистограмм. Пикселы с небольшими значениями интенсивности или насыщенности игнорируются, так как преобразование между цветовыми системами численно неустойчиво. Три построенные гистограммы нормируются относительно площади областей образов продуктов. На рис. 16.3 приведены гистограммы для трех цветовых компонент, которые были построены для изображений яблок и апельсинов.

Небольшой набор **текстурных** признаков вычисляется в каждом пикселе с учетом только зеленой цветовой компоненты исходного цветного изображения.

Для вычисления текстурных признаков используются маски, имеющие вид двух прямоугольных областей с общим центром. Внутренняя область маски содержит положительные весовые коэффициенты, а окружающая ее внешняя область содержит отрицательные весовые коэффициенты. Для ускорения вычислений обработка выполняется при пониженном разрешении изображения. Положительные и отрицательные значения отклика масок используются для построения гистограммы. Величина центрального пика гистограммы служит характеристикой общей текстуры объекта. Большой центральный пик свидетельствует о наличии большого количества малых значений отклика на текстурные маски, т. е. о том, что текстура выражена очень слабо. Ширина гистограммы представляет уровень контраста текстуры, например, с ее помощью можно отличить резкие тени между листьями от тонких полос на поверхности. Асимметричность гистограммы характеризует размеры текстурных элементов относительно размера маски. Например, перекося гистограммы в сторону положительных разностей может говорить о том, что на изображении продукта присутствуют большие листья с узкими промежутками между ними. Продукты такого типа удастся отличить от продуктов с сильно извивающимися границами (например, такими, как границы на изображениях петрушки).

Для описания **формы** применяется простой способ. Границы областей, соответствующих продуктам, подвергаются сглаживанию и затем прослеживаются. В процессе прослеживания в каждом граничном пикселе вычисляется значение кривизны. Учитываются только внешние сегменты границы области продуктов. Не используются также края изображения и фрагменты, в которых различные продукты соприкасаются друг с другом. Для повышения качества результатов сегментации строится гистограмма по квадратам значений кривизны. У овощей и фруктов сферической формы эта гистограмма будет содержать один узкий пик, соответствующий радиусу. По местоположению пика гистограммы можно отличить, например, лимоны от грейпфрутов. На гистограммах формы для продолговатых овощей и фруктов (например, бананов и моркови) будет присутствовать большое количество значений и пик в окрестности нуля. Для изображений овощей с листьями получаются широкие распределения значений кривизны.

**Размер** является четвертым признаком для построения гистограммы. Значение *размера* вычисляется не для каких-либо совокупностей пикселей, а для каждого пиксела переднего плана. В пределах маски, соответствующей области переднего плана, по четырем направлениям вычисляются количества подряд идущих пикселей переднего плана. Эти значения длин цепочек вычисляются для горизонтального, вертикального и двух диагональных направлений. Таким образом для различных направлений просмотра обрабатываемого изображения формируются четыре изображения, которые будем называть «направленными изображениями». На каждом из них пиксели представляют длину цепочки пикселей той области, которой принадлежит пиксел. Затем в качестве размера пиксела переднего плана берется минимальное значение среди пикселей с такими же координатами на направленных изображениях.

Таким образом, размер объекта определяется без использования параметрических моделей или каких-либо операций сегментации области переднего плана дополнительно к сегментации, выполнявшейся для отделения фона. Части виноградной грозди приводят к формированию маски переднего плана в форме «расплывчатого облака». Пиксели на внешних выпуклостях будут образовывать цепочки небольшой длины, а внутренние пиксели будут находиться внутри намного более длинных цепочек. Таким образом на гистограмме размера будут присутствовать два пика: один будет соответствовать размеру отдельных виноградин, а другой — размеру всей грозди. Для моркови гистограмма размера будет содержать один узкий пик, характеризующий ширину образа моркови. Это значение ширины может оказаться близким к ширине помидоров мелких сортов. Но у моркови также имеется пик вблизи нулевого значения кривизны на гистограмме формы. Он означает, что объект является продолговатым, а для помидоров это неверно.

**Супервизорное обучение** Использование классификации по ближайшему соседу обеспечивает выполнение обработки за ограниченное время, простое обучение и адаптацию системы. На этапе первоначального обучения системе можно предъявить несколько примеров продуктов различных типов в магазине и ввести в память системы идентификаторы класса для каждого продукта (инвентарные коды). В процессе использования системы оператор-человек может потребовать, чтобы новый вектор признаков  $Q$  был добавлен в множество эталонных образцов. Образец может быть удален в случае избыточности или при малой частоте использования в сравнении с другими образцами данного класса. Если в процессе обучения новый образец удается корректно классифицировать с использованием существующих образцов и расстояние до наилучшего совпадающего образца не превышает  $t_2$ , то новый образец не запоминается в базе данных. В противном случае образец сохраняется в базе данных. Таким образом, в пространстве признаков эталонные образцы одного класса могут образовывать многомодальные распределения. Например, одна мода класса «капуста брокколи» может соответствовать отдельным кочанам брокколи, а другая — брокколи с лиственной. Если количество образцов класса превышает предельно допустимое количество  $M$ , то из базы данных удаляется *наиболее давно* использовавшийся образец. Счетчик использования эталонного образца увеличивается на  $I+$  каждый раз, когда образец оказался ближайшим к неизвестному вектору признаков  $Q$ , и уменьшается на  $I-$  в противном случае.

Нет необходимости выполнять полное обучение системы *VeggieVision* в каждом магазине. Эксперименты показали, что при использовании эталонных образцов, взятых из различных магазинов, производительность системы распознавания несколько снижается. Однако, как было описано выше, система способна к адаптации. Частое вмешательство человека в работу системы требуется только в начале использования системы, когда она настраивает свою

базу данных образцов на продукты, продаваемые в данном магазине. Но в целом усилия человека при этом оказываются намного меньше, чем в случае полного обучения системы VeggieVision с пустой базой данных.

### 16.1.5 Производительность

Разработчиками системы были опубликованы результаты ряда экспериментов с системой, выполнявшихся в течение продолжительного промежутка времени. Подробности о влиянии изменений различных параметров системы можно найти в статье, ссылка на которую приведена в конце данной главы. С системой VeggieVision были проведены эксперименты по распознаванию с использованием 5300 изображений из четырех различных магазинов. Система в 89% случаев выдавала надежные и корректные результаты; в 93% случаев либо выдавала корректный результат, либо включала его в отображаемый оператору список в качестве наиболее вероятного результата; в 96% случаев система либо выдавала корректный результат, либо он был приведен среди первых четырех вариантов выбора, представленных оператору. Оператору мог быть предъявлен запрос для однократного изменения расположения продуктов на весах. Если VeggieVision получала ненадежные результаты в двух попытках, то идентификация могла выполняться оператором посредством нажатия пиктограммы на сенсорном экране. Выяснилось, что применение системы способно существенно сократить человеческие трудозатраты даже в том случае, если бы все решения принимались оператором с использованием сенсорного экрана системы.

---

#### Упражнение 16.1

Предположим, что на изображении образы бананов имеют прямоугольную форму. Как будет выглядеть их гистограмма формы?

---

#### Упражнение 16.2

Изобразите примерный вид и затем сравните гистограммы цвета, текстуры и формы, которые могли бы получиться для красных яблок и для желтых бананов.

---

#### Упражнение 16.3

Допустим, что в полиэтиленовом пакете лежат три яблока и два апельсина. Сможет ли описанная система выполнить идентификацию этих фруктов? Почему?

## 16.2 ИДЕНТИФИКАЦИЯ ЛИЧНОСТИ ЧЕЛОВЕКА ПО РАДУЖНОЙ ОБОЛОЧКЕ ГЛАЗА

В данном разделе описывается система для идентификации личности посредством анализа текстуры образа радужной оболочки глаза на изображении. Датчики для этой системы, предназначенной для применения в банкоматах, были разработаны компанией Sensar. Программное обеспечение для выделения и сопоставления признаков компания Sensar лицензировала у компании IriScan. Мы выражаем особую благодарность Gary Zhang из Sensar и John Daugman из Кембриджского университета за предоставленную информацию и иллюстрации по описываемой прикладной системе.

Задача идентификации личности всегда имела важное общественное значение. Корректная идентификация необходима для выполнения коммерческих и юридических операций; например, при снятии денег с банковского счета или при изменении места жительства человека. Сейчас идентификация личности часто производится следующим образом: человек предъявляет некоторый документ, например, водительские права или паспорт, другому человеку, контролирующему выполнение желаемой операции. В настоящее время множество операций выполняется с помощью машин или компьютерных сетей: при этом безопасность и конфиденциальность обычно обеспечивается человеком, который (а) знает уникальный номер счета и (б) знает *пароль* или персональный идентификационный номер (Personal Identification Number, *PIN*), связанный с этим счетом. Любой человек, узнавший эти коды (с разрешения или без такового), сможет выполнять операции со счетом. Таким образом, права ответственности и контроля могут быть нарушены.

Кроме приложений электронной торговли, идентификация личности необходима в работе полиции. Для этой цели широко используются отпечатки пальцев. Например, место преступления обследуется на предмет поиска отпечатков пальцев, которые могли бы позволить установить личности присутствовавших в данном месте людей. Отпечатки пальцев также применяются для добровольной идентификации личности (например, для идентификации личности сотрудников на предприятиях повышенной секретности). Отпечатки пальцев долгое время были предметом интенсивных исследований. Они используются уже более ста лет. Для считывания отпечатков пальцев были разработаны несколько электронных устройств, которые могут использоваться для ввода этих данных в компьютерных сетях и других системах (см. [Jain и др., 1999]). Применительно к использованию в системах идентификации и аутентификации (авторизации) также интенсивно разрабатываются методы распознавания образов человеческих лиц. Системы распознавания лиц способны идентифицировать людей без их согласия; например, в аэропортах, банках или отелях. Поэтому такие системы находят очевидное применение в работе полиции и для обеспечения безопасности. С ними связан ряд проблем, касающихся приемлемости применения таких систем и обеспечения соблюдения прав личности.

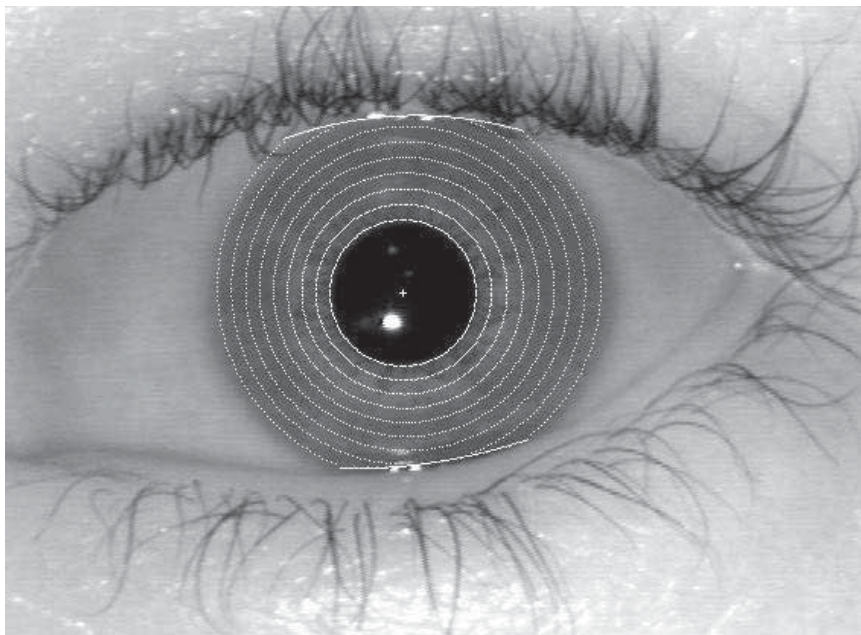
### 16.2.1 Требования к системам идентификации личности

Будем рассматривать системы, которые выполняют операции одного из двух типов: (а) идентификацию личности (по согласию или без согласия человека) на большом множестве возможных вариантов, и (б) подтверждение заявленной личности человека, взаимодействующего с системой. Операции второго типа часто называются аутентификацией. Некоторые из требований к таким системам неочевидны, поэтому ниже приводится их список. Устройство системы существенно зависит от типа используемых *биометрических признаков* и способов получения и представления этих признаков для машинной обработки. Тремя важными биометрическими признаками человека являются внешний вид (1) отпечатков пальцев, (2) лица и (3) радужной оболочки глаза. Ниже будут приведены аргументы, свидетельствующие в пользу применения радужной оболочки глаза для обеспечения более надежной идентификации по сравнению с использованием отпечатков пальцев или образов человеческих лиц.

1. Система идентификации должна получать информацию о личности человека, причиняя ему минимальные неудобства.
2. Биометрический код должен меняться незначительно при его получении от одного и того же человека через произвольные промежутки времени.
3. Биометрические коды разных людей должны существенно отличаться. (Множества идентифицируемых людей будут разными в различных приложениях.)
4. Система должна быть устойчива к вводу подложных данных (например, при предъявлении изображений, напечатанных на бумаге.)
5. Система должна иметь приемлемую стоимость (оценивается применительно к конкретному приложению).

Перед описанием системы сканирования радужной оболочки глаза приведем результаты сравнения систем на основе различных биометрических признаков по критерию соответствия этих систем перечисленным выше требованиям. В качестве способа биометрической идентификации, в дополнение к перечисленным выше, также будет рассматриваться анализ ДНК.

1. **Удобство получения информации:** отпечатки пальцев (приемлемо), образ лица (хорошо), радужная оболочка (хорошо), ДНК (плохо). Для снятия отпечатков пальцев доступны недорогие цифровые сканеры, но пользователь должен аккуратно расположить палец для снятия отпечатка. Изображение лица легко можно получить с минимальными неудобствами для человека с помощью видеокамеры. Для получения качественного изображения радужной оболочки глаза необходима достаточно дорогая оптическая система с дополнительными средствами управления. Получение ДНК, конечно, является дорогостоящей процедурой, которую



**Рис. 16.5.** Изображение глаза, полученное с помощью камеры с узкоугольным полем зрения. В результате обработки на изображении были выделены восемь различных полос радужной оболочки. Из этих полос извлекаются текстурные признаки. (С любезного разрешения J. Daugman, Кембриджский университет. См. Web-страницу по адресу [www.cl.cam.ac.uk/~jgd1000/](http://www.cl.cam.ac.uk/~jgd1000/))

можно выполнить в специальной лаборатории. Обычно это делается только для важных судебных разбирательств.

- 2. Низкая внутриклассовая дисперсия:** отпечатки пальцев (хорошо), лицо (приемлемо), радужная оболочка (отлично), ДНК (отлично). Отметим, что снятие отпечатков пальцев сопровождается сильными деформациями и что вид лица может зависеть от позы, настроения, прически и возраста человека. Структура радужной оболочки, проявляющаяся на изображении в виде текстуры, формируется еще до рождения ребенка и очень мало меняется на протяжении жизни человека. Для получения кодов, описывающих текстуру радужной оболочки, была разработана специальная сканирующая система.
- 3. Высокая межклассовая дисперсия:** отпечатки пальцев (хорошо), лицо (хорошо), радужная оболочка (отлично), ДНК (отлично). Хотя отпечатки пальцев обеспечивают отличные результаты распознавания при анализе экспертами-людьми, но результаты автоматической обработки оказываются хуже. Для большинства людей можно найти людей с похожими лицами. Одни только близнецы могут привести к получению



коэффициента ошибок системы распознавания лиц в 1%! Близнецы также обладают схожими ДНК. Интересно, что текстура радужной оболочки глаз у близнецов различается. Оказывается, что текстуры радужной оболочки даже двух глаз одного и того же человека так же некоррелированы, как и текстуры радужных оболочек глаз разных людей.

4. **Сложность подмены:** отпечатки пальцев (хорошо), лицо (хорошо), радужная оболочка (отлично), ДНК (отлично). Ряд систем на основе анализа отпечатков пальцев или распознавания лиц можно ввести в заблуждение с помощью изображений или простых моделей внешнего вида человека. В центре радужной оболочки глаза располагается зрачок, размер которого изменяется и автоматическая система может отследить эти изменения с целью защиты от подложных данных.
5. **Стоимость:** отпечатки пальцев (приемлемо), лицо (приемлемо), радужная оболочка (приемлемо), ДНК (плохо). Существуют недорогие устройства для получения отпечатков пальцев и изображений лиц, однако методы сопоставления их описаний могут оказаться достаточно дорогими в реализации. Сканирование радужной оболочки глаза требует применения более дорогих устройств, но сопоставление описаний не представляет сложностей. Процедура анализа ДНК, конечно, требует значительного времени, человеческих трудозатрат и дорогих вспомогательных материалов.

### 16.2.2 Устройство системы

В качестве конкретного примера рассмотрим применение системы сканирования радужной оболочки глаза для идентификации пользователей банкомата. Эта система *Sensar...Secure<sup>TM</sup> System* автоматически сканирует радужную оболочку глаза пользователя, подошедшего к банкомату, и выполняет его идентификацию посредством поиска в базе данных регистрационных записей клиентов. После успешной идентификации клиент может получить доступ к своему счету или, возможно, в качестве дополнительной меры безопасности ему потребуется ввести пароль. Эта система может применяться и для других целей, например, для открытия дверей в охраняемые помещения. Для применения в других приложениях может потребоваться изменить различные параметры системы, которые будут рассматриваться далее.

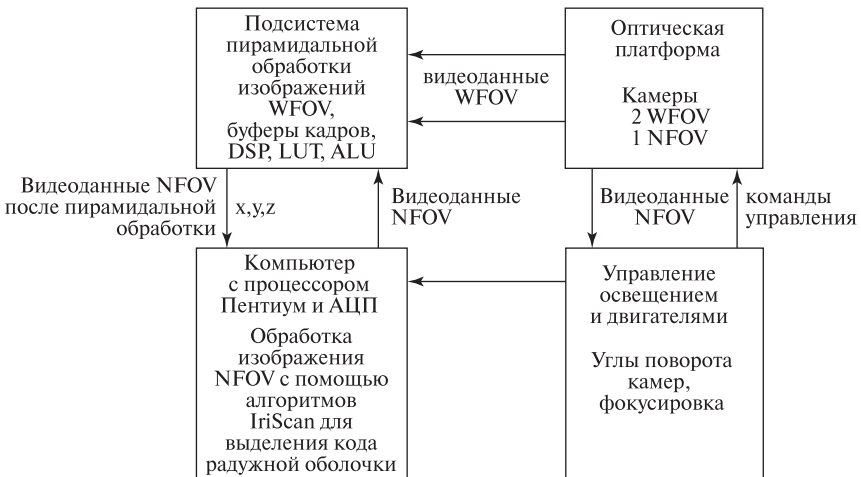
Для получения с высоким разрешением изображения объекта, размер которого мал относительно размеров той области пространства, в которой может появиться этот объект, необходимо тщательно спроектировать сканер и специальную оптическую систему. Кроме того, необходимо выполнять распознавание образов людей в трехмерном пространстве, чтобы при образовании очереди к банкомату можно было обнаружить первого человека в очереди. После определения положения глаз первого человека в очереди выполняется сканирование радужной оболочки одного из его глаз. Затем приме-



няется запатентованное программное обеспечение, формирующее в качестве представления полутоновой текстуры радужной оболочки  $d = 2048$ -мерный бинарный вектор  $Q$ . Сопоставление этого вектора с множеством векторов, представляющих некоторое множество людей, выполняется простым способом на основе вычисления минимального расстояния Хэмминга. Это расстояние равно количеству бит, которые имеют различные значения у двух векторов, между которыми вычисляется расстояние.

