

See discussions, stats, and author profiles for this publication at:  
<https://www.researchgate.net/publication/272487320>

# Цифровые изображения: от получения до обработки

BOOK · MAY 2014

---

READS

922

## 1 AUTHOR:



Valery Starovoitov

United Institute of Inform...

62 PUBLICATIONS 234

CITATIONS

SEE PROFILE

**В.В. СТАРОВОЙТОВ  
Ю.И. ГОЛУБ**

## **ЦИФРОВЫЕ ИЗОБРАЖЕНИЯ: ОТ ПОЛУЧЕНИЯ ДО ОБРАБОТКИ**



Минск 2014

УДК 004; 004.93'1; 004.932

**Старовойтов В.В.** Цифровые изображения: от получения до обработки / В.В. Старовойтов, Ю.И. Голуб – Минск: ОИПИ НАН Беларуси, 2014. – 202 с. - ISBN 978-985-6744-80-1.

Цель данной монографии – систематизировать главные понятия и способы формирования цифровых изображений, основы их цифровой обработки. В книге даются понятия изображения, цифрового изображения, излагаются базовые методы получения и обработки цифровых изображений. Приведены примеры решения прикладных задач, в основе которых лежит последовательное применение описанных алгоритмов, представлены популярные инструментарии обработки изображений – система Matlab и библиотека OpenCV.

Книга будет полезна специалистам и разработчикам, а также аспирантам и студентам, занимающимся обработкой цифровых изображений и анализом данных, представленных в виде изображений.

Табл. 12. Ил. 137. Библиогр.: 27 назв.

Печатается по решению редакционной коллегии Объединенного института проблем информатики Национальной академии наук Беларусь, бюро Отделения физики, математики и информатики и редакционно-издательской комиссии Национальной академии наук Беларусь.

Рецензенты:

доктор технических наук М.М. Татур  
доктор технических наук А.А. Дудкин

**ISBN 978-985-6744-80-1**

© В.В. Старовойтов, Ю.И. Голуб, 2014  
© Оформление ГНУ «Объединенный  
институт проблем информатики  
Национальной академии наук  
Беларусь, 2014

# ОГЛАВЛЕНИЕ

<b>ВВЕДЕНИЕ .....</b>	<b>6</b>
<b>ЧАСТЬ I. ПОЛУЧЕНИЕ ЦИФРОВЫХ ИЗОБРАЖЕНИЙ .....</b>	<b>9</b>
<b>Глава 1. Что такое изображение.....</b>	<b>9</b>
1.1. Изображение как многомерный сигнал.....	10
1.2. Зрительная система человека.....	11
1.3. Модели цветного изображения .....	15
1.3.1. Трехкомпонентная теория цвета.....	15
1.3.2. Оппонентная теория цвета.....	18
<b>Глава 2. Что такое цифровое изображение .....</b>	<b>18</b>
2.1. Дискретизация изображений .....	19
2.2. Квантование изображений .....	20
2.3. Кодирование и типы цифровых изображений .....	24
2.4. Сжатие изображений .....	25
2.4.1. Алгоритмы сжатия без потерь.....	26
2.4.2. Алгоритмы сжатия с потерями.....	27
2.5. Форматы представления цифровых изображений .....	38
2.6. Видео и форматы видео.....	41
<b>Глава 3. Регистрация цифровых изображений.....</b>	<b>43</b>
3.1. Пиксель как регистрирующий сенсор .....	43
3.2. Сканер .....	47
3.3. Цифровые фотоаппараты .....	49
3.4. Получение изображений с помощью ультразвука .....	54
3.5. Рентгеновские изображения .....	56
3.6. Томографические изображения.....	58
3.6.1. Магнитно-резонансная томография.....	58
3.6.2. Компьютерная томография.....	59
3.6.3. Отличия изображений компьютерной томографии от магнитно-резонансной.....	61
3.7. Изображения дистанционного зондирования Земли .....	61
3.7.1. Мультиспектральные изображения.....	63
3.7.2. Гиперспектральные изображения.....	64
3.7.3. Радиолокационные изображения .....	66
<b>Выводы по части I.....</b>	<b>72</b>

<b>ЧАСТЬ II. ОБРАБОТКА ЦИФРОВЫХ ИЗОБРАЖЕНИЙ.....</b>	<b>73</b>
<b>Глава 4. Классификация методов цифровой обработки изображений .....</b>	<b>73</b>
<b>Глава 5. Геометрические преобразования изображений .....</b>	<b>76</b>
5.1. Аффинные преобразования .....	77
5.2. Проективные преобразования .....	80
5.3. Уточнение координат и значений яркости пикселов после преобразования .....	83
5.4. Геометрическая коррекция изображений.....	83
5.4.1. Построение мозаики из изображений.....	83
5.4.2. Коррекция проективных искажений .....	87
5.5. Нелинейная геометрическая коррекция .....	88
5.5.1. Общий подход к вычислению параметров преобразования .....	88
5.5.2. Коррекция дисторсии .....	90
<b>Глава 6. Преобразования изображений точечного типа .....</b>	<b>93</b>
6.1. Логические операции над изображениями.....	93
6.2. Линейные преобразования яркости полутоновых изображений .....	93
6.3. Арифметические операции над изображениями .....	98
6.4. Нелинейная коррекция яркости изображений .....	99
6.5. Преобразования яркости на базе гистограммы изображения.....	105
6.6. Локально-адаптивная обработка изображений.....	107
<b>Глава 7. Преобразования локального типа .....</b>	<b>111</b>
7.1. Типы шумов.....	111
7.2. Фильтрация изображений .....	113
7.3. Низкочастотные фильтры .....	116
7.4. Высокочастотные фильтры.....	120
7.5. Нелинейная фильтрация.....	124
7.6. Внесение эффектов в изображение .....	129
7.7. «Продвинутые» алгоритмы.....	131
<b>Глава 8. Операции математической морфологии .....</b>	<b>132</b>
8.1. Математическая морфология бинарных изображений .....	134
8.1.1. Эрозия и дилатация .....	134
8.1.2. Отмыкание и замыкание.....	139
8.1.3. Операция утоньшения .....	140
8.1.4. Модифицированный алгоритм Зонга – Суня .....	141