**Аппаратные компоненты** Схема распределенной архитектуры обработки данных в системе *Sensar...Secure<sup>TM</sup>* показана на рис. 16.6. Система содержит четыре основных блока: (1) компьютер общего назначения, который обеспечивает интерфейс с приложением, блоком управления камерами и блоком обработки видеоданных; (2) оптическая платформа с тремя камерами, две из которых имеют широкоугольное поле зрения (wide field of view, WFOV), а одна — узкоугольное поле зрения, которое будем называть «ближним» (near field of view, NFOV); (3) блок управления оптической платформой; и (4) блок обработки видеоданных, который содержит специальное аппаратное обеспечение для обработки стереоскопических видеоданных в реальном времени.

Две камеры с широкоугольными полями зрения используются для получения видеоданных, с помощью которых выполняется обнаружение ближайшего человека в поле зрения системы. Видеопотоки с двух этих камер поступают на вход блока обработки видеоданных. В этом блоке установлен цифровой сигнальный процессор (DSP), выполняющий обработку стереоскопических данных на основе пирамидальной обработки с переменным разрешением.



WFOV означает Широкоугольное поле зрения  
NFOV означает Ближнее поле зрения

**Рис. 16.6.** Распределенная архитектура обработки данных в системе *Sensar...Secure<sup>TM</sup>*.

Координаты  $x$ - $y$ - $z$  обнаруженного глаза человека передаются в главный блок, который затем использует эту информацию для управления оптической платформой, чтобы получить изображение глаза в ближнем поле зрения. Этот цикл обработки может выполняться с частотой два раза в секунду. Таким образом, система способна следить за медленно движущимся человеком. Изображения из ближнего поля зрения обрабатываются главным блоком для обнаружения заданной области глаза и построения кода радужной оболочки. Общая схема обработки приведена в алгоритме 16.2.

Следует отметить, что стоимость аппаратуры для сканирования радужной оболочки выше по сравнению с устройствами ввода данных в других системах идентификации. Это ограничивает сферу применения рассматриваемой системы. Высокая стоимость во многом определяется тем, что система идентификации является пассивной по отношению к пользователю. Пользователь может свободно перемещаться в рабочем пространстве системы, при этом система должна быть способна определять местоположение пользователя. Поэтому оказывается необходимо использовать камеры с широкоугольным

#### **Вычисление идентификатора личности (ID) для ближайшего человека в наблюдаемой сцене.**

1. Видеоданные от камер с широкоугольными полями зрения подвергаются корреляционной стереоскопической обработке. В результате определяется местоположение головы ближайшего человека.
2. С использованием шаблонов устанавливается местоположение признаков лица. Затем выполняется обнаружение левого (или правого) глаза и вычисляются его координаты  $(x, y, z)$ .
3. Координаты глаза  $(x, y, z)$  используются для управления монохромной камерой с ближним полем зрения. С ее помощью выполняется ввод изображения  $I$ , на котором глаз человека располагается в центральной части и находится в фокусе камеры.
4. По данным изображения  $I$  строится 2048-битный (в памяти занимает 256 байт) код  $Q$ , служащий представлением радужной оболочки глаза. На этом этапе используется запатентованное программное обеспечение обработки изображений.
5. Код радужной оболочки  $Q$  сопоставляется с кодами, хранящимися в базе данных. Для сопоставления используется операция ИСКЛЮЧАЮЩЕЕ ИЛИ (XOR). Если два кода отличаются менее чем на  $K$  бит, то возвращается идентификатор личности человека, иначе возвращается результат «отказать».

**Алгоритм 16.2.** Идентификация личности человека, данные о котором хранятся в базе данных, с помощью системы сканирования радужной оболочки глаза.



**Рис. 16.7.** Графическое представление 2048-битного кода, значения битов которого определяются знаками результатов применения фильтров Габора различных размеров к областям внутри 8 полос образа радужной оболочки, показанных на рис. 16.5. (С любезного разрешения John Daugman, Кембриджский университет.)

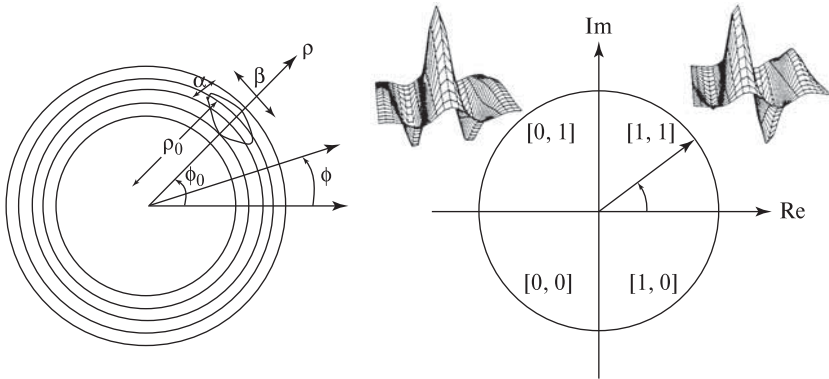
полем зрения для первоначального обнаружения объекта, который затем анализируется по данным изображения из ближнего поля зрения. Ближнее поле зрения позволяет получать изображения глаз, пригодные для анализа текстуры радужной оболочки. Для обработки стереоскопических данных в реальном времени требуется специальное оборудование. Пирамидальная обработка с использованием двух градаций разрешения ускоряет вычисления для корреляционного определения соответствующих точек в двух видеопотоках.

**Представление** Окончательное представление радужной оболочки глаза (т. е. личности человека для идентификации) имеет простой вид бинарного вектора размерности 2048 (для хранения такого вектора требуется 256 байт). Графическое представление одного из таких векторов показано на рис. 16.7: значения 0 на этом рисунке показаны черным цветом, а значения 1 — белым цветом. Каждый бит кода определяется знаком значения корреляции заданного фильтра Габора с заданной окрестностью изображения радужной оболочки. Поэтому очень важно, чтобы изображение глаза перед вычислением корреляции было нормировано относительно поворота.

Как показано на рис. 16.8, значение каждого бита кода радужной оболочки определяется путем вычисления корреляции двумерного вейвлета Габора с изображением радужной оболочки в окрестности  $(\rho_0, \phi_0)$ . В качестве параметров вейвлета задаются значения ширины (демодуляции)  $\alpha$  и  $\beta$ . Поперечное сечение вейвлета вдоль направления  $\rho$  имеет вид гауссиана со среднеквадратическим отклонением  $\alpha$ , а поперечное сечение вдоль направления  $\phi$  является гауссианом со среднеквадратическим отклонением  $\beta$ , модулированного синусоидой. Результат вычисления корреляции вейвлета с функцией изображения в некоторой окрестности точки  $(\rho_0, \phi_0)$  записывается в виде комплексного числа  $c$ :

$$c = \int_{\rho} \int_{\phi} f(\rho, \phi) [ e^{-j2\pi(\phi-\phi_0)} e^{-(\rho-\rho_0)^2/\alpha^2} e^{-(\phi-\phi_0)^2/\beta^2} ] \rho d\rho d\phi \quad (16.2)$$

Комплексное корреляционное значение преобразуется в два бита кода радужной оболочки посредством проверки знака: *если*  $(Re(c) \geq 0.0)$   $b_{real} = 1$  *иначе*  $b_{real} = 0$  *и если*  $(Im(c) \geq 0.0)$   $b_{img} = 1$  *иначе*  $b_{img} = 0$ . Очевидно, любой



**Рис. 16.8.** (Слева) схема кольцевых областей радужной оболочки, задаваемых в виде диапазона значений радиуса  $\rho_a \leq \rho \leq \rho_b$ , и позиционирование вейвлета Габора с параметрами ширины  $\alpha$  и  $\beta$  в окрестности изображения  $(\phi_0, \rho_0)$ . (Справа) форма комплекснозначного двумерного вейвлета Габора. (С любезного разрешения John Daugman, Кембриджский университет.)

поворот изображения радужной оболочки вокруг оптической оси камеры влияет на параметр местоположения  $\phi_0$ . В рассматриваемой системе углы поворота могут принимать только небольшие значения, так как при вводе изображения в ближнем поле зрения учитывается информация о положении обоих глаз на изображениях в широкоугольных полях зрения. В процессе сопоставления выполняется обработка нескольких вариантов кода радужной оболочки, соответствующих небольшим поворотам ее изображения. В качестве результата сопоставления выбирается наилучшее совпадение одного из этих «повернутых» кодов с любым кодом из базы данных системы. Ось  $\rho$  задается с учетом положения границы зрачка и внешней границы радужной оболочки. Считается, что эти границы имеют круговую форму, но их центры не обязательно должны совпадать. Эти границы обнаруживаются на основе анализа данных о краях изображения. Применяется метод, аналогичный преобразованию Хафа для обнаружения окружностей. Границы задаются в виде двух множеств параметров  $\rho, x_c, y_c$ , которым соответствует максимальное значение модуля градиента вдоль кругового контура.

$$\max_{(\rho, x_0, y_0)} \left| \frac{\partial}{\partial \rho} \oint \frac{f(x, y)}{2\pi\rho} ds \right| \quad (16.3)$$

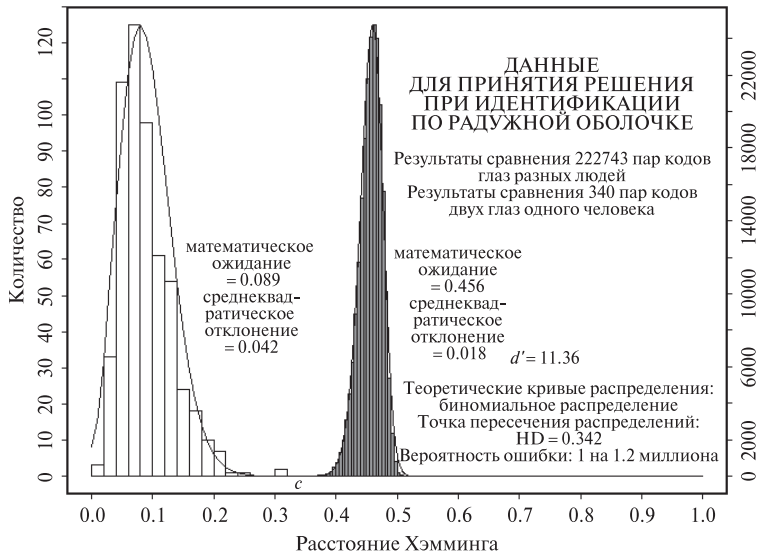
**Процедура идентификации** Схема процедуры идентификации представлена в виде алгоритма 16.2. Выше уже были рассмотрены все основные особенности этой процедуры. Замечания относительно производительности системы приведены в следующем разделе.

### 16.2.3 Производительность системы

В зависимости от условий наблюдения, для получения и идентификации радужной оболочки системе обычно требуется от одной до пяти секунд. Такое время обработки удовлетворительно для системы, предназначенной для использования в банкоматах, но может оказаться слишком большим для идентификации людей, проходящих через существующие системы безопасности (например, в аэропортах). При использовании в банкоматах фактором, ограничивающим быстродействие, является механическое управление оптической платформой. Порядка 90% времени система тратит на получение изображения. После передачи изображения на программную обработку, для обнаружения границ радужной оболочки и генерации ее кода требуется примерно 200 мс. Сопоставление кодов выполняется со скоростью, позволяющей обрабатывать данные примерно о 100000 людей в секунду.

Важной характеристикой системы является вероятность ошибки идентификации. Дагманом [Daugman, 1998] на основе теоретической модели, построенной с учетом большого количества тестовых результатов, были получены следующие оценочные значения коэффициентов ошибок. Допустим, для успешного подтверждения личности требуется совпадение 70% из 2048 бит кода радужной оболочки. Тогда ложное подтверждение возможно в 1 случае из  $6 \times 10^9$ , а вероятность ложного отказа составляет 1 на 46000. Если уменьшить порог соответствия битов до 66 процентов, то коэффициенты ложного подтверждения и ложного отказа будут примерно равны 1 на 1 миллион попыток.

В заключение сделаем несколько замечаний относительно модели, использовавшейся для получения приведенных выше оценочных значений вероятностей (подробности читатель может найти в статье Дагмана (Daugman, 1998)). Было проведено сравнение всех возможных пар кодов радужных оболочек примерно 300 людей. Полученные результаты показаны на рис. 16.9. Было обнаружено, что (а) при сравнении кодов разных глаз распределение расстояний Хэмминга (более чем для 200000 пар) лежит в интервале от 0.4 до 0.6 бит; (б) наблюдаемое распределение очень хорошо описывается биномиальным распределением с  $N = 266$  степенями свободы и  $p = 0.5 = q$ ; и (в) вероятно, неожиданным является то, что распределение того же типа было получено при сравнении кодов для левого и правого глаза одного и того же человека. Это подтверждает тот факт, что текстуры радужных оболочек двух глаз одного человека некоррелированы друг с другом, так же, как и текстуры глаз разных людей. На рис. 16.9 представлено распределение расстояний Хэмминга между кодами различных людей (справа) и распределение расстояний Хэмминга между кодами глаз одного и того же человека (слева). Эти распределения пересекаются в точке, соответствующей вероятности порядка  $10^{-6}$ . Порог для принятия решения необязательно выбирать в точке пересечения. Если допустимым считать расстояние, не превышающее 30 процентов битов вектора, то вероятность некорректного сопоставления составит примерно 1 на 6 миллиардов. Выбор порога подобным образом может быть желателен в случаях, когда



**Рис. 16.9.** Распределения значений расстояния Хэмминга между кодами глаз одного и того же человека (*слева*) и глаз разных людей (*справа*). Точка пересечения распределений соответствует доле 0.34 от количества битов кода. При этом вероятность отклонения корректного результата сопоставления равна вероятности принятия некорректного результата (значения обеих вероятностей равны примерно  $10^{-6}$ ). (С любезного разрешения J. Daugman, Кембриджский университет.)

стоимость ложного подтверждения намного превышает стоимость ложного отказа.

### 16.3 ЛИТЕРАТУРА

1. Bolle, R., J. Connell, N. Haas, R. Mohan, and G. Taubin. 1996. VeggieVision: a produce recognition system. *Proc. IEEE Workshop on Applications of Comput. Vision*.
2. Camus, T., U. M. Cahn von Seelen, G. G. Zhang, P. L. Venetianer, and M. Salganicoff. 1988. Sensor...Secure™ Iris Identification System, in *Proc. IEEE Workshop on Applications of Comput. Vision* (9–21 Oct. 1998), Princeton, NJ, 254–255.
3. Daugman, J. 1994. Biometric Personal Identification System Based on Iris Analysis. U.S. Patent No. 5,291,560 issued to John Daugman (March 1, 1994).
4. Daugman, J. 1998. Recognizing persons by their iris patterns. In *Biometrics: Personal Identification for a Networked Society*, A. Jain, R. Bolle, and S. Pankanti, eds. Kluwer Academic, Dordrecht, Netherlands and Norwell, MA.
5. DellaVecchia, M., T. Chmielewski, T. Camus, M. Salganicoff, and M. Negin. 1998. Methodology and apparatus for using the human iris as a robust biometric. *Ophthalmic Technologies VIII: SPIE Proc.* (Jan. 1998), 24–30.

---

# Аннотированный список литературы, добавленный при переводе

## 1 РАСПОЗНАВАНИЕ ОБРАЗОВ И КОМПЬЮТЕРНОЕ ЗРЕНИЕ

1. Дуда Р., Харт П., Распознавание образов и анализ сцен. М.: Мир, 1976.  
В этой классической книге рассматриваются вопросы, связанные с описанием изображений сцен и распознаванием образов. Среди этих вопросов — байесовская классификация, сопоставление с образцами, калибровка камеры и восприятие трехмерных сцен.
2. Розенфельд А. Распознавание и обработка изображений. М.: Мир, 1972.  
Классическая книга, представляющая собой одно из первых систематических изложений методов обработки изображений с целью извлечения из них информации о присутствующих на изображениях образах объектов.
3. Эксперименты с машинным зрением / В. П. Андреев, Д. А. Белов, Г. Г. Вайнштейн, Е. А. Москвина. М.: Наука, 1987.  
Обсуждаются вопросы формирования изображений трехмерных сцен и распознавания объектов. Описываются результаты экспериментов с лабораторной системой машинного зрения для распознавания объектов в виде многогранных блоков с плоскими гранями.

## 2 ПРЕДСТАВЛЕНИЕ И ОБРАБОТКА ИЗОБРАЖЕНИЙ

1. Введение в контурный анализ и его приложения к обработке изображений и сигналов / Я. А. Фурман и др. М.: ФИЗМАТЛИТ, 2002.  
В книге описываются математические методы представления контуров на изображениях и применение контурной информации в задачах обработки изображений, распознавания объектов и обработки сигналов.
2. Линдли К. Практическая обработка изображений на языке Си. М.: Мир, 1996.  
Рассмотрено устройство специфического фреймграббера для IBM-совместимого ПК. Описаны несколько файловых форматов изображений. Приведен ряд элементарных алгоритмов обработки изображений. Среди них пороговая бинаризация, масочная фильтрация, детектирование движения на основе вычитания фона.

3. Мюррей Д. Д., ван Райпер У. Энциклопедия форматов графических файлов. Киев: BHV, 1997.

В книге содержатся спецификации большого количества форматов графических файлов. Приводится ряд программ и описаны некоторые проблемы, возникающие при использовании файловых форматов в программах.

4. Прэт У. Цифровая обработка изображений. В 2-х книгах. М.: Мир, 1982.

Книга посвящена обработке двумерных изображений. Рассматриваются методы как на основе функции изображения, так и на основе обработки дискретных пиксельных массивов. Описываются различные способы представления изображений, в т. ч. телевизионные стандарты. Содержит часть, посвященную анализу изображений. В ней описаны некоторые приемы выделения характерных признаков, методы структурного описания содержимого изображений и ряд методов обнаружения объектов.