8.2. Математическая морфология полутоновых изображений .....	142
<b>Глава 9. Обработка цветных изображений.....</b>	<b>144</b>
9.1. Коррекция цвета на изображениях .....	144
9.2. Автоматическое вычисление порога бинаризации цветного изображения.....	146
9.3. Повышение четкости цветного портрета .....	148
<b>Глава 10. Сегментация изображений .....</b>	<b>151</b>
10.1. Сегментация изображений по яркости на основе пороговой обработки .....	152
10.2. Статистический подход к текстурной сегментации.....	157
10.3. Сегментация методом морфологического водораздела.....	160
Выводы по части II.....	161
<b>ЧАСТЬ III. ПРИМЕРЫ ПРИМЕНЕНИЯ ЦИФРОВОЙ ОБРАБОТКИ ИЗОБРАЖЕНИЙ.....</b>	<b>162</b>
<b>Глава 11. Решение прикладных задач .....</b>	<b>162</b>
11.1. Повышение контраста рентгеновских изображений.....	162
11.2. Морфологическая фильтрация бинарных изображений....	163
11.3. Коррекция неравномерного фона изображения и выделение объектов .....	164
11.4. Повышение разрешения мультиспектрального изображения.....	166
11.5. Детекция предупреждающих дорожных знаков.....	168
11.6. Автоматическое определение нефтяных пятен .....	170
11.7. Контроль отверстий в балках .....	171
11.8. Улучшение изображений в системах видеонаблюдения ...	172
<b>Глава 12. Программные средства обработки цифровых изображений .....</b>	<b>174</b>
12.1. Использовать готовый код библиотеки или написать собственный? .....	174
12.2. Система Matlab .....	177
12.3. Открытая библиотека OpenCv .....	180
Выводы по части III .....	181
<b>ЗАКЛЮЧЕНИЕ .....</b>	<b>183</b>
<b>СПИСОК ЛИТЕРАТУРЫ .....</b>	<b>184</b>
<b>ГЛОССАРИЙ.....</b>	<b>186</b>
<b>ПРИЛОЖЕНИЕ. Преобразование цветовых моделей друг в друга .....</b>	<b>196</b>

## **ВВЕДЕНИЕ**

Изображение – это отражение естественных, возникающих в природе сигналов, т. е. волн различной длины, которые могут быть зарегистрированы зрительной системой человека или приборами. Причем глаз человека воспринимает волны в достаточно узком участке электромагнитного спектра, который называется видимым. Все видимое глазом, а также регистрируемое приборами, изначально представляет собой непрерывные сигналы, образующие аналоговые изображения. Они описываются непрерывными функциями, а современные компьютеры обрабатывают информацию в цифровом виде. Поэтому обработка изображений является одним из самых востребованных направлений современной информатики.

Процесс регистрации отраженного непрерывного сигнала и преобразование его в цифровое представление называется получением цифрового изображения. Преобразование также называют обработкой изображений, она выполняется по определенным алгоритмам с помощью программ, реализующих эти алгоритмы.

В настоящей книге авторы попытались минимальными средствами описать главные способы получения цифровых изображений, опустили математическое и физическое описание процессов формирования изображений, поскольку чаще всего пользователь не может изменить способ получения изображений, ему приходится работать с уже полученными данными. Акцент сделан на описание основных методов и алгоритмов обработки, демонстрацию их применения для решения реальных прикладных задач посредством последовательного применения базовых алгоритмов.

Следует отметить, что в научно-технической литературе часто используются общенаучные термины, которые не имеют однозначного определения, а словари также дают разные толкования этих понятий. Например, метод и алгоритм – синонимы, а в другом источнике это разные понятия. В объектно-ориентированном программировании понятие метод деградировало до следующего определения: *метод – это функция или процедура, принадлежащая какому-то классу или объекту*.

Авторы предлагают свою интерпретацию некоторых терминов. Следует отметить, что информационные технологии – это область человеческой деятельности, балансирующая между наукой (в частности, математикой) и техникой (конкретной реализацией научных идей), поэтому

му здесь не совсем строгие определения также представляют некий научно-технический симбиоз.

Базу информационных технологий составляют данные и информация. Поясним, в чем разница. Пусть имеется множество фактов, они могут быть представлены как данные в форме фраз, изображений, звукозаписи и т. п. *Данные* – это любые сведения, представленные в пригодном для понимания и обработки виде. Основными типами данных являются графические (видео, фото, чертеж) и символьные (числа, буквы). Формы представления данных: аналоговая (надпись на бумаге), дискретная (фильм на кинопленке имеет дискретность по временной шкале) и цифровая (песня в формате MP3). Одни и те же цифровые данные можно записать в разных форматах, например изображение в формате JPEG, число в формате с плавающей запятой, бинарные данные в виде нулей и единиц [14].

*Информация* – это данные, в которых устранена избыточность их описания, т. е. это обработанные данные. Описание одного и того же факта может быть представлено данными разного типа (текстом, звуком, изображением). Информацию в отличие от данных можно измерить согласно теории К. Шеннона. Основная форма представления информации – цифровая. В цифровом виде данные могут быть обработаны на ЭВМ. Они представляются нулями и единицами, а их интерпретация зависит от описывающего формата, т. е. один и тот же набор нулей и единиц может быть фрагментом изображения, песни или книги.

В результате абстрагирования данных одного или нескольких классов возникают *знания*. Например, если нагреть алюминиевую деталь, то она расплавится. Стальная деталь тоже расплавится, но при большей температуре. Это знания, полученные в результате наблюдений, осмысления разных фактов и их обобщения.

Рассмотрим следующий набор понятий: метод, методика, технология. *Научный метод* – это совокупность способов получения новых знаний и методов решения задач. Метод объясняет, почему и как из набора данных можно получить другие данные или решить определенную задачу. Например, согласно молекулярно-кинетической теории в алюминиевом сплаве при нагреве до  $N$  градусов изменяется молекулярная структура и он становится жидким, поэтому его можно залить в форму, понизить температуру до  $H$  градусов. Связи между молекулами восстанавливаются и можно получить твердый сплав заданной формы. *Метод в информатике* – это обобщенный способ решения задач определенного класса (обосновывающий почему и как в пределах этого класса можно применять данный метод с гарантированным получением результата). Например, метод декомпозиции – основной метод параллельных вычислений: задача разбивается на несколько более простых подзадач, решив

их, можно получить решение исходной задачи путем комбинации решений отдельных подзадач.

*Методика* – это перечень действий для получения некоторого результата, но в отличие от метода она не объясняет, почему надо делать именно так (например, методика обучения плаванию). При применении методики достижение результата не гарантировано, это понятие чаще используется в сфере обучения. Так, применяя некоторую методику, можно и не добиться желаемого результата, но для достижения одного результата могут использоваться разные методики.

В отличие от методики *технология* гарантирует получение запланированного результата при выполнении соответствующих условий (например, технология производства чугуна). *Информационная технология* описывает четко упорядоченную последовательность применения средств получения и хранения, обработки, передачи и получения цифровых данных для создания некоторого информационного продукта. Отсюда *теория информации* – это математическая теория, включающая в себя описание и оценки методов извлечения, передачи, хранения и классификации информации, а *информатика* – область человеческой деятельности, связанная с преобразованием цифровых данных средствами вычислительной техники и взаимодействия этих средств со средой применения.

Ключевое понятие информатики – *алгоритм* – это набор точных инструкций, описывающих порядок действий для решения задачи за конечное время. Алгоритм может базироваться на некотором методе, но он не объясняет, почему надо делать так, а не иначе, и носит чисто прикладной характер. Алгоритм, как правило, не ориентирован на язык программирования и операционную систему, т. е. это набор абстрагированных инструкций универсального назначения.

*Компьютерная программа* – это последовательность инструкций, закодированных на определенном языке программирования, предназначенная для исполнения с помощью ЭВМ в определенной операционной системе, т. е. это реализация алгоритма. Один алгоритм можно реализовать разными способами, даже программируя на одном языке.

# **ЧАСТЬ I. ПОЛУЧЕНИЕ ЦИФРОВЫХ ИЗОБРАЖЕНИЙ**

## **Глава 1. Что такое изображение**

Для корректной обработки цифровых изображений необходимо понимать, что такое изображение и как с помощью современной техники можно его получить, не вникая в узкотехнические детали.

В философском смысле слово «изображение» означает «отражение» и используется для обозначения правдоподобного представления реальных объектов. Уточним некоторые понятия.

*Изображение* – это образ, в той или иной степени подобный (но не идентичный) изображаемому объекту. В области изобразительного искусства художественные произведения (например, рисунок, картина) являются, как правило, субъективным отражением реального мира.

*Оптическое изображение* – это картина, получаемая в результате прохождения через оптическую систему отраженных от объекта световых лучей и воспроизводящая его контуры и детали на некоторой поверхности (чаще всего плоской).

Наш глаз тоже видит различные изображения. *Зрение человека* реализует процесс психофизиологического восприятия изображений объектов окружающего мира, который осуществляется зрительной системой человека, и позволяет получать представление о величине, форме и цвете предметов, их взаимном расположении и расстоянии между ними.

Фотоаппарат и другие устройства могут запечатлеть изображения на пленке, бумаге или в цифровом виде.

Итак, изображение – это то, что может увидеть человеческий глаз или регистрирующая аппаратура. При этом следует различать понятия «изображение сцены» и «изображение объекта». Сцена – это реальный мир или его представление, созданное с помощью компьютера, она состоит из множества трехмерных объектов. Можно сказать, что сцена бесконечна в пространстве. При ее рассмотрении человек выделяет некоторые объекты, которые называются объектами интереса. На одной сцене в разных ситуациях нас интересуют разные типы объектов. То, что видит наш глаз или прибор, обычно является изображением части некоторой сцены.

## 1.1. Изображение как многомерный сигнал

Таким образом, наш глаз (как и регистрирующая аппаратура) воспринимает некий сигнал, отраженный в определенном участке электромагнитного спектра.

*Спектр* – это совокупность значений или их распределение по какому-либо параметру наблюдаемой величины. Поскольку первыми были получены оптические спектры, то до сих пор наиболее часто термин «спектр» употребляется для описания распределения потока излучения или частиц по длинам волн или энергии.

Электромагнитные излучения подразделяются: на радиоволны, инфракрасное излучение, видимый глазом свет, ультрафиолетовое излучение, рентгеновское излучение и гамма-излучение. Между ними нет четких границ, области этих видов излучений могут частично перекрываться. Шкала, в которой они измеряются, называется *электромагнитным спектром* (рис. 1.1). Измерения чаще производятся в метрах.

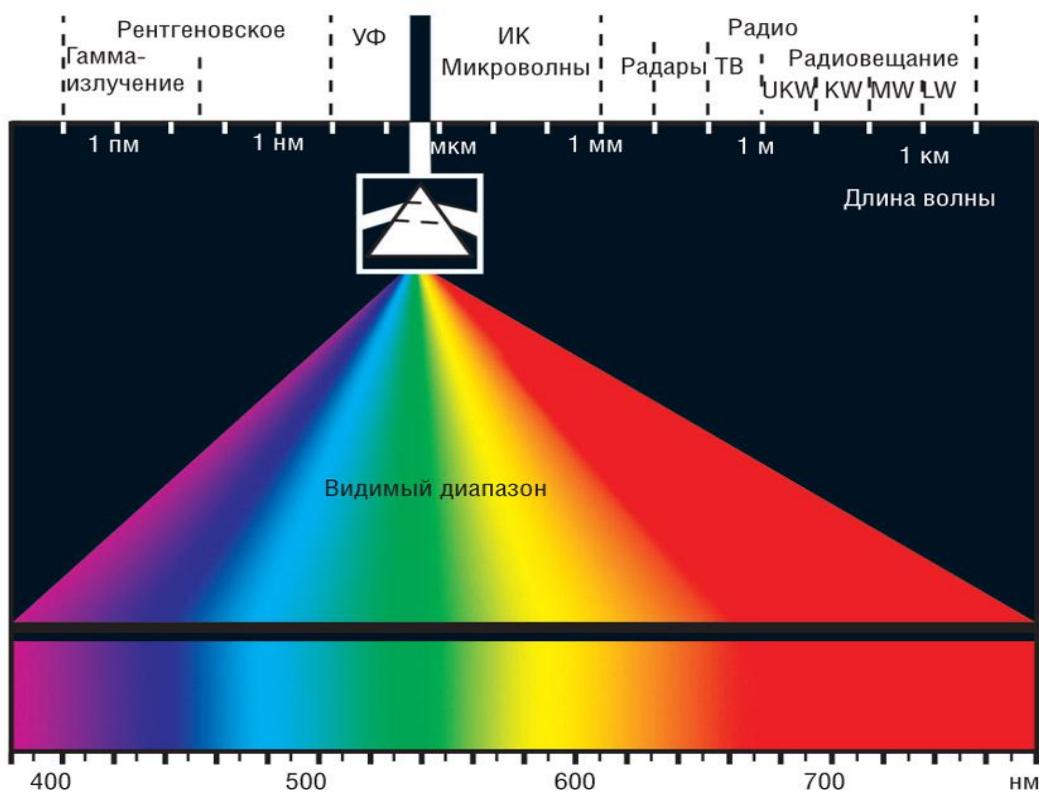


Рис. 1.1. Шкала электромагнитного спектра

Глаз человека воспринимает только часть отраженных волн, которые попадают в так называемый *видимый диапазон* электромагнитного спектра.

## 1.2. Зрительная система человека

До 80 % информации человек получает в виде изображений, поэтому важно понимать механизмы зрительного восприятия.