5. Яншин В. В., Калинин Г. А. Обработка изображений на языке Си для IBM PC: алгоритмы и программы. М.: Мир, 1994.

Описан ряд алгоритмов растровой графики и обработки изображений: вычисление статистических характеристик изображений, аффинные преобразования на плоскости и фильтрация изображений с различными масками.

### 3 КОМПЬЮТЕРНАЯ ГРАФИКА И ГЕОМЕТРИЧЕСКОЕ МОДЕЛИРОВАНИЕ

1. Голованов Н. Н. Геометрическое моделирование. М.: ФИЗМАТЛИТ, 2002.

Представлены математические методы описания и обработки трехмерных твердотельных объектов, применяемые в системах автоматизированного проектирования. Рассматриваются основные операции геометрического моделирования, предназначенные для построения твердотельных моделей. Представленные методы и алгоритмы используются в известной отечественной САПР «КОМПАС-3D».

2. Роджерс Д., Алгоритмические основы машинной графики. М.: Мир, 1989.

В книге рассмотрено большое количество алгоритмов растровой графики. Алгоритмы приведены на псевдоязыке, похожем на типичный процедурный язык программирования. Описан ряд моделей освещения.

3. Роджерс Д., Адамс Дж. Математические основы машинной графики. М.: Мир, 2001.

В книге приведено описание двумерных и трехмерных пространственных преобразований, в т. ч. различные виды проекций. Рассматриваются математические методы представления плоских и пространственных кривых, а также различные способы описания поверхностей.

4. Фокс А., Пратт М. Вычислительная геометрия. Применение в проектировании и на производстве. М.: Мир, 1982.



Книга посвящена вопросам описания и конструирования плоских и пространственных кривых и поверхностей.

5. Фоли Дж., взн Дэм А. Основы интерактивной машинной графики. В 2-х т. М.: Мир, 1985.

Классическая книга по машинной графике. Рассматриваются вопросы векторной и растровой графики. Описывается построение изображений трехмерных сцен. Подробно рассмотрены геометрические преобразования в однородных координатах.

#### 4 СМЕЖНЫЕ ВОПРОСЫ

1. Богуславский А. А. Си++ и компьютерная графика. Лекции и практикум по программированию на Си++. М.: КомпьютерПресс, 2003. 352 с.

При разработке программного обеспечения систем компьютерного зрения часто используется язык программирования Си++. В этой книге рассмотрен основной материал, необходимый для разработки программ на Си++ для Intel-совместимых персональных компьютеров: основы процедурного программирования на Си++, введение в объектно-ориентированное программирование, использование библиотеки трехмерной компьютерной графики OpenGL и программирование для операционных систем семейства MS Windows.

2. Гаврилов П. Ф., Кривилев В.Н. Видеокамеры: принципы работы, схемотехника, регулировки. М.: Радиотон, 1999.

В книге описываются принципы устройства бытовых малогабаритных видеокамер. Описывается назначение основных функциональных узлов и операций, выполняемых этими узлами (в т. ч. устройство трансфокационного объектива, автоматическая фокусировка, стабилизация изображения и др.)

3. Каллан Р. Основные концепции нейронных сетей. М.: Издательский дом «Вильямс», 2001.

Книга вводного характера по нейронным сетям. Рассматриваются основные модели нейронных сетей и их применение в ряде задач, в т. ч. для классификации образов и кластеризации.

4. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. М.: БИНОМ. Лаборатория знаний, МЦНМО, 2004.

Учебник по курсу построения и анализа эффективных алгоритмов и структур данных. В нем рассматриваются большинство известных комбинаторных алгоритмов и структур данных.

---

# Предметный указатель

- Автокорреляция (autocorrelation), для описания текстур, 287–289
- автоматический выбор порога бинаризации (automatic thresholding), 119–124
- аддитивная цветовая система (additive color system), 249
- аквариум виртуальной реальности (fishtank virtual reality), 682
- аккумуляторный массив в преобразовании Хафа (accumulator array), 399–396
- алгоритмы,
  - выделение границ (borders finding), 382–385
  - выравнивание гистограммы (histogram equalization), 173
  - вычисление векторов перемещения для точек интереса (deriving motion vectors for interesting points), 338
  - вычисление выходного изображения по данным входного (computing output image from input image), 184–185
  - вычисление твердотельного преобразования для совмещения модельного треугольника с конгруэнтным треугольником в мировой системе координат (rigid transformation for aligning model triangle with congruent world triangle with), 538
  - вычисление трехмерных координат точек поверхности с помощью калиброванной камеры и одного осветителя (computing 3D surface coordinates using calibrated camera and projector), 559
- графы смежности областей (region adjacency graphs, RAG), 115
- жадный обменный алгоритм (Greedy Exchange), 346, 348
- инспекционный контроль часовых шестеренок (watch-gear inspection), 98–102
- использование данных о цвете и движении для отслеживания знаков языка жестов ASL (using color and motion to track ASL gestures), 415–418
- итерационное решение задачи РЗР (iterative P3P solution), 565
- классификация (classification), 136–140
- классическая кластеризация (classical clustering), 365
- классическая маркировка связанных компонент, (classical connected components), 90, 94
- кластеризация методом ISODATA (isodata clustering), 368
- кластеризация методом Ши (Shi's clustering procedure), 374
- кластеризация по математическому ожиданию (iterative K-means clustering), 366
- кластеризация по расположению (pose clustering), 445
- маркировка методом дискретной релаксации (discrete relaxation labeling), 461
- маркировка ребер объекта с плоскими гранями методом перебора с

- возвратом (labeling block edges via backtracking), 487
- маркировка ребер объектов с плоскими гранями методом дискретной релаксации (labeling block edges via discrete relaxation), 489
- метод базовых локальных признаков (local-feature-focus), 443
- обнаружение отрезков прямых (finding straight line segments), 397, 399
- обнаружение прямых методом О'Гормана и Кловса (O'Gorman and Clowes method for extracting straight lines), 399
- обратное распространение (back-propagation), 166
- обучение перцептрона (perceptron learning), 162–164
- объединение-поиск (union-find), 89
- объемное удаление (space-carving), 590–594
- один шаг циклического алгоритма перемещения активного контура (single update stage for active contour), 622
- подсчет отверстий (hole counting), 19
- поиск на дереве интерпретации (interpretation tree search), 457
- поиск точек интереса (detecting interesting image points), 336
- построение гистограммы изображения (image histogram), 118
- построчная маркировка (row-by-row labeling), 86
- предварительная обработка в методе геометрического хэширования (geometric hashing offline preprocessing), 450
- преобразование Хафа для обнаружения окружностей (Hough transform for finding circles), 400
- преобразование Хафа для обнаружения прямых (Hough transform for finding straight lines), 397
- преобразование цветовых координат из системы RGB в HSI (converting RGB encoding to HSI encoding), 255
- прослеживание контуров на бинарном краевом изображении (tracking edges of binary edge image), 392
- процедура принятия решений (decision procedure), 135
- распознавание на основе оценки реляционного расстояния (relational distance matching), 640
- распознавание объекта методом геометрического хэширования (geometric hashing online recognition), 452
- распознавание по внешнему виду объектов с использованием базиса главных компонент (recognition-by-appearance using basis of principal components), 659
- рекурсивная маркировка (recursive labeling), 87
- система для идентификации личности по радужной оболочке глаза (identification by iris-scanning system), 709
- совмещение (alignment), 630
- сопоставление границ (boundary matching), 309
- сопоставление графов (graph-matching), 143
- стереофотометрический метод с использованием трех источников света (photometric stereo with three light sources), 601
- цилиндрическая деформация фрагмента изображения (cylindrical warp of image region), 471
- этапы идентификации продуктов в системе Veggie Vision, 700
- альbedo (albedo), 264
- аналитические выражения параметров моделей (closed form solutions for parameters), 405
- аналоговое изображение (analog image), определение, 48
- аффинная калибровочная матрица (affine calibration matrix), 549–557
- аффинные деформации (affine warp), 431

- аффинные преобразования (affine transformations), 436, 438
- для распознавания двумерных объектов, 439–451
  - трехмерные, 530
- аффинный метод (affine method), 555
- Базис, ортогональный, 210**
- базис Робертса (Roberts basis), 213
- базисные изображения (basis images):
- вычисление, 656
  - для множества эталонных изображений, 656–661
- базовых локальных признаков метод (local-feature-focus method) распознавания объектов, 442
- базы данных (databases):
- изображений, 16, 294
  - изображений, запросы (queries), 295–298
  - организация (organizing), 318–324
  - система QBIC (Query by Image Content), 294
- Байесовская процедура принятия решений (Bayesian decision-making), 152–156
- Байесовский классификатор (Bayesian classifier), определение, 153
- белый свет (white light), определение, 247
- бинарная морфология (binary morphology), 93–103
- для обработки медицинских изображений, 99
  - основные операции, 95–98
  - приложения, 98–101
  - структурирующие элементы (structuring elements), 93–95, 98–103
  - условное наращивание (conditional dilation), 103
- бинарные деревья решений (binary decision trees), 145
- бинарные изображения (binary images), 40
- анализ, 78–125
  - групповое кодирование, 58
  - замыкание (closing), 95, 97
  - маркировка связанных компонент (labeling connected components), 84–92
  - морфология (morphology), 93–103
  - наращивание (dilation), 95, 96
  - определение, 49
  - перенос (translation), 95
  - размыкание (opening), 95, 97
  - эрозия (erosion), 95, 96
- ближайшие соседи (nearest neighbors), использование для классификации, 139
- близости примитив (proximity primitive), 652
- блик (highlight), определение, 268
- блуминг (blooming), 46
- буфер глубины (z-buffer), 686
- буфер кадра (frame buffer), 40
- быстрая сортировка (quicksort), модифицированная, 183
- быстрое преобразование Фурье (fast Fourier transform), 238
- Вейвлеты (wavelets), 238**
- вектор характерных признаков (feature vector), 136
- векторное пространство (vector space), 208, 213
- всех сигналов, 208
- векторы (vectors):
- перемещения (motion), 331, 414
  - перемещения, для точек интереса, 338
  - признаков (feature), 136
  - текстурного описания (texture description), 303
- векторы перемещения (motion vectors):
- вычисление для точек интереса, 338
  - вычисление, 331–345
- верификация (verification):
- и оптимизация при вычислении положения объектов, 583
  - определение, 127
- вершина (vertex), модели типа «поверхность–ребро–вершина», 609–612
- видео (video):
- алгоритм MPEG (Motion Picture Experts Group) для сжатия, 339
  - камеры (cameras), 43
  - обнаружение изменений условий съемки, 353–359

- сегментация
    - видеопоследовательностей, 354
  - формат MPEG, 62
  - хранение видеопоследовательностей, 359
  - визуальное событие (visual event), 618
  - виртуальная реальность (virtual reality):
    - аквариум (fishtank), 682
    - определение, 677
    - устройства, 677–682
  - виртуальные прямые (virtual lines):
    - извлечение дальностных признаков, 504
    - определение, 504
  - внешние параметры камеры (extrinsic camera parameters), 568
  - внешние углы (external corners), 17
  - внешняя энергия (external energy), 623
  - внутренние изображения (intrinsic images), 477
  - внутренние параметры камеры (intrinsic camera parameters), 567
  - внутренние углы (internal corners), 17
  - внутригрупповая дисперсия (within-group variance), 120
  - воксели (voxels), 615
  - Вороного многоугольники (Voronoi polygons), 279
  - восприятие (perception):
    - освещенных объектов, 246
    - света, 38
    - структуры объектов по данным о движении, 600–603
    - учет освещенности, 271
    - человеческое цветовое, 272
  - восстановление изображений (image restoration), 171
  - восьмисвязная окрестность (8-neighbors), 80
  - временная избыточность (temporal redundancy), 62
  - второго порядка (second-order):
    - момент по столбцу изображения (column moment), 109
    - момент по строке изображения (row moment), 109
    - смешанный момент (mixed), 109
  - выбросов класс (reject class),
    - определение, 128
    - выбросы (outliers), 407
  - выделение характерных признаков (feature extraction), 631
  - выравнивание гистограммы (histogram equalization), 173
  - высокая размерность (high dimensionality), 409
  - высокий контраст (high contrast),
    - обнаружение, 186
  - вытеснение (wipe), 353
  - вычисление формы по освещенности (shape-from-shading), определение, 595
  - вычитание изображений (image subtraction), 26, 329
- Гамма-коррекция (gamma correction), 172**
- гармонические функции, применение для частотного анализа, 225–240
  - Гауссиан, свойства (properties of Gaussian), 199
  - Гаусса
    - маски, 190
    - сглаживание (Gaussian smoothing), 205
    - фильтр (Gaussian filter), 181
    - фильтрация, при обнаружении перепадов краёв, 196–207
    - функция (Gaussian function), 199
    - шум (Gaussian noise), 179, 407
  - геометрические искажения (geometric distortion) изображений, 45
  - геометрические модели (geometric models), сопоставление посредством совмещения, 629–639
  - геометрическое хэширование (geometric hashing), 447
  - геоны (geons), объемные примитивы, 640
  - гистограммы (histograms):
    - выравнивание (equalization), 173–176
    - использование в методах кластеризации, 368
    - использование в рекурсивном гистограммном методе Оландера 369–371
    - использование для выбора порога бинаризации, 117
    - определение, 117

- поиск моды (mode seeking), 369
- сравнение, 356
- текстурные (texture), 305
- формы (shape), 307
- цветové, 253, 300, 302
- главная ось, 111
- глубина (depth):
  - определение с помощью фокусировки (depth from focus), 503
  - признаки (cues), 65
  - трехмерные дальностные признаки на двумерных изображениях (3D cues in 2D images), 491–497
- человеческое восприятие (human perception), 505
- глубина поля зрения (depth of field), 503
  - определение, 520
  - связь с фокусом, 519
- глубины восприятие, с помощью стереосистемы, 509–517
- головной дисплей (head-mounted display, HMD), 678
- голубой–пурпурный–желтый, субтрактивная цветовая система CMY (cyan-magenta-yellow, CMY subtractive color system), 252
- границы (boundaries):
  - движущихся объектов, 414
  - извлечение признаков, 504
  - кодирование, 379
  - обнаружение, 382
  - определение формы, 499
  - освещенности, 480
  - сопоставление (matching), 309
- графов разбиение в методе кластеризации Ши, 371–375
- графы (graphs):
  - ракурса (aspect), 618
  - смежности областей (region adjacency), 114
- группировка согласующихся соседних контурных фрагментов в кривые (aggregating consistent neighboring edges into curves) 387
- групповое кодирование (run-length encoding):
  - бинарных изображений, 58
  - при маркировке связанных компонент, 91
- Дальномер лазерный (LIDAR), 74
- дальномеры (range scanners), 74–76
- дальнометрические (range):
  - данные (data), 588, 590
  - изображения (images), 74–76
  - устройства, 74–76
- данные (data):
  - дальнометрические (range), 588, 590, 591
  - изображения (image data), 56
  - информационная перчатка (data glove), 676
  - многомерные (multidimensional), принятие решений, 156
  - сбор при реконструкции трехмерных объектов (acquisition in 3D object reconstruction), 585
  - сжатие (compression), 57
- датчик/преобразователь (sensor/transducer), 128
- датчики, (sensors), 69–76
  - в системах виртуальной реальности, 682
  - LIDAR (light detection and range), 74
  - мультиспектральные (multispectral), 70
- движение (motion):
  - в системах виртуальной реальности, 684
  - вычисление траекторий движущихся точек, 345–353
  - использование для оценки геометрической структуры объектов, 600–603
  - моделирование человеческого сердца, 626
  - на последовательностях двумерных изображений, 327–361
  - накопление траекторий (aggregating motion trajectories), 415
  - связанные с движением признаки, 504
  - согласованное (coherence), для сегментации, 414–418
  - явления и прикладные задачи, 328
- двоичное разбиение (binary partition), 282
  - локальное, 282
- двумерное (2D):
  - оценка положения на основе соответствия 2D–3D, 578

- совмещение данных, 422
- сопоставление, 421–475
- двумерные изображения (2D images), 37
- восприятие трехмерных сцен, 476–525
- использование свойств движения, 327–361
- обнаружение трехмерных признаков, 491–497
- обработка с использованием дифференциальных операторов, 190–196
- оценка пространственных свойств объектов, 65
- типы изображений, 47
- двумерные модели (2D models), совмещение типа 2D-3D, 633
- двумерные преобразования (2D transformations), определение, 422
- двумерные функции интенсивности (2D picture functions), 230
- действительная система координат изображения (real image coordinate frame), 69
- дерево категорий, система GRUFF (category definition tree), 652
- дерево решений (decision tree), 134, 144
- автоматическое построение (automatic construction), 146
- бинарное (binary), определение, 145
- узлы (nodes), 145
- деревья интерпретации (interpretation trees, IT), 454
- для контурных изображений, 488
- детектор краев Кэнни (Canny edge detector), 207
- и компоновщик (linker), 386
- деформации (warping):
- изображений, 26
- нелинейные (nonlinear), 469
- деформационные модели (deformable models), 619
- деформация, аффинная (affine warp), 431
- динамическая пороговая бинаризация (dynamic thresholding), 123
- дискретизация (quantization):
- изображений, 50
- эффекты, 47
- дискретизация изображения (image quantization), 50
- дискретная релаксация (discrete relaxation):
- маркировка (labeling), 453, 481
- маркировка ребер объекта с плоскими гранями (labeling block edges), 489
- дискретное преобразование Фурье (discrete Fourier transform, DFT), 234
- дисперсия, 46
- дисторсия (distortion):
- геометрическая (geometric), 45
- радиальная (radial), 471
- хроматическая (chromatic), 47
- дифференциальные маски, 186–190
- для обнаружения краев, 186–196
- свойства, 189
- дифференциальные операторы для двумерных изображений (difference operators for 2D images), 190
- дифференцирование одномерных сигналов, 186
- диффузное отражение (diffuse reflection), 266
- дополненная реальность (mixed reality), 671
- определение, 677
- дочерние узлы (child nodes), 145
- дуги окружностей, обнаружение с помощью преобразования Хафа, 393
- Евклидово расстояние (Euclidean distance):**
- определение, 136
- шкалированное (scaled), 139
- Жадный обменный алгоритм (Greedy Exchange algorithm), 346, 348, 351**
- жесткие преобразования (rigid transformations), 428
- жесткое ограничение (hard constraint), 621
- жесткость (stiffness), 626
- Заголовок файла изображения (file header), 56**
- задача поиска документов (document retrieval, DR), 133
- залив (bay), 140, 150