*Зрительная система* – это оптико-биологическая бинокулярная система, эволюционно возникшая у животных и способная воспринимать излучение видимого участка электромагнитного спектра, создавая изображение в виде сенсорного ощущения положения предметов в пространстве, их размеров и цвета.

Глаз человека обеспечивает адаптивную настройку восприятия изображений в соответствии с внешними условиями. Рассмотрим некоторые свойства зрительной системы человека.

Глаз представляет собой систему линз, формирующую на сетчатке уменьшенное и перевернутое изображение внешнего мира (рис. 1.2), включает в себя роговицу, хрусталик, стекловидное тело, позади которого лежит воспринимающая свет сетчатка.

Важной характеристикой зрительной системы является чувствительность, т. е. способность реагировать на внешние изменения. Она характеризуется верхним и нижним абсолютными порогами. Существует несколько видов чувствительности. *Световая чувствительность* характеризует свойство глаза реагировать на минимальный световой поток. Однако следует отметить, что вероятность распознавания минимального светового потока зависит от разных факторов, например угла зрения.

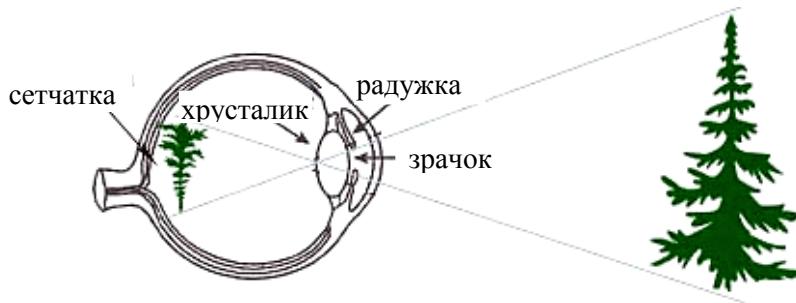


Рис. 1.2. Схема отображения видимого предмета на сетчатке глаза

Зрительная система по-разному реагирует на излучения, которые равны по мощности, но имеют разные длины волн, т. е. принадлежат разным участкам спектра. Такая *чувствительность* называется *спектральной* или *чувствительностью к цветовому тону*. Человеческий глаз воспринимает волны электромагнитного спектра в диапазоне от 380 до 780 нм. Кроме того, зрительная система характеризуется *чувствительностью к насыщенности цвета*.

Способность глаза различать минимальные различия яркости смежных областей изображения характеризуется *контрастной чувствительностью*.

Чувствительность зрительной системы приведенных типов не является постоянной, а зависит от многих факторов, в частности условий освещения. Например, при переходе из темной комнаты в светлую нужно некоторое время для восстановления светочувствительности глаза. Этот процесс называется *яркостной адаптацией глаза*.

Цветоощущение описывают три основные характеристики: *светлота, цветовой тон и насыщенность* [2].

У животных и человека органами зрения являются глаза, а зрение бинокулярно, оно обеспечивает объемное представление изображения. В глазу человека содержатся два типа светочувствительных клеток (рецепторов): высокочувствительные палочки и менее чувствительные колбочки (рис. 1.3). Они преобразуют световые раздражения в нервное возбуждение. Палочки функционируют при свете слабой интенсивности, в сумерках. Зрительные ощущения, получаемые при этом, бесцветны. Колбочки функционируют днем и при ярком освещении. Их функция – определение ощущения цветности.

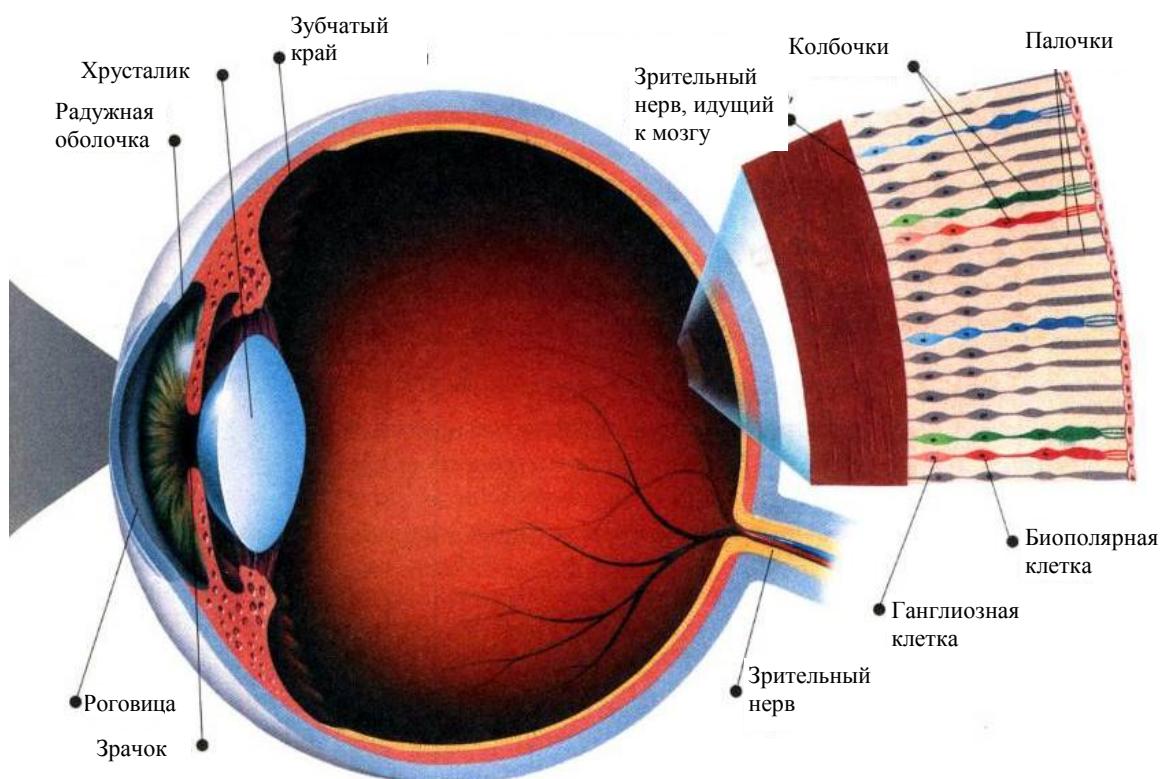


Рис. 1.3. Строение глаза человека

*Палочки* – это периферические отростки светочувствительных клеток сетчатки глаза, названные так за свою вытянутую цилиндрическую форму. В сетчатке глаза человека содержится около 120 млн палочек длиной 0,06 мкм и диаметром 0,002 мкм.