- закрывание бинарных изображений (closing of binary images), 97
- запаздывания теорема (shift theorem), 239
- запрос по образцу (query-by-example, QBE), 297
- запросы (queries), в базах данных изображений, 295–297
- звуковая информация, в системах виртуальной реальности, 683
- зеркальное отражение (specular reflection), 265
- знак (mark), определение, 480
- зона обзора, взаимосвязь с ошибками (coverage versus errors), 517
- зрение, стереоскопическое, 509–517
- Игра Decathlete, 333**
- идентификация (identification):
- личности человека по радужной оболочке глаза, 704
  - требования к системам идентификации, 705
- иерархия категорий (category hierarchy), система GRUFF, 652
- излучение (radiation) одиночного источника света, 264
- изменения, обнаружение в видеопоследовательностях, 353
- измерения пространственные (spatial measurement) и дискретизация изображений (image quantization), 50
- изображений повышение качества (image enhancement):
- выравнивание гистограммы (histogram equalization), 173
  - гауссовская фильтрация и ЛОГ-фильтрация для обнаружения краев (Gaussian filtering and log edge detection), 196
  - изменение тонового распределения (gray-level mapping), 172
  - медианная фильтрация (median filtering), 181
  - обнаружение краев с использованием дифференциальных масок (detecting edges using differencing masks), 186
  - определение, 171
  - свертка (convolution) и кросс-корреляция (cross correlation), 219
  - сглаживание изображения (image smoothing), 179
  - удаление малых областей изображения (removal of small image regions), 176
- изображений представление (image representation), 37–77
- изображений сегментация (image segmentation), 362–420
- обнаружение областей (identifying regions), 363
- изображений файловые форматы:
- GIF (Graphics Interchange Format), 60
  - TIFF (Tag Image File Format) 60
  - сравнение, 63
- изображения (images):
- аналоговые (analog), определение, 48
  - базисные (basis), вычисление, 657
  - базисные (basis), для набора эталонных изображений, 656
  - бинарные (binary), анализ, 78–125
  - бинарные (binary), определение, 49
  - внутренние (intrinsic), 477
  - вычисление выходного изображения по данным входного изображения, 184
  - вычисление характерных признаков (features), 27
  - групповое кодирование (run-coded) бинарных, 58
  - дальнометрические (range), 74
  - двумерные, возможные трехмерные признаки, 491
  - двумерные, обработка дифференциальными операторами, 190
  - двумерные, определение движений на последовательностях, 327–361
  - двумерные, оценка пространственных свойств объектов, 65
  - использование для восприятия трехмерных сцен, 476–525
  - классификационные (thematic), 49, 273
  - комбинация нескольких, 26
  - маркированные (labeled), 378



- маркированные (labeled),  
определение, 49
- многоспектральные (multispectral),  
определение, 49
- повышение качества (enhancing), 26
- подсчет объектов (counting objects),  
82
- поиск на основе содержания  
(retrieving content-based), 293–326
- поиск на основе содержания, с  
использованием различных мер  
расстояния, 322
- полутоновые (gray-scale),  
определение, 49
- полутоновые (gray-scale), пороговая  
бинаризация, 116
- применение масок, 80
- псевдоцветовые (pseudo-colored), 50
- сбор данных (acquisition), 585
- сглаживание (smoothing), 179
- сопоставление (matching) в  
двумерном пространстве, 421–475
- типы, 47
- улучшение (improving), 170
- фильтрация (filtering) и повышение  
качества (enhancing), 169–243
- функции интенсивности (picture  
functions), 47
- цветные (color), 70
- цифровые (digital), определение, 48
- цифровые (digital), проблемы  
формирования; 45
- цифровые (digital), форматы, 55
- См. также* Трехмерные изображения,  
Двумерные изображения
- изображения гистограмма (image  
histogram), 117
- изображения плоскость, передняя  
(front image plane), 507
- изображения файловый заголовок (file  
header), 56
- инвариантные признаки (invariant  
features), 29
- индексация (indexing):
- для систем поиска изображений на  
основе содержания, 322
- на основе В+-деревьев (В+-tree  
indexes), 320
- реляционная (relational), 467, 639
- индексы (indexes):
- В+-деревья (В+-tree), 320
- К-мерные деревья (K-d tree), 322
- R-деревья (R-tree), 322
- пространственные (spatial), 322
- стандартные (standard), 318
- хэшированные (hash), 319
- индексы на основе деревьев (tree  
indexes):
- В+-деревья, 320
- К-мерные деревья, 322
- R-деревья, 322
- инкапсулированный PostScript  
(encapsulated postscript, EPS  
format), 62
- инспекционный контроль поперечных  
балок (crossbar inspection), 17
- интегральный учет пространственных  
ограничений (spatial constraints),  
599
- интенсивность (intensity):
- изменение тонового распределения  
(gray-level mapping), 172
- значения (values), 71
- интерактивная сегментация (interactive  
segmentation) анатомических  
структур, 670
- искусственная среда (synthetic  
environment, SE), 677
- См. также* Виртуальная реальность
- искусственные изображения, 687–691
- искусственные нейронные сети  
(artificial neural network, ANN), 159
- искусственные нейроны (artificial  
neurons, AN), 160
- искусственных и реальных  
изображений сочетание, 687
- истинные трехмерные модели (true 3D  
models), в сравнении с моделями на  
основе классов видимости  
(view-class models), 618
- Калибровка (calibration):**
- камер, 549
- камер, пример, 573
- камер, улучшенный метод, 567
- наилучшая аффинная калибровочная  
матрица (best affine calibration  
matrix), 549
- направленных источников света, 557
- камера:

- видео, 43
- внешние параметры (extrinsic parameters), 568
- внутренние параметры (intrinsic parameters), 567
- вычисление координат трехмерных точек с помощью нескольких камер, 547
- использование для сбора данных, 585
- калибровка (calibration), 549
- калибровка, улучшенная, 567
- модель человеческого глаза, 44
- ПЗС, прибор с зарядовой связью (CCD, charge-coupled device), 39
- пример калибровки, 573
- расположение в стереоскопической системе (posing for stereo configuration), 527–530
- формирование изображений, 41
- капли (blobs), 640
- каркасные модели (mesh):
  - оболочечные (balloon), 624
  - разновидности, 608
  - регулярные (regular), 608
  - триангулированные (triangular), 609
- квадрантные деревья (quadtrees), 322
- для представления областей, 380
- классификатор (classifier), 130
  - определение, 128
  - реализация, 136
- классификационные изображения (thematic images), 50, 273
- классификация (classification):
  - алгоритм, 136–140
  - использование цветowych данных, 257
  - нечеткая (fuzzy), 165
  - определение, 128
  - по ближайшему среднему значению класса, 137
  - по расстоянию до ближайших соседей, 139
  - распространенные модели, 128
- См. также* Дерево решений
- классический алгоритм маркировки связанных компонент (classical connected components algorithm), с использованием структуры данных для объединения-поиска, 90, 94
- классы (classes), определение, 128
- классы видимости (view classes), 618
- кластеризация (clustering), 158
  - гистограммные методы (methods based on histograms), 368
  - итерационная по математическому ожиданию (iterative K-means) 366
  - классическая, алгоритмы, 365
  - метод ISODATA, 366
  - метод графового разбиения Ши (Shi's graph-partitioning technique), 371
  - методы, 364
  - по расположению (pose), 444
  - рекурсивный гистограммный метод Оландера (Ohlander's recursive histogram-based technique), 369
- кластеризация методом ISODATA, 366
- код, Фримана цепной (Freeman chain code), 379
- кодирование (encoding):
  - RGB (красный–синий–зеленый), преобразование в HSI (тон–насыщенность–интенсивность), 255
  - YUV, 257
  - границ, 379
- групповое кодирование (run-length), применение для маркировки связанных компонент, 91
- октантные деревья (octrees), 615
- коллекции изображений (image collections), 294
- коллизии (collision), 319
- компоновщик и детектор перепадов Кэнни (Canny edge detector and linker), 386
- компьютерное зрение (computer vision),
  - определение, 13
- контраст (contrast)
  - обнаружение, 186
  - увеличение, 173
- контролирующее наблюдение (surveillance), 330
- контроль отверстий в поперечных балках (crossbars inspection), 17
- контурные изображения (line drawings):
  - использование дерева интерпретации (interpretation tree), 488
  - объектов с плоскими гранями (blocks), 483
- контурные (contours):

- внутреннее изображение (intrinsic image), 477
- внутренняя энергия контура (internal contour energy), 623
- движущихся объектов (of moving objects), 414
- использование для обнаружения областей (identifying regions by), 382
- модели на основе активных контуров (active contour models), 620
- обнаружение с помощью детектора краев и компоновщика Кэнни (identifying with Canny edge detector and linker), 386
- обнаружение с помощью преобразования Хафа для прямых и дуг окружностей (detecting with Hough transform for lines and arcs), 393
- координаты, однородные (homogeneous coordinates), 425
- корреляция (correlation), 170
- край (edge):
  - гауссовская и ЛОГ-фильтрация для обнаружения перепадов, 196
  - группировка в кривые, 387
  - обнаружение в человеческой зрительной системе, 201
  - обнаружение с помощью дифференциальных масок, 186
  - обнаружение с помощью ЛОГ-фильтра, 200
  - плотность и направление при анализе текстур, 280
  - разрывный край (jump edge), 478
- красный–зеленый–синий (red-green-blue, RGB):
  - преобразование в систему HSI, 255
  - преобразование в систему YUV, 256
- цветовая система, 249
- кривая функционирования приемника (receiver operating curve, ROC), 132
- кривые (curves):
  - группировка согласующихся соседних контурных фрагментов в кривые (aggregating consistent neighboring edges into), 387
  - обнаружение с помощью преобразования Хафа (detecting with Hough transform), 393
  - сегментация с использованием методов аппроксимации (segmenting via fitting), 409
  - критерий максимальной ошибки (max-error criteria), определение, 405
  - кросс-корреляция (cross correlation), 513
  - нормированная, 223
  - определение, 221
  - свертка (convolution), 221
  - кругок рассеяния (circle of confusion), 41
  - крышки (lids), 140, 150
  - кубические элементы (cubes):
    - в алгоритме объемного удаления (space-carving algorithm), 591
    - в октантных деревьях (octrees), 615
- Лавса энергетические текстурные характеристики (Laws texture energy measures), 285**
- лазерные осветители для структурной подсветки, 559
- лазерный дальнометр LIDAR (light detection and range), 74
- Ламбертова модель отражения (Lambertian reflectance model), 594, 266
- лезвие (blade), 477
  - определение, 480
- ленты (ribbons), 410, 613
  - прямолинейные, обнаружение, 412
- лимба (limb), 480
- линейная оптимизация при наличии ограничений (constrained linear optimization), 580
- линейные преобразования, 426
- листовые узлы (leaf nodes), 145, 320
  - квадрантное дерево (quadtree), 380
- лица человеческие (faces),
  - идентификация, 261
  - обнаружение, 313
- логические функции И, ИЛИ и НЕ, 160
- логические характерные признаки (boolean features), 150
- ложные пропуски (false dismissals), 132

- ложные срабатывания (false alarms), 132
- локальное двоичное разбиение (local binary partition), 282
- Магниторезонансная ангиография (magnetic resonance angiography, MRA), 73
- магниторезонансные изображения (magnetic resonance imaging, MRI), 20, 73, 273
- макет, цветовой (color layout), 302
- малые компоненты (small components), удаление, 178
- малые области изображений (small image regions), удаление, 176
- маркированные изображения (labeled images), 378
- определение, 48
- маркировка:
  - контурных изображений плоскогогранных объектов, 483
  - кубических объемных элементов, 590
  - методом дискретной релаксации (discrete relaxation), 456
  - методом непрерывной релаксации (continuous relaxation), 459
  - ребер плоскогогранных объектов методом дискретной релаксации, 489
  - ребер плоскогогранных объектов методом перебора с возвратом (backtracking), 487
  - релаксационная для контурных изображений, 489
  - связанных компонент (connected components), 84
  - связанных компонент с использованием группового кодирования (run-length encoding), 91
  - согласованная (consistent), 453
- маркировки алгоритмы (labeling algorithms):
  - построчный (row-by-row), 86
  - рекурсивный (recursive), 86
- Марра-Хилдрет теория и ЛОГ-фильтрация, 204
- маски (masks), 177
  - гауссовского сглаживания (Gaussian smoothing), 190, 200
  - дифференциальные (differencing), применение обнаружения краев, 186
  - для аппроксимации производных (derivative masks), 186
  - для реализации ЛОГ-фильтра (LOG filter), 200, 201
  - Превитт (Prewitt), 191, 195, 398
  - применение для представления операций, 219
  - применение к изображениям, 80
  - Робертса (Roberts), 192
  - Собеля (Sobel), 192, 194
  - согласованных фильтров (matched filters), 207
  - усредняющего сглаживания (box smoothing), 190
- массивы:
  - PARENT, 88
  - ПЗС (CCD), 39, 44
  - пиксельный, 68, 70
- масштабирование (zooming), 331, 353
- масштабирования преобразование (scaling):
  - в трехмерном пространстве, 532
  - на плоскости, 425–432
  - перспективное (perspective), 493
- масштабирование камеры (camera zoom), 353
- математическая морфология (mathematical morphology), 92
- матрица (matrix):
  - вхождений (co-occurrence), 282
  - наилучшая для аффинной калибровки (best affine calibration), 549
  - неточностей (confusion matrix), 143
  - перспективной проекции (perspective transformation), 540
- машинное зрение (machine vision), definition, 13
- медиана (median):
  - определение, 182
  - фильтрация (filtering), 181
- меры расстояния между изображениями (image distance measures), 298

местоположение модельной точки (model point) и точки изображения (image point), 433

метка (label), 452

метод наименьших квадратов (method of least squares), 403

микроденситометр (microdensitometer), 70

мировая система координат (world coordinate frame), 69

многомерные данные (multidimensional data), принятие решений, 156

многослойные нейронные сети (multilayer network), 165

многоугольник Вороного (Voronoi polygon), 279

модели (models):

- геометрические (geometrical), 608, 629
- истинные трехмерные (true 3D) в сравнении с моделями на основе классов видимости (view-class), 618
- каркасные (mesh), 608
- на основе активных контуров (active contour), 620
- на основе обобщенных цилиндров (generalized-cylinder), 612
- оболочечные (balloon), 624
- параметрические (parametric), 154
- перцептрона (perceptron), 160
- подбор для контурных сегментов (fitting to segments), 403
- получения изображений на основе перспективной проекции (perspective imaging), 506
- проволочные каркасные (wire-frame), 609
- реляционные (relational), 639
- реляционные на основе классов видимости (view-class relational), 642
- совмещение (alignment) типа 2D-3D, 633
- типа «поверхность-ребро-вершина» (surface-edge-vertex), 609
- трехмерные, и сопоставление, 607–666
- физические (physics-based) и деформационные (deformable), 619
- функциональные (functional), 651

*См. также* Трехмерные модели моделирование полетов (flight simulation), 670

модель камеры (camera model), 539

- параметры, 555

момент второго порядка (second moment):

- минимальный, 114
- относительно оси, 112

морфология (morphology), бинарных изображений, 93

мультиспектральные (multispectral):

- датчики, 70
- изображения, 70
- определение, 49

**Назначение (assignment), 452**

наиболее выразительные признаки (most expressive features, MEF), 661

наиболее различимые признаки (most discriminating features, MDF), 661

наилучшая аффинная калибровочная матрица (best affine calibration matrix), 549

наименьших квадратов (least-squares):

- критерий, определение, 404
- метод, 403
- пример задачи, 550

наклон (tilt), определение, 498

накопление траекторий движения (aggregating motion trajectories), 415

наплыв (dissolve), 354

направление и плотность краев в анализе текстур (direction and density of edges), 280

наращивание (dilation):

- бинарных изображений (binary images), 95
- бинарных изображений, определение, 96
- условное (conditional), 103

неграфические описания (non-iconic representations) изображений, 28

независимые тестовые данные (independent test data), определение, 131

нейронные сети (artificial neural network, ANN), 159

- применение для генерации полос Маха (Mach band effect), 202

нейроны (neurons), 159

- искусственные (artificial), 160

- Некера эффект (Necker phenomena), 490
- нелинейная оптимизация (nonlinear optimization), 409
- нелинейные деформации (nonlinear warping), 469
- немаксимальное подавление (nonmaximum suppression), 387
- непрерывная релаксация, маркировка (continuous relaxation labeling), 459
- неравенство Коши-Шварца, 212
- нечеткая классификация (fuzzy classification), 165
- низкоуровневые признаки (low-level features), обнаружение, 170
- номинальное разрешение (nominal resolution), определение, 50
- нормированное скалярное произведение (normalized dot product), 212
- нормированные (normalized) цветовые координаты RGB, 251
  
- О’Гормана (O’Gorman) и Кловса (Clowes) алгоритм, 395, 399**
- области (regions), 483
  - кодирование границ, 379
  - ленты (ribbons), 410
  - наращивание (growing), 375
  - обнаружение в результате сегментации, 363
  - обнаружение с использованием контуров, 382
  - оверлейные представления (overlays), 377
- обнаружение (detection):
  - краев в зрительной системе человека, 201
  - краев с использованием ЛОГ-фильтрации 200
- обнаружение областей (identifying regions):
  - классический алгоритм кластеризации, 365
  - методы кластеризации, 364
  - наращивание областей (region growing), 375
  - при сегментации изображений, 363–377
- обобщенные цилиндры, в моделях трехмерных объектов (generalized-cylinder models), 612
- обозначения (notations), 47–50
  - значений пикселей (pixel values), 40
- оболочечные модели (balloon models), трехмерные 624
- обработка изображений (image processing): 169–243
  - определение, 31
  - с использованием преобразований Фурье, 229
- обратное дискретное преобразование Фурье (inverse discrete Fourier transform, IDFT), 235
- обратное перспективное преобразование (inverse perspective), 560
- обратное проектирование (backprojection), 260
- обучение (learning):
  - супервизорное, в системе Veggio Vision, 702
  - без учителя (unsupervised), 158
  - машин, 158
  - супервизорное (supervised), 158
- объединение–поиск:
  - алгоритмы (union-find algorithms), 90
  - структура данных (union-find structure), 88
- объектив, 41
- См. также* Тонкой линзы формула
- объектная система координат (object coordinate frame), 68
- объекты с плоскими гранями:
  - маркировка контурных изображений (labeling of line drawings of blocks), 483
  - маркировка ребер методом дискретной релаксации (labeling edges of via discrete relaxation), 489
  - маркировка ребер методом перебора с возвратом (labeling edges of via backtracking), 487
- объемное удаление (space-carving), 590
- оверлейные представления (overlays), 377
- огораживания примитив (enclosure primitive), 652
- ограничения (constraints):

- жесткое (hard), 621
- интегральный учет пространственных ограничений (integrating spatial), 599
- при распознавании трехмерных объектов (3D object recognition), 627
- реляционные, при символическом сопоставлении (relational, in symbolic matching), 514
- упорядочения (ordering), 517
- эпиполярное (epipolar), 515
- одномерные сигналы, дифференцирование, 186
- однородность группы (group homogeneity), 120
- однородные координаты (homogeneous coordinates), определение, 425
- озера (lakes), 149
- окрестности пикселей (pixels neighborhoods), 79
- округлость (circularity), 106
- окружности, обнаружение с помощью преобразования Хафа (Hough transform), 396
- октантное дерево (octrees), 593, 613
- Оландера (Ohlander) рекурсивный гистограммный метод кластеризации, 369
- оператор (operator):
  - дифференциальный (difference), 190
  - Кэнни (Canny), 207
  - Превитт (Prewitt), 191, 193
  - признаковый (interest), 335
  - Робертса крестообразный (Roberts cross), 192
  - Собеля (Sobel), 192
- операции, определение с помощью масок, 219
- операции обработки изображений, 24
- описывающий прямоугольник (bounding box), 108
- определитель прямых Барнса (Burns line finder), 401
- оптимизация (optimization):
  - и верификация положения объектов, 583
  - линейная при наличии ограничений (constrained linear), 580
  - нелинейная (nonlinear), 409
- организация баз данных, 318
- ортогональные преобразования (orthogonal transforms), определение, 428
- ортогональный базис (orthogonal basis), использование, 210
- ортономированные преобразования (orthonormal transforms), определение, 428
- осветитель (projector):
  - использование вместо камеры в стереосистеме, 529
  - калибровка, 557
  - лазерный, 559
- освещенности граница (illumination boundary), 480
- освещенность (shading), 244–275
  - использование для вычисления формы, 594
  - модель Фонга (Phong model), 271
  - определение формы, 497
  - учет зрительной системой человека, 271
- освещенные объекты (illuminated objects), восприятие, 246
- осмотр архитектурных сооружений (architectural walkthrough), 669
- ось (оси):
  - главная, 111
  - с наименьшим вторым моментом, 114
  - эллипс, 110
- осязательные ощущения (haptic sense), 684
- отверстий, подсчет (holes counting), 17
- относительной ориентации примитив (relative orientation primitive), 652
- отношение (relation), определение, 453
- отображения (mapping):
  - аффинные (affine), использование для распознавания двумерных объектов, 439
  - для изменения тонового распределения (gray-level), 172
  - полиномиальные (polynomial), 472
  - текстурные (texture), 687
  - функции, аффинные, 425
- отражение (reflection):
  - диффузное (diffuse), 266
  - зеркальное (specular), 267
- отражение, по закону Ламберта (Lambertian reflectance), 596, 266



- отрезки прямых (straight lines segments), обнаружение, 394
- отсечение (clipping), 47
- отсутствия препятствий примитив (clearance primitive), 652
- оценка (estimation):
  - положения (pose), 578
  - процедура оценки положения (pose estimation procedure), 560
- Оцу (Otsu) метод для автоматического выбора порога бинаризации, 119
- ошибки (errors):
  - коэффициенты ошибок (rate), классификация, 131
  - ложные срабатывания (false alarms) и ложные пропуски (false dismissals), 132
  - системы распознавания, 130
  - среднеквадратическая (root-mean-square error, RMSE), 404
  - связь с зоной обзора (coverage), 517
  - статистическая интерпретация (statistical interpretation), 407
  - эмпирическая интерпретация (empirical interpretation), 406
- Панорамирование (panning), 353**
- параллактическое смещение (motion parallax), 495
- параллельная реализация (parallel implementation) 225
- параметрические модели (parametric models) распределений, 154
- параметры:
  - аналитические выражения, 405
  - внешние параметры камеры (extrinsic camera), 568
  - внутренние параметры камеры (intrinsic camera), 567
  - модели камеры, 555
- первичный ключ (primary key), 318
- перебор с возвратом (backtracking), для маркировки ребер объектов с плоскими гранями, 487
- передняя плоскость изображения (front image plane), 507
- перекрытие (occlusion), 492
- перенос (translation):
  - в трехмерном пространстве, 531
  - на бинарном изображении, определение, 95
  - на плоскости, 428
  - параметры расположения камеры, 568
  - переходы через 0 (zero crossings), 188, 203
- периметр, 106
- перспективная задача трех точек (perspective 3 point problem, P3P), 561
- перспективное масштабирование (perspective scaling), 493
- перспективное преобразование (perspective):
  - матрица преобразования (transformation matrix), 540
  - модель формирования изображений (imaging model), 506
  - обратное (inverse), 560
  - проекции, 544
- перцептрон (perceptron), 160
- ПЗС, приборы с зарядовой связью (CCD, charge-coupled devices):
  - камеры, 39
  - массивы, 41, 43
  - неоднородности, 46
- пиксели:
  - модификация, 25
  - окрестности (neighborhoods), 79
  - определение, 15
  - переднего плана (foreground), 79
  - стыка (junction), 388
  - фоновые (background), 79
- пиксельная система координат (pixel coordinate frame), 68
- пиксельные значения (pixel values), обозначение, 40
- пиксельный массив (pixel array), 68
- плавное исчезновение (fade), 354
- пластины (plates), 640
- плоскость изображения, передняя (front image plane), 507
- плотность и направление краев в анализе текстур (density and direction of edges), 280
- плотность краев поверхностная (edgeness per unit area), 280
- площадь, 105
- поверхности, самоперекрывающиеся (self-occluding surfaces), 478



- поверхность-ребро-вершина, модели, 609
- поворот (rotation):
  - в трехмерном пространстве, 532
  - на плоскости, 428, 429
  - относительно произвольной оси, 535
- поворотные изображения (spin images), 631
- повышение качества изображений (enhancing images), 26, 169–243
  - определение, 171
- подбор моделей (fitting):
  - для контурных сегментов, 403–410
  - для сегментации кривых, 409
  - ограничения (constraints), 409
  - типичные проблемы, 408
- подсчет (counting):
  - объектов на изображении, 82
  - отверстий, 17–19
- поиск (retrieval):
  - изображений, индексация для систем поиска с несколькими мерами расстояния, 322
  - изображений на основе содержания, 293–326
  - документов (document retrieval, DR), 133
  - задачи, 16
- поле движения (motion field):
  - вычисление с использованием соответствующих точек, 334
  - определение, 331
- поле зрения (field of view, FOV), угловое 52
- полигональная аппроксимация (polygonal approximation), 380
- полиномиальные отображения (polynomial mappings), 472
- полнота выборки (recall):
  - взаимосвязь с точностью, 133
  - определение, 133
- полоснопропускающая фильтрация (bandpass filtering), 337
- полосы Маха (Mach band effect), 203
- полутоновые изображения (gray-scale images):
  - определение, 49
  - пороговая бинаризация (thresholding), 116
- понимание изображений (image understanding), 31
- пороговая бинаризация (thresholding):
  - верхняя (above), 116
  - внешняя (outside), 117
  - динамическая (dynamic), 123
  - на основе априорной информации (knowledge-based), 123, 369
  - нижняя (below), 116
  - по диапазону (inside), 117
  - полутоновых изображений, 116–124
  - с автоматическим выбором порога, 119
- пороговые значения (threshold values), 41
- построчный алгоритм маркировки (row-by-row labeling algorithm), 86
- потемнение при увеличении расстояния (darkening with distance), 269
- поток изображения (image flow):
  - вычисление, 341
  - определение, 332
  - распространение ограничений при вычислении, 344
  - уравнение, 342
- правило ближайшего соседа (nearest-neighbor rule), 139
- Превитт (Prewitt)
  - маски, 398
  - оператор (Prewitt operator), 191
- предварительная обработка изображений (conditioning images), 169
- представление (representation):
  - в системе Veggio Vision, 697
  - в системе идентификации по радужной оболочке глаза, 710
  - в виде векторов признаков (feature vector), 136
  - каркасные модели (mesh models), 608
  - модели типа «поверхность-ребро-вершина» (surface-edge-vertex models), 609
  - символов с использованием признаков, 133
  - трехмерных моделей (3D models), 608–617
- представление областей
  - в виде маркированных изображений, 378

- представление с помощью квадрантных деревьев, 380
- представление с помощью таблиц свойств (property tables), 381
- прослеживание границ (tracking boundaries), 382
- способы представления, 377
- угловые (corners), 413
- преобразование (converting):
  - из цветовой системы RGB (красный–зеленый–синий) в YUV, 257
  - из цветовой системы RGB (красный–зеленый–синий) в систему HSI (тон–насыщенность–интенсивность) encoding, 255
- преобразования (transformations):
  - вычисление преобразования  $T_r = \{RT\}$ , 581
  - двумерные (2D), 422
  - для кластеризации по расположению, 444
  - для отображения модельных признаков на признаки изображения, 443
  - жесткие (rigid), 428
  - линейные (linear), 426
  - матрица перспективной проекции, 540
  - ортогональные и ортонормированные (orthogonal and orthonormal), 428
  - совмещение посредством аффинных преобразований, 537
  - трехмерные аффинные (3D affine), 530
  - Фурье (Fourier), 232
- приборы с зарядовой связью, ПЗС (charge-coupled devices, CCD):
  - камеры, 39, 41
  - массивы, 41, 43
  - неоднородности, 46
- признаки дальностные (cues):
  - границы и виртуальные прямые (boundaries and virtual lines), 504
  - на основе неслучайных совмещений (from non-accidental alignments), 505
  - на основе фокусировки (depth from focus), 503
  - связанные с движением (motion phenomena), 504
  - точки схода (vanishing points), 502
  - трехмерные признаки на двумерных изображениях, 491
- форма по данным освещенности (shape from shading), 497
- форма по контурам (shape from boundary), 499
- форма по текстурной информации (shape from texture), 498
- признаковые операторы (interest operators), 335
- прикладные задачи (application problems), 15
- приложения бинарной морфологии (binary morphology applications), 98
- примеры систем компьютерного зрения:
  - идентификация личности по радужной оболочке глаза (identifying humans via iris of eye), 704
  - система Veggie Vision, 694
- примитивы знаний (knowledge primitives), в системе GRUFF, 651
- принятие решений (decision-making):
  - Байесовский подход (Bayesian method), 152
  - при обработке многомерных данных, 156
- проблемы двух классов (two-class problems), 132
- проволочные каркасные модели (wire-frame models), 609
- проекция (projection):
  - максимальной интенсивности (maximum intensity projection, MIP), 73
  - прямоугольная (orthographic), 544, 596
  - слабая перспективная (weak perspective), 544
- промежуточное расположение (interposition), 66
- определение, 492
- пространственная индексация (spatial indexing), 322
- пространственно-временной градиент (spatio-temporal gradient), 414

- пространственное положение и ориентация объектов (object pose), вычисление, 526–606
- пространственные измерения (spatial measurement) и дискретизация изображений, 50
- пространственные ограничения (spatial constraints), 599
- пространственные отношения (spatial relationships), 316
- пространственные частоты (spatial frequency) и гармонический частотный анализ, 225
- прямого распространения нейронные сети (feedforward networks), 165
- прямолинейные ленты (straight ribbons), 412
- прямоугольная проекция (orthographic projection), 544, 596
- прямые (lines):
- виртуальные (virtual), 504
  - обнаружение с помощью преобразования Хафа (Hough transform), 393
  - определитель прямых Барнса (Burns line finder), 401
  - подбор моделей (fitting), 403
  - релаксационная маркировка (labeling via relaxation), 489
  - эпиполярные (epipolar), 516
- псевдоцвет (pseudo color), 273
- псевдоцветовое изображение (pseudo-colored image), 50
- пятна (blobs), 27
- Радиальная дисторсия (radial distortion), устранение, 471**
- радиальное расстояние (radial distance):
- среднее (mean), 107
  - среднеквадратическое отклонение (standard deviation), 107
- различение (discrimination), улучшение, 661
- размер, использование в системе Veggie Vision, 701
- размерность, высокая (high dimensionality), 409
- размеров примитив (dimensions primitive), 652
- размыкание (opening) бинарных изображений, 97
- размытие (blur), связь с разрешением, 520
- разрешающая способность (resolving power), определение, 521
- разрешение (resolution):
- номинальное (nominal), определение, 50
  - определение, 50
  - связь с размытием (blur), 520
  - субпиксельное (subpixel), определение, 50
- разрывный край (jump edge), 478
- определение, 480
- ракурс (aspect), 618
- ракурса эффект (foreshortening), 493
- распознавание (recognition):
- в системе Veggie Vision, 697
  - по внешнему виду (by appearance), 661
  - посредством совмещения (by alignment), 436
  - символов, 133
  - структурные методы (structural techniques), 140
  - структурные образы (structural patterns), 143
- распознавание двумерных объектов с использованием реляционных моделей, 451
- распознавание образов (pattern recognition):
- задачи, 126
  - основные понятия, 126–168
- распознавание объектов (object recognition), 433
- в системе GRUFF, 651–655
  - двумерных, с использованием аффинных преобразований, 439
  - двумерных, с использованием реляционных моделей, 451
  - метод базовых локальных признаков (local-feature-focus method), 442
  - метод геометрического хэширования (geometric hashing), 447
  - система RIO, 639–650
  - система TRIBORS, 635
  - трехмерных, классификация методов, 627

- трехмерных, по внешнему виду (by appearance), 654
- распознавание по внешнему виду (recognition by appearance), 654
- распознавание по внешнему виду с использованием пространства собственных векторов (eigenspace recognition by appearance), 661
- распознавание символов (character recognition), 133–136
- расположение (pose):
  - в системах виртуальной реальности, 683
  - верификация и оптимизация, 583
  - кластеризация (clustering), 444
  - на основе соответствия точек 2D-3D, 578
  - определение расположения объектов и восприятие трехмерных сцен, 526–606
  - определение, 444
  - оценка (estimation), 578
  - процедура оценки (estimation procedure), 560
- распределение
  - вероятности (probability), 153
  - нормальное (normal), определение, 154
  - параметрические модели (parametric models), 154
- рассеянный свет (ambient light), определение 270
- рассогласование (disparity), определение, 511
- расстояние (distance):
  - в реляционном сопоставлении (relational matching), 463
  - меры расстояния и индексация в системах поиска изображений на основе содержания, 322
  - меры расстояния между изображениями, 298
  - от выбранного образца (pick-and-click), 303
  - потемнение при увеличении (darkening with), 269
  - Хэмминга (Hamming distance), 712
- растровые системы координат (raster-oriented coordinate systems), 49
- растровый порядок (raster order), 56
- реальные и искусственные изображения, сочетание, 687
- ребро (edge), 483
  - маркировка ребер объектов с плоскими гранями методом дискретной релаксации, 489
  - маркировка ребер объектов с плоскими гранями, 487
  - модели типа поверхность-ребро-вершина (surface-edge-vertex models), 609
- реконструкция (reconstruction):
  - поверхности (surface), 589
  - трехмерных объектов, 584
- рекурсивный алгоритм маркировки (recursive labeling algorithm), 86
- релаксация (relaxation):
  - дискретная (discrete), 456
  - дискретная, для маркировки плоскогранных объектов, 489
  - для маркировки контурных изображений, 489
  - непрерывная (continuous), 459
- реляционная индексация (relational indexing), 467, 648
- См. также* Система распознавания объектов RIO
- реляционное описание (relational description), определение, 463
- реляционное расстояние (relational distance), 463
- реляционное сопоставление (relational matching), для распознавания двумерных объектов, 451
- реляционные модели (relational models):
  - на основе классов видимости (view-class), 642
  - сопоставление (matching), 639
- реляционные ограничения (relational constraints), 514
- рендеринг (rendering):
  - на основе изображений (image-based), 688
  - трехмерных моделей (3D models), 685
- рентгеновские изображения (X-ray images), 72
- рецепторы (receptors), чувствительность, 248

- Робертса маски (Roberts masks), 192
- Самоперекрывающаяся поверхность (self-occluding surface), 478
- свертка (convolution), 170
- кросс-корреляция (cross correlation), 219
  - операция, 221
  - определение, 221
  - теорема, 238
- свет (light):
- белый, 247
  - восприятие (sensing), 38, 246
  - диффузное отражение (diffuse reflection), 266
  - зеркальное отражение (specular reflection), 267
  - излучение одиночного источника, 264
  - использование при получении изображений, 65
  - потемнение при увеличении расстояния (darkening with distance), 269
  - рассеянный (ambient), 270
  - структурированный (structured), 557
- свойства областей (region properties), 102
- связные компоненты (connected components):
- алгоритм, классический, 90, 94
  - маркировка (labeling), 84
  - маркировка, с применением группового кодирования (using run-length encoding for labeling), 91
- сгиб (crease), 479, 480, 483
- определение, 480
- сглаживание (smoothing):
- гауссовское (Gaussian), 205
  - изображений, 179
- сглаживающие маски (smoothing masks), 189, 219
- сдвиг (shear), 436
- сегментация (segmentation):
- видеопоследовательностей, 354
  - изображений, 362–420
  - кривых с использованием методов аппроксимации, 409
  - на основе согласованного движения, 414
  - текстурная, 289
  - цветных изображений, 261, 416
- сегменты (segments):
- подбор моделей, 403
  - прямых, 394
- сеточный шаблон освещения, 557
- сжатие (compression): данных, 57
- без потерь (lossless), 57
  - с потерями (lossy), 57
- формат JPEG (Joint Photographic Experts Group), 60
- формат Motion JPEG (Joint Photographic Experts Group), 62
- формат MPEG для видеоданных, 339
- сигналы (signals):
- дифференцирование одномерных сигналов, 186
  - представление в виде комбинации базисных сигналов, 211
  - телевизионные, цветковые системы YIQ и YUV, 256
- сигнальный уровень (signal level), 654
- символическое сопоставление (symbolic matching), 514
- система Veggie Vision, 694
- система классификации (classification system):
- ложные срабатывания (false alarms) и ложные пропуски (false dismissals), 132
  - оценка коэффициента ошибок (evaluating error rate), 130
  - построение, 130
- система координат камеры C (camera coordinate frame), 68
- система поиска изображений на основе содержания QBIC (Query by Image Content), 294
- система распознавания объектов GRUFF (Generic Object Recognition Using Form and Function), 651
- дерево категорий (definition tree), 653
  - иерархия категорий (category hierarchy), 652
  - обработка данных, 655
  - примитивы знаний (knowledge primitives), 651
  - функциональные свойства (functional properties), 652
- система распознавания объектов RIO, 643