Палочки чувствительны к свету благодаря наличию в них специфического пигмента – родопсина. Под действием света происходят очень быстрые превращения и обесцвечивание зрительного пигмента. Чувствительность палочки достаточна, чтобы зарегистрировать попадание на них даже двух-трех фотонов.

Плотность размещения палочек на различных участках сетчатки глаза неравномерна и может составлять от 20 до 200 тыс. на  $\text{мм}^2$ . Причем на периферии сетчатки их плотность выше, чем к ее середине, что определяет их участие в ночном и периферийном зрении. В центре сетчатки в желтом пятне палочки практически отсутствуют.

Палочки обладают интересной особенностью. В связи с тем что все они содержат один и тот же светочувствительный пигмент родопсин, их спектральная характеристика зависит от уровня освещения. При слабом освещении максимум поглощения родопсина составляет около 500 нм (спектр сумеречного неба) и при этом палочки ответственны за сумеречное зрение, когда цвета предметов неразличимы.

К другому типу фоторецепторов относятся *колбочки*, названы так за свою коническую форму. Они чувствительны к свету благодаря наличию в них пигмента йодопсина. В свою очередь, йодопсин состоит из нескольких зрительных пигментов. На сегодняшний день хорошо исследованы два из них: хлоролаб (чувствительный к желто-зеленой области спектра) и эритролаб (чувствительный к желто-красной части спектра). В сетчатке глаза у взрослого человека со 100 %-м зрением насчитывается около 6–7 млн колбочек с длиной около 50 мкм и диаметром от 1 до 4 мкм. Колбочки приблизительно в 100 раз менее чувствительны к свету, чем палочки, но гораздо лучше воспринимают быстрые изменения света.

Различают три вида колбочек по чувствительности к разным длинам волн света (цветам) (табл. 1.1, рис. 1.4). Колбочки *S*-типа (*Short* – коротковолновый спектр) чувствительны в фиолетово-синей, *M*-типа (*Medium* – средневолновый) – в зелено-желтой и *L*-типа (*Long* – длинноволновый) – в желто-красной областях спектра. Наличие трех видов колбочек и палочек позволяет человеку различать цвета.

Таблица 1.1  
Характеристики различных видов колбочек

Тип колбочек	Воспринимаемые длины волн, нм	Максимум чувствительности, нм
<i>S</i>	380–500	420
<i>M</i>	450–630	534
<i>L</i>	500–750	564

Водоем кажется синим потому, что молекулы воды поглощают все лучи видимого спектра, кроме синих. Это заметно только при его глубине более метра, поэтому лужи не выглядят синими. В ночное время, когда поток фотонов недостаточен для нормальной работы колбочек, зрение обеспечивают только палочки, поэтому ночью человек не может различать цвета.

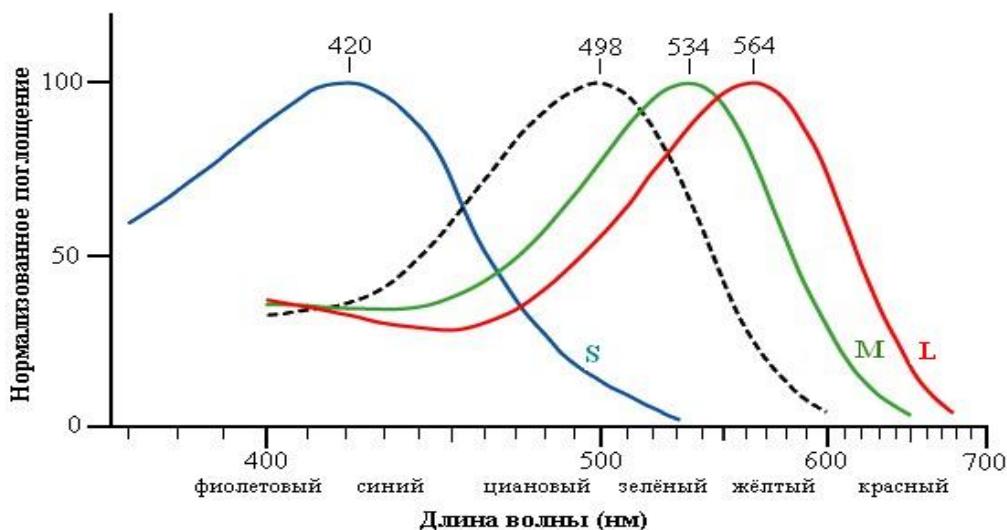


Рис. 1.4. Графики светочувствительности колбочек человеческого глаза *S*, *M*, *L*-типов (пунктиром показана сумеречная, черно-белая восприимчивость палочек)

Распределение типов колбочек в сетчатке неравномерно: «синие» колбочки находятся ближе к периферии, в то время как «красные» и «зеленые» распределены случайным образом. (Это стало известно в 1970-х гг. и было подтверждено более детальными исследованиями в начале XXI в.) Соответствие типов колбочек трем «основным» цветам обеспечивает распознавание человеком тысяч оттенков цветов. Кривые спектральной чувствительности трех видов колбочек частично перекрываются (рис. 1.4), что способствует явлению метамерии. Очень сильный свет возбуждает все три типа рецепторов и потому воспринимается как излучение слепящего-белого цвета.

Равномерное раздражение всех трех элементов, соответствующее средневзвешенному дневному свету, вызывает ощущение белого цвета.

Согласно трехкомпонентной теории цвета за цветовое зрение человека и обезьян отвечают гены, кодирующие светочувствительные белки опсины трех типов, которые реагируют на разные длины волн, что является достаточным для цветового восприятия. У большинства млекопитающих таких генов только два, поэтому они имеют черно-белое зрение. Если у человека два белка, кодируемые разными генами, оказываются слишком схожи или один из белков не синтезируется, у него развивается дальтонизм.

## 1.3. Модели цветного изображения

### 1.3.1. Трехкомпонентная теория цвета

Впервые трехкомпонентную теорию цветового зрения описал М.В. Ломоносов в 1756 г. Сто лет спустя ее развил немецкий ученый Г. Гельмгольц, но он не упомянул работу Ломоносова «О происхождении света», хотя она была опубликована на немецком языке.

Для описания цвета разработано много моделей. Термин «цветовая модель» означает абстрагированное описание цветов в виде набора трех чисел, называемых цветовыми координатами. Множество допустимых оттенков цвета и метод интерпретации этих данных определяют цветовое пространство.

Большинство цветов получаются смешением нескольких основных цветов [19]. Поэтому для описания цвета используется понятие цветовой модели как способа представления большого количества оттенков цвета посредством разложения их на простые составляющие. Возникает вопрос: зачем нужны различные цветовые модели? Одна из причин – излучаемый и поглощаемый цвет описывается по-разному.