- используемые характерные признаки, 643, 646
- система распознавания объектов TRIBORS, 635
- системы виртуальной реальности (virtual reality systems), 667–693
- виртуальное выполнение сложных действий (dextrous virtual work), 680
- вывод зрительной информации, 682
- головной дисплей (head-mounted display) HMD, 671, 678
- датчики, 682
- дополненная реальность (augmented reality), 671
- интерактивная сегментация анатомических структур, 670
- использование информации о движении, 684
- моделирование полетов (flight simulation), 670
- осмотр архитектурных сооружений, 669
- основные признаки, 668
- осязательное восприятие (haptic sense), 684
- приложения, 669
- телеуправление (teleoperation), 674
- устройства вывода стереоскопических изображений (stereoscopic display devices), 681
- человеко-машинный интерфейс, 691
- системы дополненной реальности (augmented reality, AR), 671
- См. также* Виртуальная реальность системы координат (coordinate frames), 67, 424, 530
- растровая (raster-oriented), 49
- скос (slant), определение, 499
- слабая перспективная проекция (weak perspective projections), 544
- слежение с учетом априорной информации, 351
- сложение изображений (image addition), 26
- сложные ручные операции, виртуальное выполнение (dextrous virtual work), 680
- случайные совмещения (accidental alignments), 505
- смежности областей графы (region adjacency graphs, RAG), 114
- смена монтажного кадра (shot change), 353
- смена сцены (scene change), 353
- Собеля маски (Sobel masks), 192
- совмещение (registration):
- видов (views), 588
- изображений, 423
- совмещение:
- использование при распознавании трехмерных моделей (matching geometric models via alignment), 629
- объектов с гладкими поверхностями (smooth object), 636
- посредством преобразований (via transformation calculus), 537
- случайное (accidental), 505
- типа 2D-3D, 633
- типа 3D-3D, 630
- См. также* Сопоставление согласованная маркировка (consistent labeling), 453
- сонификация (sonification), определение, 683
- соответствие (correspondence):
- кросс-корреляция (cross-correlation), 513
- ограничение упорядочения (ordering constraint), 517
- ошибки и зона обзора (error versus coverage), 517
- при совмещении типа 3D-3D (3D-3D alignment), 630
- символическое сопоставление и реляционные ограничения (symbolic matching and relational constraints), 514
- точечное типа 2D-3D для оценки положения (pose from 2D-3D point), 578
- установление (establishing), 513
- эпиполярное ограничение (epipolar constraint), 515
- соответствующие точки, при вычислении поля движения, 334
- сопоставление (matching):
- в двумерном пространстве, 421–475

- геометрических моделей (geometric models) посредством совмещения (alignment), 629
- границ, 309
- графов, алгоритмы, 143
- по реляционному расстоянию (relational distance), 463
- реляционное (relational), для распознавания двумерных объектов, 451
- реляционных моделей (relational models), 639
- с использованием трехмерных моделей, 607–666
- символическое (symbolic), 514
- функциональных моделей (functional models), 651
- эскизное (sketch), 310
- См. также* Совмещение
- соседи (neighbors):
  - ближайшие (nearest), использование для классификации, 139
- составляющие модельные точки (contributing points), 632
- спектр мощности (power spectrum), 232
- и автокорреляция для описания текстуры, 287
- спутниковые изображения (satellite images), 22
- среднее значение класса (class mean), использование для классификации, 137
- среднее радиальное расстояние (mean radial distance), 107
- среднеквадратическая ошибка (root-mean-square error, RMSE), 404
- среднеквадратическое отклонение (standard deviation), 139
- радиального расстояния (radial distance), 107
- стандартные индексы (standard indexes), 318
- статистическая интерпретация ошибки (statistical interpretation of error), 407
- стерео (stereo):
  - изображения, 512
  - использование для определения глубины, 509
  - конфигурация (configuration), 527
  - система сбора данных (acquisition system), 585
  - системы, обнаружение соответствующих признаков, 513
  - фотометрический метод, 598
  - стереоскопические устройства вывода (stereoscopic display devices), 681
  - стереофотометрический метод (photometric stereo), 598
  - стержни (sticks), 640
  - стробоскопический свет (strobe light), 65
  - структурирующие элементы (structuring elements), 93, 98
  - структурная подсветка (structured light), 65, 557
  - структурное распознавание образов (structural pattern recognition), 140
  - структуры (structures):
    - высокоуровневые, 410
    - для объединения-поиска (union-find), 88
    - ленты (ribbons), 410
    - определение по данным о движении, 600
  - пространственная, оценка по двумерным изображениям, 65
  - углы (corners), 413
  - стыки (junctions), 483
    - вилки (fork junctions), 485
    - пиксели (junction pixels), 388
    - разновидности, 484
    - стрелки (arrow junctions), 485
    - типа L (L-junctions), 485
  - субпиксельное разрешение (subpixel resolution), определение, 50
  - субтрактивная цветовая система CMY, голубой–пурпурный–желтый (subtractive color system CMY, cyan-magenta-yellow), 252
  - суперквадрики (superquadrics), 615
  - схода точки (vanishing points), 66, 502
  - сходство (similarity):
    - по взаимному расположению объектов, 313
    - тектурное, 302
    - формы, 305
    - цветное, 299



- Таблица свойств (property table), 381
- теговый формат файлов изображений (Tag Image File Format, TIFF), 60
- текселы (texels), 278
- использование для описания текстур, 278
- текстовые приложения (text applications), 21
- текстура (texture), 276–292
- использование в системе Veggie Vision, 700
  - автокорреляция и спектр мощности (autocorrelation and power spectrum), 287
  - вектор текстурного описания (texture description vector), 303
  - гистограммы (histograms), 305
  - двоичное разбиение (binary partition), 282
  - использование для определения формы, 498
  - количественные характеристики (quantitative measures), 280
  - матрицы вхождений (co-occurrence matrices) для вычисления характерных признаков, 282
  - плотность и направление краев (edge density and direction), 280
  - сегментация (segmentation), 289
  - статистическое описание, 278
  - структурное описание, 278
  - тексельное описание (texel-based description), 278
  - характеристики сходства (similarity measures), 303
  - энергетические характеристики (texture energy measures), 285
  - энергия (energy), 285
- текстурирование на основе видов (view-based texturing), 688
- текстурное отображение (texture mapping), 687
- текстурный градиент (texture gradient), 66, 495
- телевизионный сигнал, цветовые системы YIQ и YUV, 256
- телеуправление (teleoperation), 674
- теневые области (shadows), 503
- теорема (theorem):
- запаздывания (shift), 239
  - о свертке (convolution), 238
- тон–насыщенность–интенсивность (HSI) для представления цвета, 252
- тоновое распределение, изменение (gray-level mapping), 172
- тонкой линзы формула (thin lens equation), 518
- точный оператор (point operator), определение, 173
- точки:
- вычисление трехмерных точек с использованием нескольких камер, 547
  - интереса (interesting points), 335
  - оценка положения на основе соответствия точек 2D–3D, 578
  - представление на плоскости, 424
  - составляющие модельные (contributing), 632
  - схода (vanishing points), 66, 502
  - управляющие (control), 430
- См. также* Перспективная задача трех точек
- точность (precision):
- взаимосвязь с полнотой выборки (recall), 133
  - определение, 133
- траектории (trajectories):
- вычисление, 345
  - накопление траекторий движения, 415
  - определение, 346
- трассировка лучей (ray tracing), 686
- треугольники, совмещение, 537
- треугольное дерево (triangle-tree), 324
- трехмерная структура (3D structure) сцен, оценка по двумерным изображениям, 65
- трехмерное совмещение (3D–3D alignment), 630
- трехмерные (3D):
- аффинные преобразования (affine transformations), 530
  - объекты, методы распознавания, 627
  - оценка положения на основе соответствия точек 2D–3D, 578
- трехмерные изображения (3D images):
- восприятие по двумерным изображениям, 476–525



- интерпретация на основе данных об освещенности, 497
- интерпретация на основе контуров, 499
- интерпретация на основе текстурной информации, 498
- интерпретация на основе точек схода, 502
- маркировка контурных изображений, 483
- трехмерные модели (3D models):
  - истинные трехмерные и на основе классов видимости, 618
  - каркасные (mesh), 608
  - методы представления (representation methods), 608–617
  - на основе обобщенных цилиндров (generalized-cylinder), 612
  - на основе октантных деревьев (octrees), 615
  - на основе суперквадрик (superquadrics), 615
  - оболочечные (balloon), 624
  - разновидности и использование для распознавания объектов, 607–666
  - реляционные (relational), 639
  - рендеринг (rendering), 685
  - совмещение типа 2D-3D, 633
  - совмещение типа 3D-3D, 630
  - типа «поверхность-ребро-вершина» (surface-edge-vertex), 609
  - физические и деформационные (physics-based and deformable), 619
  - человеческого сердца (human heart), 626
  - человеческого тела (human body), 612
- См. также* Модели
- трехмерные объекты (3D objects):
  - распознавание по внешнему виду (recognition by appearance), 654
  - реконструкция (reconstruction), 584
  - система распознавания объектов RIO, 643
- трехмерные признаки (3D cues) на двумерных изображениях, 491
- трехмерные сцены (3D scenes),
  - восприятие и оценка расположения объектов, 526–606
  - трехмерные точки (3D points), вычисление с использованием нескольких камер, 547
  - трехцветное цветовое кодирование (trichromatic encoding), 249
  - триангуляция (triangulation), 75
  - триплеты (triplets), 635
- У**величение контраста (contrast stretching), определение, 173
- угловое поле зрения (angular field of view), 52
- углы (corners), 483
  - обнаружение (detecting), 413
  - шаблоны, 18
- удаление (removing):
  - малых областей из изображений, 176
  - малых компонент (small components), 178
  - шума «соль и перец» (salt-and-pepper noise), 177
- узлы (nodes), 145, 320
  - квадрантные деревья (quadtrees), 380
  - октантные деревья (octree), 615
- упорядочения ограничение (ordering constraint), 517
- управляющие точки (control points), 430
- условное наращивание (conditional dilation):
  - в бинарной морфологии, 103
  - определение, 103
- усредняющего сглаживания маски (box smoothing masks), 190
- усредняющий фильтр, определение, 181
- устойчивости примитив (stability primitive), 652
- Ф**айловые форматы (file formats):
  - GIF (Graphics Interchange Format), 60
  - JPEG (Joint Photographic Experts Group), 60
  - MPEG (Motion Picture Experts Group) для видеоданных, 62
  - TIFF (Tag Image File Format), 60
- См. также* Форматы
- физические (physics-based) модели, 619
- фильтрация (filtering):
  - гауссовская и ЛОГ-фильтрация для обнаружения краев, 196

- изображений, 169–243
- ЛОГ-фильтрация и теория Марра-Хидрес, 204
- медианная (median), 181
- полоснопропускающая (bandpass), 237
- фильтры (filters):
  - Гауссовский, 181
  - ЛОГ-фильтр, 200
  - согласованные (matched), 207
  - усредняющий, определений, 181
- фокус (focus):
  - использование для определения глубины, 503
  - расширения (focus of expansion, FOE), 332, 504
  - связь с глубиной поля зрения (depth of field), 519
  - схода (focus of contraction, FOC), 332
- Фонга модель освещения (Phong model of shading), 271
- фоновые пиксели (background pixels), 79
- форма (shape):
  - вычисление по данным освещенности, 594
  - гистограммы формы, 307
  - использование в системе Veggio Vision, 701
  - определение по контурам, 499
  - определение по освещенности, 497
  - определение по текстурной информации, 498
  - характеристики сходства (similarity measures), 305
- форматы (formats):
  - EPS (инкапсулированный Postscript), 62
  - GIF (Graphics Interchange Format), 60
  - JPEG (Joint Photographic Experts Group), 60
  - MPEG (Motion Picture Experts Group) для видеоданных, 62
  - PostScript, 62
  - TIFF (Tag Image File Format), 60
  - для представления цифровых изображений, 55
  - сравнение, 63
  - часто используемые, 57
- формирование изображений (imaging):
  - и представление изображений (image representation), 37–77
  - устройства (devices), 39
  - формирование неграфических описаний (non-iconic representations), 28
  - формула тонкой линзы (thin lens equation), 518
  - фотография, модель 38
  - фреймграббер (frame grabber), 40
  - Фрея-Чена базис (Frei-Chen basis), 214
  - Фримана цепной код (Freeman chain code), 379
  - функции аффинных преобразований (affine mapping functions), 425
  - масштабирование (scaling), 426
  - перенос (translation), 428
  - поворот (rotation), 427
  - функциональные:
    - модели (functional models), сопоставление, 651
    - свойства (functional properties) в системе GRUFF, 652
  - функция интенсивности (picture function):
    - двумерная, 230
    - и цифровые изображения, 47
    - определение, 48
    - Фурье анализ (Fourier analysis), 225
    - Фурье базис (Fourier basis), 227
    - использование в операциях обработки изображений, 229
    - Фурье преобразование (Fourier transform):
      - быстрое (fast), 238
      - дискретное (discrete), 234
      - Фурье спектр мощности (Fourier power spectrum), 232
- Характеристики (measures):
  - для построения мер расстояния между изображениями, 298
  - сходства формы (shape similarity), 305
  - сходства, учитывающие присутствующие на изображении объекты, 313
  - текстурного сходства (texture similarity), 303
  - текстуры (texture), 280

– цветового сходства (color similarity), 299

характерных признаков экстрактор (feature extractor), 129

Хафа преобразование (Hough transform):

- алгоритм, 397
- для обнаружения прямых и дуг окружностей, 393
- дополнения, 399
- использование в определителе прямых Барнса (Burns line finder), 401
- обнаружение окружностей, 396
- обобщенное (generalized), 401

хранение видеопоследовательностей (storing video sequences), 359

хроматическая дисторсия (chromatic distortion), 47

хэш-индексы (hash indexes), 319

хэш-таблицы (hash tables), 319

- в реляционной индексации (relational indexing), 645

хэш-функция (hash function), 319

хеширование геометрическое (geometric hashing), 447

**Ч**ая метод калибровки (Tsai calibration method), 567

цвет (color), 244–275

- гистограммы, 289, 300, 302
- изображения, 70
- использование в системе Veggie Vision, 700
- использование для классификации, 257
- куб, 253
- приложения, 272
- псевдоцвета (pseudocolors), 273
- сегментация (segmentation), 261, 416
- система RGB, красный-зеленый-синий (RGB, red-green-blue), 249
- субтрактивная цветовая система CMY, голубой-пурпурный-желтый (subtractive color system CMY, cyan-magenta-yellow), 252
- треугольник (triangle), 251
- физические свойства, 245
- характеристики сходства (similarity measures), 299

- цветовая система HSI, тон-насыщенность-интенсивность (hue-saturation-intensity), 252
- цветовой макет (color layout), 300
- человеческое восприятие (human perception), 272
- шестигранный конус (hexacone), 253, 254

цветовая система HSI (тон-насыщенность-интенсивность), 252

- кодирование, преобразование из системы RGB (красный-синий-зеленый), 255

цветовая система HSV (тон-насыщенность-светлота), 253

цветовая система RGB (красный-синий-зеленый):

- для представления цвета, 249
- преобразование в систему HSI, 255
- преобразование в систему YUV, 257

цветоразностные данные (chrominance), 256

центр тяжести (centroid), 105

цепной код Фримана (Freeman chain code), 379

циклический возврат (wrap-around), 47

цилиндрическая деформация фрагмента изображения (cylindrical warp of image region), 469

цилиндры (cylinders): модели на основе обобщенных цилиндров, 612

цифровое изображение (digital image), 15

- определение, 48
- проблемы формирования, 45
- форматы (formats), 55
- функция интенсивности (picture function), 47

**Ч**астота, пространственная, анализ с использованием гармонических функций, 225

часть (part), 452

человеко-машинный интерфейс, в системах виртуальной реальности, 691

человеческая зрительная система, обнаружение перепадов, 201

человеческий глаз, как камера, 44

- человеческое восприятие (human perception):  
– глубины (depth), 505  
– учет освещенности поверхностей, 271  
– цвета, 272  
человеческое сердце, моделирование движения, 626  
человеческое тело, трехмерные модели, 614  
четные функции (even functions), 231  
четырёхсвязная окрестность (4-neighbors), 79
- Ш**  
**Шестигранный цветовой конус** (hexacone), 253  
**Ши (Shi) метод кластеризации**, 371  
**шкалированное Евклидово расстояние** (scaled Euclidean distance), определение, 139
- шум**  
– Гауссовский (Gaussian noise), 179, 407  
– «соль и перец» (salt-and-pepper noise), удаление 177
- Эвристические правила (heuristics) для обнаружения масштабирования**, 358  
**экстрактор характерных признаков** (feature extractor), 129  
**экстремальной оси длина** (extremal axis length), 109  
**экстремальные точки** (extremal points), 108  
**эластичное сопоставление** (elastic matching), 310  
**электромагнитный спектр** (electromagnetic spectrum), 245  
**эллипс** (ellipse), 613  
– оси (axes), длина и направление, 110
- эмпирическая интерпретация ошибок** (empirical interpretation of error), 406  
**эмпирический коэффициент**:  
– выбросов (empirical reject rate), 131  
– ошибок (empirical error rate), 131  
**энергия изображения** (image energy), 623  
**энергия, минимизация**, 623  
**энтропия** (entropy):  
– вычисление, 147  
– множества событий, 147  
**эпиполе** (epipole), 516  
**эпиполярные** (epipolar):  
– конфигурации (geometry), 515  
– ограничения (constraint), 515  
– плоскости (plane), 516  
– прямые (lines), 516  
**эрозия бинарных изображений** (erosion of binary images), 96  
**эскизное сопоставление** (sketch matching), 310  
**эталонные изображения** (training images), использование для вычисления базисных изображений, 656  
**эффект масштабирования** (camera zoom), определение, 353  
**эффекты камеры** (camera effects):  
– игнорирование, 357  
– определение, 353  
**эффекты пространственной дискретизации** (spatial quantization effects), 53
- Якобиан, матрица** (Jacobian matrix), 563  
**яркость** (luminance), 256

---

# Оглавление

Предисловие редактора перевода .....	5
Предисловие к русскому изданию .....	7
Предисловие .....	8
Замечания относительно языка программирования .....	10
Варианты работы с книгой .....	10
Глава 1. <b>Введение</b> .....	13
1.1. Могут ли машины видеть? .....	14
1.2. Прикладные задачи .....	15
1.2.1. Предварительное рассмотрение структуры цифрового изображения .....	15
1.2.2. Поиск в базе данных изображений .....	16
1.2.3. Контроль отверстий в поперечных балках .....	17
1.2.4. Контроль медицинских изображений головы человека .....	20
1.2.5. Обработка сканированных страниц текста .....	21
1.2.6. Оценка снежного покрова по спутниковым изображениям .....	22
1.2.7. Анализ сцен, содержащих детали промышленного производства .....	23
1.3. Операции обработки изображений .....	24
1.3.1. Модификация пикселей в малых окрестностях .....	25
1.3.2. Глобальное улучшение качества изображения .....	26
1.3.3. Комбинация нескольких изображений .....	26
1.3.4. Вычисление характерных признаков изображения .....	27
1.3.5. Формирование неграфических описаний .....	28
1.4. Успехи, сложности и неудачи компьютерного зрения .....	29
1.5. Применение компьютеров и программного обеспечения .....	30
1.6. Смежные области .....	31
1.7. Структура оставшейся части книги .....	32
1.8. Литература .....	32
1.9. Дополнительные упражнения .....	34
Глава 2. <b>Формирование и представление изображений</b> .....	37
2.1. Восприятие света .....	38
2.2. Устройства для формирования изображений .....	39
2.2.1. Камеры на основе ПЗС .....	39
2.2.2. Формирование изображения .....	41