В 1931 г. после серии экспериментов по оценке восприятия цвета человеком Международная комиссия по освещению разработала стандарт CIE 1931 XYZ, который описывал цветовое пространство, вмещающее в себя все воспринимаемые человеком цвета [12]. Однако это пространство (для краткости XYZ) было существенно нелинейным.

С появлением цветного телевидения и цветной печати возникла необходимость в разработке новых моделей, ориентированных на разные приложения.

Наиболее популярной цветовой моделью на сегодняшний день является RGB (Red – красный, Green – зеленый, Blue – синий). В ней все три цвета равноправны и независимы. Если три параметра RGB имеют нулевые значения, они описывают черный цвет, если максимальные значения – белый. Эта модель хороша для цветовоспроизведения на устройствах с черным базовым фоном, например на мониторе, который излучает цвет.

RGB – это аддитивная цветовая модель. Отображение реального цвета на мониторе зависит от производителя монитора и настроек.

Если базовый фон белый, модель RGB применять затруднительно (например, белая бумага поглощает цвет). В этом случае удобнее использовать субтрактивную модель, например CMY (Cyan – голубой, Magenta – пурпурный, Yellow – желтый). «Субтрактивный» означает «вычитаемый», т. е. из белого вычтываются основные цвета. Она является дополнительной к модели RGB и фактически определяет негативные оттенки, т. е. от максимального значения яркости в модели RGB (обычно это 255) вычтываются собственно оттенки цвета:  $C = 255 - R$ ,  $M = 255 - G$ ,  $Y =$

= 255 – В. Эту модель активно используют в полиграфии при печати на белой бумаге. Теоретически равное количество краски С=М=Y должно давать серый цвет, однако при печати такой цвет на бумаге имеет бурые оттенки, кроме того, расходуются лишние краски. Поэтому на практике модель дополняют черным цветом, и называется она CMYK. Для перехода из модели CMY в модель CMYK используют следующее соотношение:

$$K = \min(C, M, Y);$$

$$C = C - K;$$

$$M = M - K;$$

$$Y = Y - K.$$

Каждое число в модели CMYK определяет процент краски базового цвета, составляющей цветовую комбинацию. Например, для получения цвета хаки смешивают 30 частей голубой краски, 45 частей пурпурной, 80 – желтой и 5 – черной. Это можно обозначить следующим образом: CMYK = (30, 45, 80, 5) или C30M45Y80K5.

Важно отметить, что числовое значение краски в CMYK не описывает оттенок цвета. Цифры – это набор данных, используемых в печатном процессе для формирования изображения. Реальный цвет зависит от многих факторов: размера пикселя на фотовыводе, растиривания краски, качества бумаги, влажности в цеху, спектральных характеристик источника освещения и др.

В 1976 г. Международной комиссией по освещению была разработана цветовая модель CIE Lab, которая является международным стандартом. Используемые в ней величины означают: L – светлота (изменяется от 0 до 100, т. е. от самого темного до самого светлого); a – красная и зеленая цветности; b – желтая и синяя цветности. В ней значение светлоты отделено от значения хроматической составляющей цвета (тона и насыщенности).

При разработке Lab преследовалась цель создания модели цветового пространства, изменение цвета в котором будет максимально линейным с точки зрения человеческого восприятия, т. е. одинаковое изменение значений координат цвета в разных областях цветового пространства должно производить одинаковое ощущение изменения цвета. Это позволяет математически корректировать нелинейность восприятия цвета человеком.

Модель Lab однозначно определяет цвет, поэтому она применяется в обработке изображений в качестве промежуточного цветового пространства (см. приложение), через которое происходит конвертирование данных между другими цветовыми пространствами (например, из модели RGB сканера в CMYK печатного устройства). При этом представление цвета в Lab активно используется в цветокоррекции, можно отдельно воздействовать на яркость, контраст изображения и его цвет. Недостатком модели Lab является использование кубических корней при пересчете.

Модель HSV (Hue – тон, Saturation – насыщенность, Value – значение) удобна для создания изображений художниками. В ней координатами цвета являются:

Hue – цветовой тон, обычно варьируется в пределах  $0\text{--}360^\circ$ , где  $0^\circ$  – красный,  $120^\circ$  – зеленый,  $240^\circ$  – синий (тон цвета задается в градусах);

Saturation – насыщенность, изменяется в пределах  $0\text{--}100$  или  $0\text{--}1$ ;

Value – яркость, задается в пределах  $0\text{--}100$  или  $0\text{--}1$ .

*Цветовой охват* означает диапазон цветов, который может быть представлен цветовой моделью устройства относительно цветов, воспринимаемых человеческим глазом и моделью Lab. Разными цветовыми охватами обладают электронно-лучевая трубка монитора и телевизора, полиграфические краски и глаз человека. На рис. 1.5 показано сравнение цветовых охватов человеческого глаза (модель Lab), монитора (модель RGB) и печатной машины (модель CMYK) в плоскости, определяемой осями  $x = X/(X + Y + Z)$  и  $y = X/(X + Y + Z)$  в пространстве XYZ.

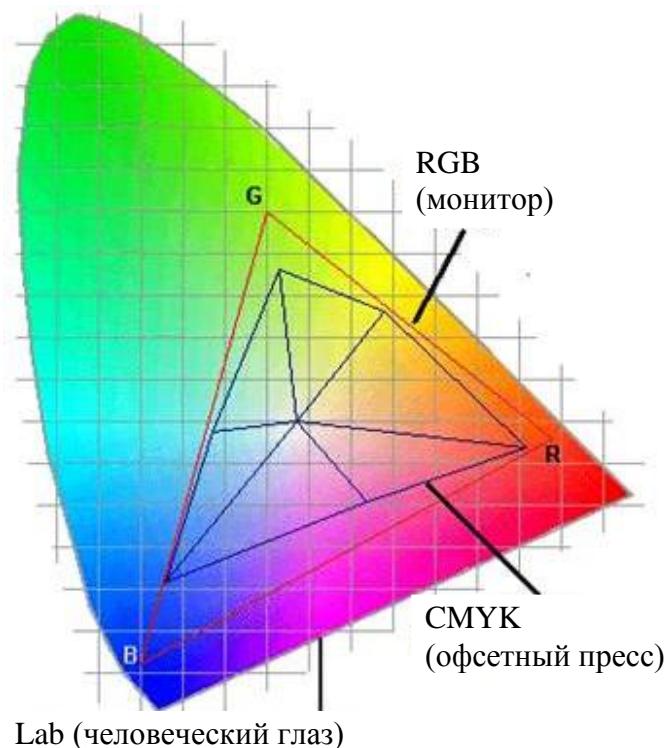


Рис. 1.5. Цветовой охват разных устройств

Разные цветовые модели используются в разных приложениях:  
XYZ и Lab – для описания восприятия цвета человеческим глазом;  
аддитивные модели – для получения цвета на мониторе (например, RGB, sRGB);

полиграфические модели – для получения цвета на бумаге при использовании разных красок и полиграфического оборудования (например, CMYK);

модели, не связанные с физическими процессами оборудования, ориентированы на передачу данных;

математические модели – для цветокоррекции без связи с оборудованием, например HSV.