2.2.3. Видеокамеры . . . . .	43
2.2.4. Человеческий глаз . . . . .	44
2.3. Проблемы формирования цифровых изображений . . . . .	45
2.3.1. Геометрические искажения . . . . .	45
2.3.2. Дисперсия . . . . .	46
2.3.3. Блюминг (избыточная яркость) . . . . .	46
2.3.4. Неоднородности ПЗС-матрицы . . . . .	46
2.3.5. Отсечение и циклический возврат . . . . .	47
2.3.6. Хроматическая дисторсия . . . . .	47
2.3.7. Эффекты дискретизации . . . . .	47
2.4. Функции интенсивности и цифровые изображения . . . . .	47
2.4.1. Типы изображений . . . . .	47
2.4.2. Дискретизация изображений и пространственные измерения . . . . .	50
2.5. Форматы цифровых изображений* . . . . .	55
2.5.1. Заголовок файла изображения . . . . .	56
2.5.2. Данные изображения . . . . .	56
2.5.3. Сжатие данных . . . . .	57
2.5.4. Часто используемые форматы . . . . .	57
2.5.5. Групповое кодирование бинарных изображений . . . . .	58
2.5.6. Формат PGM: Portable Gray Map . . . . .	58
2.5.7. Формат файлов изображений GIF . . . . .	60
2.5.8. Формат файлов изображений TIFF . . . . .	60
2.5.9. Формат JPEG для хранения фотографий . . . . .	60
2.5.10. Язык PostScript . . . . .	62
2.5.11. Файловый формат MPEG для хранения видеоданных . . . . .	62
2.5.12. Сравнение форматов файлов изображений . . . . .	63
2.6. Особенности и проблемы получения изображений естественных сцен . . . . .	64
2.7. Оценка пространственных свойств объектов по двумерным изображениям . . . . .	65
2.8. Пять систем координат . . . . .	67
2.8.1. Пиксельная система координат I . . . . .	68
2.8.2. Система координат объекта O . . . . .	68
2.8.3. Система координат камеры C . . . . .	68
2.8.4. Действительная система координат изображения F . . . . .	69
2.8.5. Мировая система координат W . . . . .	69
2.9. Другие типы датчиков* . . . . .	69
2.9.1. Микроденситометр* . . . . .	70
2.9.2. Цветные и мультиспектральные изображения* . . . . .	70
2.9.3. Рентгеновские изображения* . . . . .	72
2.9.4. Получение изображений методом ядерного магнитного резонанса (ЯМР)* . . . . .	73
2.9.5. Дальномеры и дальнометрические изображения* . . . . .	74
2.10. Литература . . . . .	76
<b>Глава 3. Анализ бинарных изображений . . . . .</b>	<b>78</b>
3.1. Пикселы и окрестности пикселов . . . . .	79
3.2. Применение масок к изображениям . . . . .	80
3.3. Подсчет объектов на изображении . . . . .	82

3.4. Маркировка связанных компонент . . . . .	84
3.5. Морфология бинарных изображений . . . . .	93
3.5.1. Структурирующие элементы . . . . .	93
3.5.2. Основные операции . . . . .	95
3.5.3. Некоторые приложения бинарной морфологии . . . . .	98
3.5.4. Условное наращивание . . . . .	103
3.6. Свойства областей . . . . .	103
3.7. Графы смежности областей бинарного изображения . . . . .	114
3.8. Пороговая бинаризация полутоновых изображений . . . . .	116
3.8.1. Выбор порога бинаризации по гистограмме . . . . .	117
3.8.2. Автоматический выбор порога бинаризации: метод Оцу* . . . . .	119
3.9. Литература . . . . .	124
<b>Глава 4. Основные понятия распознавания образов . . . . .</b>	<b>126</b>
4.1. Задачи распознавания образов . . . . .	126
4.2. Общая модель классификации . . . . .	128
4.2.1. Классы . . . . .	128
4.2.2. Датчик/преобразователь . . . . .	128
4.2.3. Экстрактор характерных признаков . . . . .	129
4.2.4. Классификатор . . . . .	130
4.2.5. Построение системы классификации . . . . .	130
4.2.6. Оценка ошибок системы . . . . .	130
4.2.7. Ложные срабатывания и ложные пропуски . . . . .	132
4.3. Взаимосвязь точности системы и полноты выборки . . . . .	133
4.4. Признаки, используемые для описания объектов . . . . .	133
4.5. Представление объектов в виде векторов признаков . . . . .	136
4.6. Реализация классификатора . . . . .	136
4.6.1. Классификация по ближайшему среднему значению . . . . .	137
4.6.2. Классификация по расстоянию до ближайших соседей . . . . .	139
4.7. Структурные методы распознавания . . . . .	140
4.8. Матрица неточностей . . . . .	143
4.9. Деревья решений . . . . .	144
4.10. Байесовский подход к принятию решений . . . . .	152
4.10.1. Параметрические модели распределений . . . . .	154
4.11. Принятие решений при обработке многомерных данных . . . . .	156
4.12. Самообучающиеся машины . . . . .	158
4.13. Искусственные нейронные сети* . . . . .	159
4.13.1. Персептрон . . . . .	160
4.13.2. Многослойная сеть прямого распространения . . . . .	165
4.14. Литература . . . . .	167
<b>Глава 5. Фильтрация и улучшение изображений . . . . .</b>	<b>169</b>
5.1. Что необходимо сделать с изображением? . . . . .	170
5.1.1. Повышение качества изображений . . . . .	170
5.1.2. Обнаружение низкоуровневых признаков . . . . .	170
5.2. Изменение тонового распределения . . . . .	172
5.2.1. Выравнивание гистограммы . . . . .	173

5.3.	Удаление малых областей изображения . . . . .	176
5.3.1.	Удаление шума «соль и перец» . . . . .	177
5.3.2.	Удаление малых компонент . . . . .	178
5.4.	Сглаживание изображения . . . . .	179
5.5.	Медианная фильтрация . . . . .	181
5.5.1.	Вычисление выходного изображения по данным входного изображения . . . . .	184
5.6.	Обнаружение краев с помощью дифференциальных масок . . . . .	186
5.6.1.	Дифференцирование одномерных сигналов . . . . .	186
5.6.2.	Дифференциальные операторы для двумерных изображений . . . . .	190
5.7.	Гауссовская фильтрация и ЛОГ-фильтрация для обнаружения краев . . . . .	196
5.7.1.	Обнаружение краев с помощью ЛОГ-фильтра . . . . .	200
5.7.2.	Обнаружение краев в зрительной системе человека . . . . .	201
5.7.3.	Теория Марра—Хилдрет . . . . .	204
5.8.	Детектор краев Кэнни . . . . .	207
5.9.	Использование масок в качестве согласованных фильтров* . . . . .	207
5.9.1.	Векторное пространство всех сигналов, заданных в виде $n$ дискретных значений . . . . .	208
5.9.2.	Использование ортогонального базиса . . . . .	210
5.9.3.	Неравенство Коши—Шварца . . . . .	212
5.9.4.	Векторное пространство изображений размерами $m \times n$ . . . . .	213
5.9.5.	Базис Робертса для окрестностей $2 \times 2$ . . . . .	213
5.9.6.	Базис Фрея—Чена для окрестностей $3 \times 3$ . . . . .	214
5.10.	Свертка и кросс-корреляция* . . . . .	219
5.10.1.	Определение операций посредством масок . . . . .	219
5.10.2.	Операция свертки . . . . .	221
5.10.3.	Возможности параллельной реализации . . . . .	225
5.11.	Анализ пространственных частот с использованием гармонических функций* . . . . .	225
5.11.1.	Базис Фурье . . . . .	227
5.11.2.	Двумерные функции интенсивности . . . . .	230
5.11.3.	Дискретное преобразование Фурье . . . . .	234
5.11.4.	Полоснопропускающая фильтрация . . . . .	237
5.11.5.	Обсуждение преобразования Фурье . . . . .	238
5.11.6.	Теорема о свертке* . . . . .	238
5.12.	Итоги и обсуждение . . . . .	240
5.13.	Литература . . . . .	241
Глава 6.	<b>Цвет и освещенность</b> . . . . .	244
6.1.	Физические свойства цвета . . . . .	245
6.1.1.	Восприятие освещенных объектов . . . . .	246
6.1.2.	Дополнительные факторы . . . . .	247
6.1.3.	Чувствительность рецепторов . . . . .	248
6.2.	Цветовая система RGB . . . . .	249
6.3.	Другие цветовые системы . . . . .	252
6.3.1.	Субтрактивная цветовая система CMY . . . . .	252
6.3.2.	Цветовая система HSI: Тон-Насыщенность-Интенсивность (Hue-Saturation-Intensity) . . . . .	252



6.3.3.	Телевизионные цветовые системы YIQ и YUV . . . . .	256
6.3.4.	Классификация с использованием цветовой информации. . . . .	257
6.4.	Цветовые гистограммы. . . . .	259
6.5.	Сегментация цветных изображений . . . . .	261
6.6.	Освещенность . . . . .	264
6.6.1.	Излучение одиночного источника света. . . . .	264
6.6.2.	Диффузное отражение. . . . .	266
6.6.3.	Зеркальное отражение . . . . .	267
6.6.4.	Потемнение при увеличении расстояния . . . . .	269
6.6.5.	Дополнительные факторы. . . . .	270
6.6.6.	Модель освещения Фонга*. . . . .	271
6.6.7.	Учет освещенности зрительной системой человека . . . . .	271
6.7.	Смежные темы* . . . . .	272
6.7.1.	Приложения . . . . .	272
6.7.2.	Человеческое цветовое восприятие. . . . .	272
6.7.3.	Многоспектральные изображения . . . . .	273
6.7.4.	Классификационные изображения . . . . .	273
6.8.	Литература . . . . .	274
<b>Глава 7.</b>	<b>Текстура . . . . .</b>	<b>276</b>
7.1.	Текстура, текстелы и текстурные статистики . . . . .	277
7.2.	Тексельное описание текстур. . . . .	278
7.3.	Количественные характеристики текстур . . . . .	280
7.3.1.	Плотность и направление краев . . . . .	280
7.3.2.	Локальное двоичное разбиение. . . . .	282
7.3.3.	Вычисление характерных признаков на основе матрицы вхождений . . . . .	282
7.3.4.	Энергетические текстурные характеристики Лавса. . . . .	285
7.3.5.	Автокорреляция и спектр мощности . . . . .	287
7.4.	Текстурная сегментация. . . . .	289
7.5.	Литература . . . . .	291
<b>Глава 8.</b>	<b>Поиск изображений на основе содержания . . . . .</b>	<b>293</b>
8.1.	Примеры баз данных изображений. . . . .	294
8.2.	Запросы к базам данных изображений . . . . .	295
8.3.	Запросы по образцу . . . . .	297
8.4.	Меры расстояния между изображениями . . . . .	298
8.4.1.	Характеристики цветового сходства. . . . .	299
8.4.2.	Характеристики текстурного сходства. . . . .	303
8.4.3.	Характеристики сходства формы. . . . .	305
8.4.4.	Характеристики сходства, учитывающие присутствующие на изображении объекты и их взаимное расположение . . . . .	313
8.5.	Организация базы данных. . . . .	318
8.5.1.	Стандартные индексы . . . . .	318
8.5.2.	Пространственная индексация . . . . .	322

8.5.3. Индексация для систем поиска изображений на основе содержания, в которых применяется несколько различных мер расстояния . . . . .	322
8.6. Литература . . . . .	324
<b>Глава 9. Движение на двумерных изображениях . . . . .</b>	<b>327</b>
9.1. Явления и прикладные задачи, связанные с движением на изображениях . . . . .	328
9.2. Вычитание изображений . . . . .	329
9.3. Вычисление векторов перемещения . . . . .	331
9.3.1. Игра Decathlete . . . . .	333
9.3.2. Использование соответствующих точек . . . . .	334
9.3.3. Алгоритм MPEG для сжатия потока видеоданных . . . . .	339
9.3.4. Вычисление потока изображения* . . . . .	341
9.3.5. Уравнение потока изображения* . . . . .	342
9.3.6. Распространение ограничений при вычислении потока изображения* . . . . .	344
9.4. Вычисление траекторий движущихся точек . . . . .	345
9.4.1. Слежение с учетом априорных знаний из предметной области . . . . .	351
9.5. Обнаружение значительных изменений условий видеосъемки . . . . .	353
9.5.1. Сегментация видеопоследовательностей . . . . .	354
9.5.2. Игнорирование некоторых эффектов камеры . . . . .	357
9.5.3. Хранение фрагментов видеопоследовательностей . . . . .	359
9.6. Литература . . . . .	359
<b>Глава 10. Сегментация изображений . . . . .</b>	<b>362</b>
10.1. Обнаружение областей . . . . .	363
10.1.1. Методы кластеризации . . . . .	364
10.1.2. Нарращивание областей . . . . .	375
10.2. Способы представления областей . . . . .	377
10.2.1. Оверлейные представления . . . . .	377
10.2.2. Маркированные изображения . . . . .	378
10.2.3. Кодирование границ . . . . .	379
10.2.4. Квадрантные деревья . . . . .	380
10.2.5. Таблицы свойств . . . . .	381
10.3. Обнаружение контуров . . . . .	382
10.3.1. Прослеживание границ существующих областей . . . . .	382
10.3.2. Детектор и компоновщик краев Кэнни . . . . .	386
10.3.3. Группировка согласующихся соседних контурных фрагментов в кривые . . . . .	387
10.3.4. Преобразование Хафа для обнаружения прямых и дуг окружностей . . . . .	393
10.4. Подбор моделей сегментов . . . . .	403
10.5. Обнаружение высокоуровневых структур . . . . .	410
10.5.1. Ленты . . . . .	410
10.5.2. Обнаружение углов . . . . .	413

10.6.	Сегментация на основе согласованного движения . . . . .	414
10.6.1.	Границы движущихся объектов . . . . .	414
10.6.2.	Накопление траекторий движения . . . . .	415
10.7.	Литература . . . . .	418
<b>Глава 11.</b>	<b>Сопоставление в двумерном пространстве . . . . .</b>	<b>421</b>
11.1.	Совмещение двумерных данных . . . . .	422
11.2.	Представление точек . . . . .	424
11.3.	Аффинные геометрические преобразования . . . . .	425
11.4.	Наилучшее двумерное аффинное преобразование* . . . . .	438
11.5.	Распознавание двумерных объектов с использованием аффинных преобразований . . . . .	439
11.6.	Распознавание двумерных объектов с использованием реляционных моделей . . . . .	451
11.7.	Нелинейные методы деформации изображений . . . . .	469
11.8.	Итоги . . . . .	473
11.9.	Литература . . . . .	474
<b>Глава 12.</b>	<b>Восприятие трехмерных сцен по двумерным изображениям . . . . .</b>	<b>476</b>
12.1.	Внутренние изображения . . . . .	477
12.2.	Маркировка контурных изображений объектов с плоскими гранями . . . . .	483
12.3.	Трехмерные признаки на двумерных изображениях . . . . .	491
12.4.	Другие способы определения пространственных свойств объектов . . . . .	497
12.4.1.	Определение формы объектов по одному признаку . . . . .	497
12.4.2.	Точки схода . . . . .	502
12.4.3.	Определение глубины с помощью фокусировки . . . . .	503
12.4.4.	Признаки, связанные с движением . . . . .	504
12.4.5.	Контурные и виртуальные прямые . . . . .	504
12.4.6.	Совмещенность . . . . .	505
12.5.	Модель формирования изображений на основе перспективной проекции . . . . .	506
12.6.	Определение глубины с помощью стереоскопической системы . . . . .	509
12.6.1.	Обнаружение соответствующих признаков . . . . .	513
12.7.	Формула тонкой линзы* . . . . .	518
12.8.	Итоги . . . . .	522
12.9.	Литература . . . . .	522
<b>Глава 13.</b>	<b>Восприятие трехмерных сцен. Оценка пространственного положения и ориентации объектов . . . . .</b>	<b>526</b>
13.1.	Устройство стереоскопической системы компьютерного зрения общего назначения . . . . .	527
13.2.	Аффинные преобразования в трехмерном пространстве . . . . .	530
13.2.1.	Системы координат . . . . .	530
13.2.2.	Перенос . . . . .	531
13.2.3.	Масштабирование . . . . .	532
13.2.4.	Поворот . . . . .	532
13.2.5.	Поворот вокруг произвольной оси . . . . .	535
13.2.6.	Совмещение фигур посредством аффинных преобразований . . . . .	537

13.3. Модель камеры . . . . .	539
13.3.1. Матрица перспективной проекции . . . . .	540
13.3.2. Прямоугольная и слабая перспективная проекции . . . . .	544
13.3.3. Вычисление трехмерных координат с использованием нескольких камер . . . . .	547
13.4. Наилучшая аффинная калибровочная матрица . . . . .	549
13.4.1. Калибровочный стенд . . . . .	550
13.4.2. Задача для решения методом наименьших квадратов . . . . .	550
13.4.3. Обсуждение аффинного метода . . . . .	555
13.5. Использование структурной подсветки . . . . .	557
13.6. Простая процедура оценки положения наблюдаемого объекта . . . . .	560
13.7. Улучшенный метод калибровки камеры* . . . . .	567
13.7.1. Внутренние параметры камеры . . . . .	567
13.7.2. Внешние параметры камеры . . . . .	568
13.7.3. Пример калибровки . . . . .	573
13.8. Оценка положения объектов* . . . . .	578
13.8.1. Оценка положения на основе соответствия двумерных и трех- мерных координат точек . . . . .	578
13.8.2. Линейная оптимизация при наличии ограничений . . . . .	580
13.8.3. Вычисление преобразования $\text{Tr} = \{\mathbf{R}, \mathbf{T}\}$ . . . . .	581
13.8.4. Верификация и оптимизация при вычислении положения объектов . . . . .	583
13.9. Реконструкция трехмерных объектов . . . . .	584
13.9.1. Сбор дальнометрических данных . . . . .	585
13.9.2. Совмещение видов . . . . .	588
13.9.3. Реконструкция поверхности . . . . .	589
13.9.4. Алгоритм объемного удаления . . . . .	590
13.10. Вычисление формы по данным освещенности . . . . .	594
13.10.1. Стереофотометрический метод . . . . .	598
13.10.2. Интегральный учет пространственных ограничений . . . . .	599
13.11. Геометрическая структура объектов по данным о движении . . . . .	600
13.12. Литература . . . . .	603