Математически цветовые модели можно однозначно преобразовать друг в друга (см. приложение), но устройства отображения имеют разные технические возможности и цветовые охваты.

### ***1.3.2. Оппонентная теория цвета***

Параллельно с трехкомпонентной разрабатывалась оппонентная теория цвета Эвальда Геринга [17], которую развили Дэвид Хьюбел и Торстен Визел и получили в 1981 г. Нобелевскую премию. Они предположили, что в мозг поступает информация не о красном (R), зеленом (G) и синем (B) цветах (как в теории цвета Гельмгольца), а о разнице яркости белого ( $I_{max}$ ) и черного ( $I_{min}$ ) цветов, зеленого и красного (G – R), синего и желтого (B – Y). Например, желтый цвет (Y = R + G) – это сумма красного и зеленого цветов. Параметры цвета в этой модели описываются тремя коэффициентами

$$K_{grey} = I_{max} - I_{min}; \quad K_{gr} = G - R; \quad K_{brg} = B - R - G,$$

где  $K_{grey}$ ,  $K_{gr}$ ,  $K_{brg}$  – коэффициенты для любого типа освещения. Оппонентная теория в целом лучше объясняет тот факт, что люди воспринимают цвет предметов одинаково при разных источниках освещения, в том числе при различном цвете источников света в одной сцене (это свойство цветовой адаптации).

Две теории не совсем согласованы друг с другом. Однако большинство ученых предполагает, что на уровне сетчатки действует трехстимульная теория (ее основной представитель – модель RGB), затем информация обрабатывается и в мозг поступают данные, уже согласующиеся с оппонентной теорией.

Следует отметить, что в любой модели (за исключением CMYK) цвет описывается тремя параметрами.

## **Глава 2. Что такое цифровое изображение**

В природе все процессы протекают непрерывно. Зрительная система человека также воспринимает непрерывные сигналы. В современной технике наоборот, все чаще данные представлены в цифровой форме.

Процесс преобразования аналогового сигнала, описывающего изображение, в цифровую форму состоит из трех этапов [20]: дискретизации, квантования и кодирования. Сигнал, к которому применены операции

дискретизации и квантования, называется *цифровым*. После операции дискретизации, но без квантования получаем дискретный сигнал (его значения изменяются в непрерывном диапазоне).

## 2.1. Дискретизация изображений

*Дискретизация* – это преобразование непрерывной функции в дискретную, т. е. значения функции определены в отдельных (дискретных) точках. При этом сами значения могут изменяться в непрерывной шкале. Другими словами, множество допустимых значений дискретной функции может быть бесконечным (рис. 2.1, вертикальная ось).

Замену непрерывного изображения дискретным можно выполнить различными способами [22]. Рассмотрим основные особенности прямоугольной дискретизации.

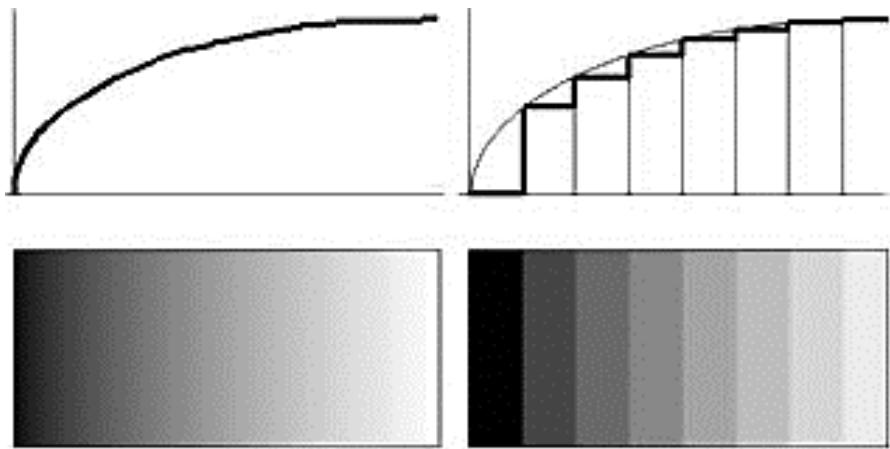


Рис. 2.1. Равномерная дискретизация непрерывной функции яркости

Пусть  $x(t_1, t_2)$  – непрерывное изображение, рассматриваемое как функция от двух переменных;  $x(i_1, i_2)$  – соответствующее ему дискретное изображение, полученное из непрерывного путем разбиения области, в которой определена функция  $x$ , на фрагменты, как правило, прямоугольной формы. Связь между функциями определяется выражением

$$x(i_1, i_2) = x(i_1\Delta t_1, i_2\Delta t_2),$$

где  $\Delta t_1, \Delta t_2$  – вертикальный и горизонтальный интервалы дискретизации. Бесконечное множество точек с координатами  $(t_1, t_2)$  области, в которой определена непрерывная функция яркости, заменяется конечным множеством с координатами  $(i_1, i_2)$ . Обычно  $\Delta t_1 = \Delta t_2$  и этот параметр  $\Delta t$  определяет разрешение изображения. В технике дискретизация определяется количеством точек на дюйм (DPI) или пикселов на дюйм (PPI) и равно целому от деления дюйма на  $\Delta t$ .

Основной вопрос при замене непрерывного изображения дискретным состоит в определении условий, при которых такая замена минимизирует потерю информации, содержащуюся в непрерывном изображении. Потери отсутствуют, если, располагая дискретным сигналом, можно полностью восстановить исходный непрерывный сигнал. С математической точки зрения вопрос заключается в восстановлении непрерывного сигнала в двухмерных промежутках между узлами, в которых его значения известны, т. е. в осуществлении двухмерной интерполяции.

На рис. 2.2 изображены отпечатки пальца при разных интервалах дискретизации. Правое изображение сильно расфокусировано и неверно передает структуру отпечатка, оно непригодно для распознавания.