<b>Глава 14. Трехмерные модели. Распознавание объектов на изображениях на основе моделей . . . . .</b>	<b>607</b>
14.1. Обзор распространенных разновидностей моделей . . . . .	608
14.1.1. Трехмерные каркасные модели . . . . .	608
14.1.2. Модели типа «поверхность-ребро-вершина» . . . . .	609
14.1.3. Модели на основе обобщенных цилиндров . . . . .	612
14.1.4. Модели на основе октантных деревьев . . . . .	615
14.1.5. Модели на основе суперквадрик . . . . .	615
14.2. Модели на основе классов видимости как альтернатива истинным трехмерным моделям . . . . .	618
14.3. Физические и деформационные модели . . . . .	619
14.3.1. Модели на основе активных контуров (snakes-модели) . . . . .	620
14.3.2. Трехмерные оболочечные модели . . . . .	624
14.3.3. Моделирование движения человеческого сердца . . . . .	626

14.4.	Основные методы распознавания трехмерных объектов . . . . .	627
14.4.1.	Распознавание по трехмерным моделям с использованием процедуры совмещения . . . . .	629
14.4.2.	Распознавание на основе сопоставления исходных данных с реляционными моделями . . . . .	639
14.4.3.	Распознавание на основе сопоставления с функциональными моделями . . . . .	651
14.4.4.	Распознавание по внешнему виду объектов . . . . .	654
14.5.	Литература . . . . .	663
<b>Глава 15.</b>	<b>Системы виртуальной реальности . . . . .</b>	<b>667</b>
15.1.	Основные признаки систем виртуальной реальности . . . . .	668
15.2.	Приложения систем виртуальной реальности . . . . .	669
15.3.	Системы дополненной реальности . . . . .	671
15.4.	Дистанционное управление . . . . .	674
15.5.	Устройства для систем виртуальной реальности . . . . .	677
15.6.	Краткий обзор датчиков для систем виртуальной реальности . . . . .	682
15.7.	Генерация простых трехмерных моделей . . . . .	685
15.8.	Сочетание реальных и искусственных изображений . . . . .	687
15.9.	Психофизиологические аспекты человеко-машинного интерфейса . . . . .	691
15.10.	Литература . . . . .	692
<b>Глава 16.</b>	<b>Примеры прикладных задач . . . . .</b>	<b>694</b>
16.1.	Veggie Vision: система для распознавания овощей и фруктов . . . . .	694
16.1.1.	Прикладная область и требования к системе . . . . .	695
16.1.2.	Устройство системы . . . . .	696
16.1.3.	Процедура идентификации . . . . .	698
16.1.4.	Более подробное описание процесса обработки . . . . .	698
16.1.5.	Производительность . . . . .	703
16.2.	Идентификация личности человека по радужной оболочке глаза . . . . .	704
16.2.1.	Требования к системам идентификации личности . . . . .	705
16.2.2.	Устройство системы . . . . .	707
16.2.3.	Производительность системы . . . . .	712
16.3.	Литература . . . . .	713
	<b>Аннотированный список литературы, добавленный при переводе . . . . .</b>	<b>714</b>
1.	Распознавание образов и компьютерное зрение . . . . .	714
2.	Представление и обработка изображений . . . . .	714
3.	Компьютерная графика и геометрическое моделирование . . . . .	715
4.	Смежные вопросы . . . . .	716
	<b>Предметный указатель . . . . .</b>	<b>717</b>



Рис. 6.1



Рис. 6.9



Рис. 6.10

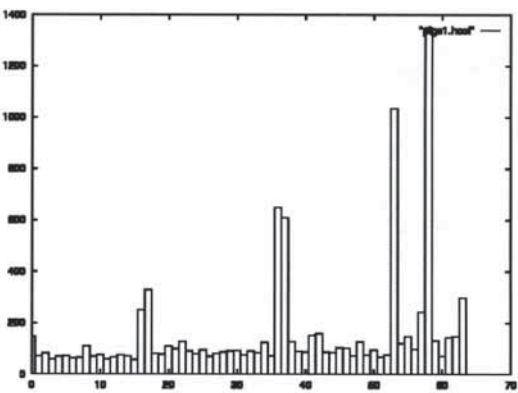
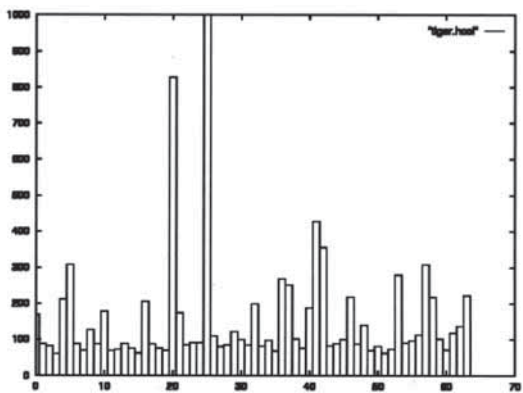


Рис. 6.11

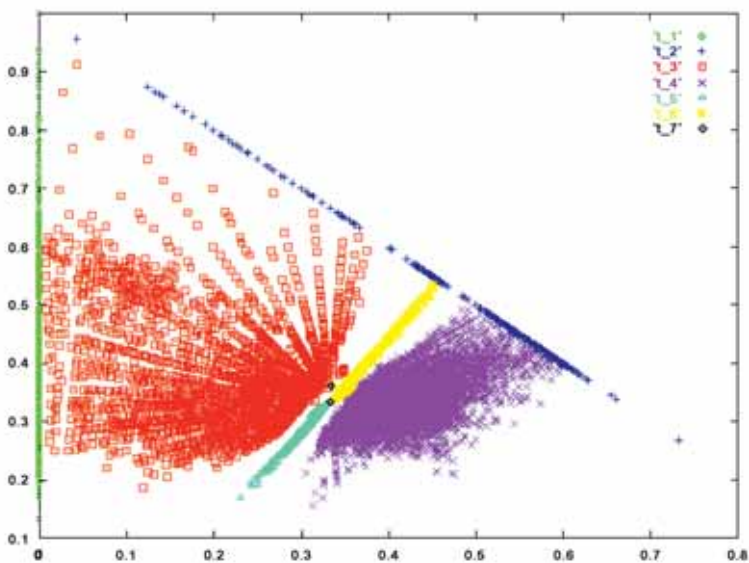


Рис. 6.12





Рис. 6.13

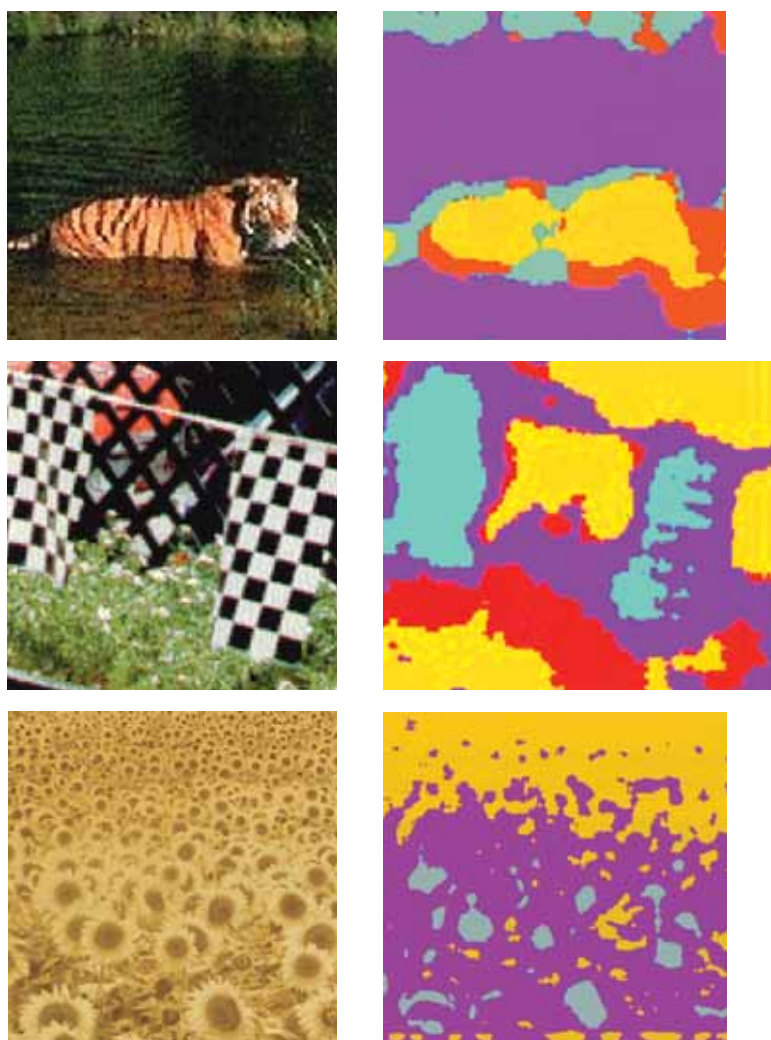


Рис. 7.8





Рис. 8.1









Images 1-8 out of 41			
			
<a href="#">view full size</a>	<a href="#">view full size</a>	<a href="#">view full size</a>	<a href="#">view full size</a>
			
<a href="#">view full size</a>	<a href="#">view full size</a>	<a href="#">view full size</a>	<a href="#">view full size</a>
<b>Columns:</b>		<b>Rows:</b>	

Рис. 8.3







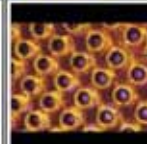

Images 1-8 out of 50			
			
<a href="#">view full size</a>	<a href="#">view full size</a>	<a href="#">view full size</a>	<a href="#">view full size</a>
			
<a href="#">view full size</a>	<a href="#">view full size</a>	<a href="#">view full size</a>	<a href="#">view full size</a>

Рис. 8.4

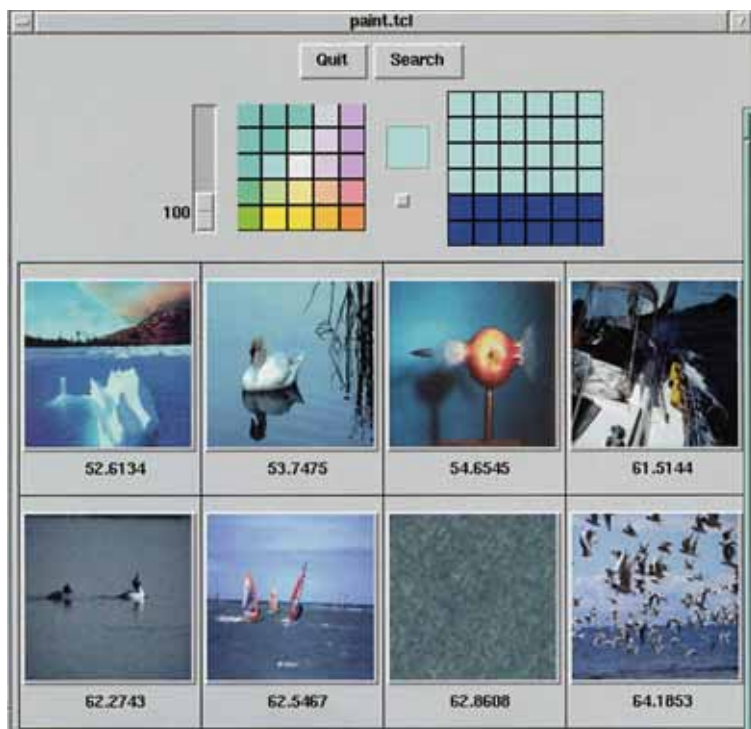


Рис. 8.5

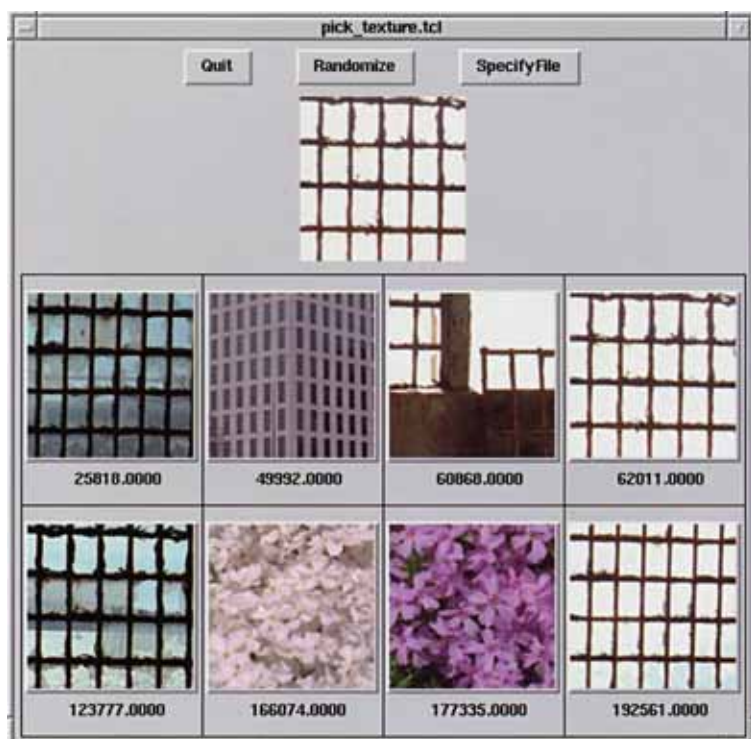


Рис. 8.6



Рис. 8.7

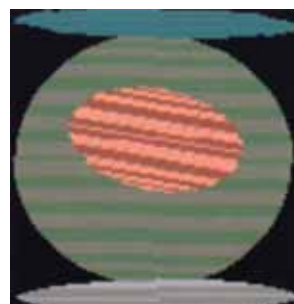


Рис. 8.9

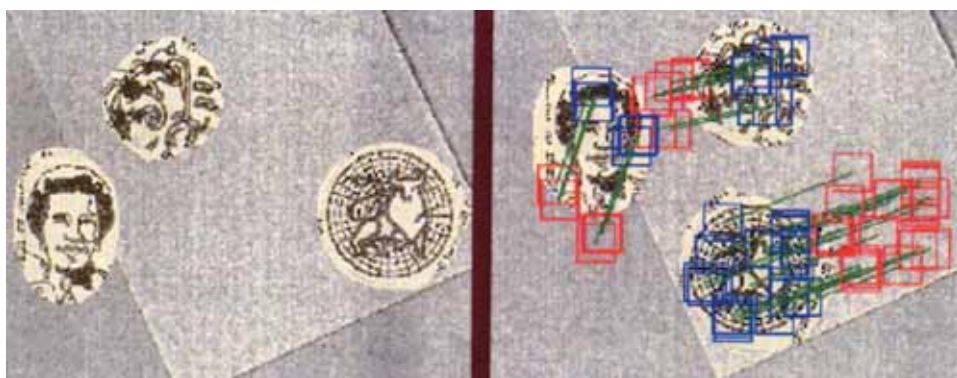


Рис. 9.6



Рис. 10.1

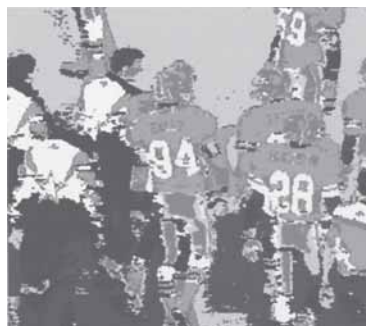


Рис. 10.4



Рис. 10.5





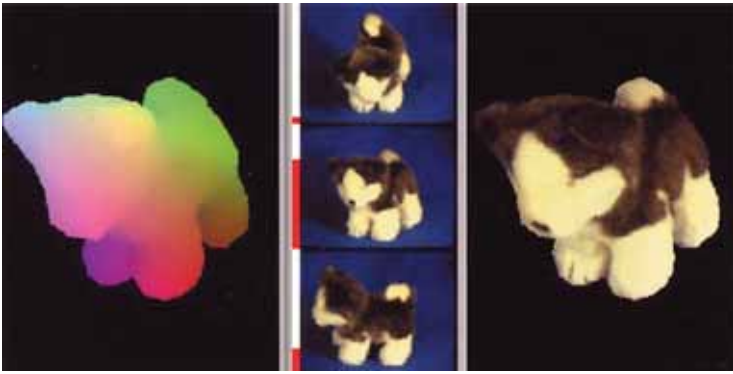


Рис. 15.17

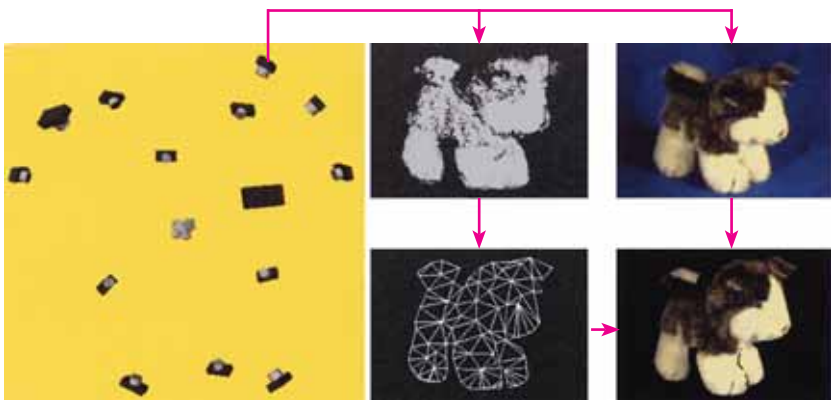


Рис. 15.18



Рис. 15.19

*Минимальные системные требования определяются соответствующими требованиями программы Adobe Reader версии не ниже 11-й для платформ Windows, Mac OS, Android, iOS, Windows Phone и BlackBerry; экран 10"*

*Учебное электронное издание*

Серия: «Лучший зарубежный учебник»

**Шапиро Линда  
Джордж Стокман**

## **КОМПЬЮТЕРНОЕ ЗРЕНИЕ**

Подписано к использованию 19.03.15. Формат 145×210 мм

Издательство «БИНОМ. Лаборатория знаний»

125167, Москва, проезд Аэропорта, д. 3

Телефон: (499) 157-5272

e-mail: [info@pilotLZ.ru](mailto:info@pilotLZ.ru), <http://www.pilotLZ.ru>