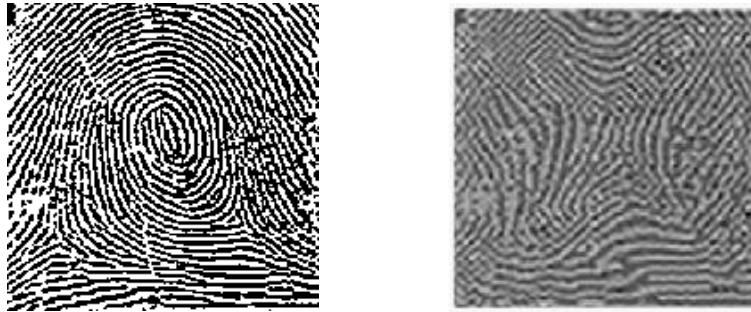


Рис. 2.2. Влияние интервала дискретизации на представление изображения отпечатка пальца

## 2.2. Квантование изображений

При цифровой обработке изображений непрерывный диапазон значений функции яркости делится на несколько поддиапазонов и все значения яркости, попавшие в один поддиапазон, округляются до единого значения, например середины этого диапазона. Такая процедура называется *квантованием по яркости*, и она преобразует непрерывное множество значений функции  $x$  в дискретное множество значений  $x_{\text{кв}}$ . Другими словами, функция принимает конечное множество значений  $\{r_1, \dots, r_L\}$ . Эти значения называются уровнями квантования. В общем случае преобразование выражается ступенчатой функцией (рис. 2.3).

Если значение яркости  $x$  попадает в диапазон  $d_j < x \leq d_{j+1}$ , то оно заменяется на число, равное уровню квантования  $r_j$ , где  $d_j, j = \overline{1, L+1}$ , – пороги квантования. При этом динамический диапазон значений яркости всего изображения ограничен и равен  $[d_1, d_{L+1}]$ . Для простоты кодирования цифрового изображения вместо значения  $r_j$  используется целочисленный номер диапазона  $j$ , начиная с нуля. Таким образом, значения яркости, используемые в описании цвета, например в модели RGB, являются абстрактными безразмерными величинами.

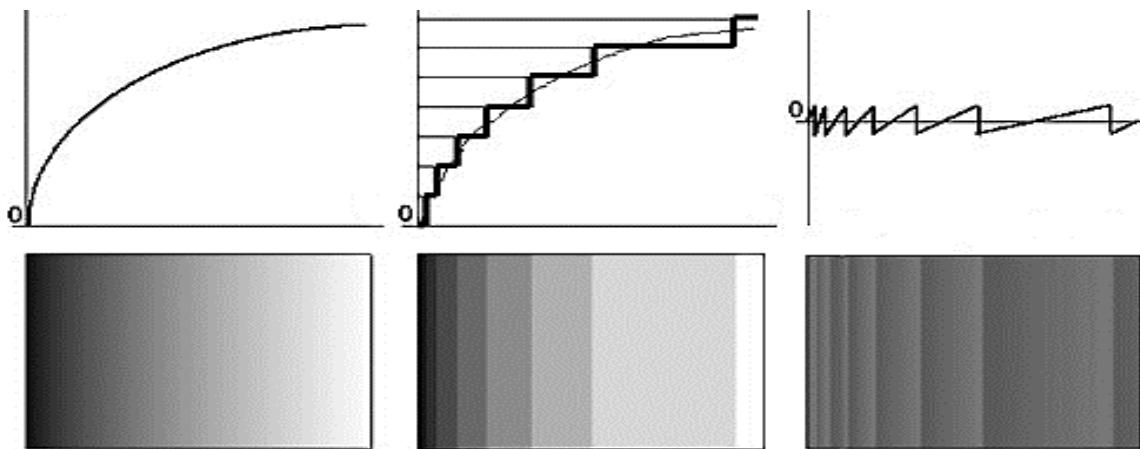


Рис. 2.3. Непрерывный сигнал (вверху слева); функция, описывающая его равномерное квантование (в центре); функция погрешности квантования (справа). Внизу показаны соответствующие этим функциям полутоновые изображения

Не следует путать процедуры квантования и дискретизации и соответственно шаг квантования с шагом дискретизации. При дискретизации выполняется пространственное разбиение области изображения, при квантовании разбивается множество значений яркости.

Квантование можно выполнить разными способами. Рассмотрим самый простой вариант – **квантование методом равномерного разбиения цветового пространства**.

Пусть  $I(x, y)$  – яркость пикселя исходного изображения с координатами  $(x, y)$ ;  $I_{new}(x, y)$  – яркость пикселя нового изображения;  $N_1$  – количество оттенков в исходном изображении (обычно  $N_1 = 256$ );  $N$  – количество элементов в новой палитре (например,  $N = 16$ ).

#### Алгоритм равномерного квантования

**Шаг 1.** Организовать цикл по строкам и столбцам изображения.

**Шаг 2.** В цикле для каждого пикселя  $(x, y)$  вычислить новое значение яркости  $I_{new}(x, y) = I(x, y)/(N_1/N)$ . Результат округлить до целого.

Для цветного изображения повторить вычисления шага 2 для каждого из трех цветов RGB. Разобьем цветовое пространство на равные части вдоль каждой оси. Например, в направлении синей или зеленой оси разобьем куб на восемь частей, а в направлении красной – на четыре. В итоге получается 256 оттенков цвета, равномерно распределенных в пространстве RGB. Каждый оттенок можно закодировать целым индексом от 0 до 255, описав в таблице, какой цвет (т. е. значения RGB) он представляет, получим *индексированное или палитровое цветное изображение*. Цветовая палитра – это фиксированный набор цветов и оттенков, описанный номерами (индексами).

Достоинством этого алгоритма являются простота и высокая скорость. Недостаток – малоконтрастные изображения будут иметь малое количество оттенков, в местах плавных переходов возникают ложные контуры.

**Квантование методом медианного сечения.** Рассмотрим алгоритм, который сформирует палитру так, чтобы каждое значение из нее отвечало равному количеству значений оттенков в исходном изображении. Это достигается путем последовательного разбиения цветового пространства на параллелепипеды со сторонами, параллельными осям цветового пространства RGB.

#### Алгоритм медианного квантования цветовой палитры

Шаг 1. Задать  $N$  – размер палитры. Найти минимальный параллелепипед в пространстве RGB, включающий все значения оттенков цвета пикселов исходного изображения:  $R_{\min}, R_{\max}, G_{\min}, G_{\max}, B_{\min}, B_{\max}$ . Далее разбить параллелепипед на части.

Шаг 2. Выполнить так называемое адаптивное разбиение, состоящее из следующих шагов: выбор самой длинной стороны параллелепипеда, сортировка значений вдоль выбранного направления, нахождение медианного значения среди множества значений вдоль выбранного направления, разделение параллелепипеда по найденной медиане на две части. Таким образом получатся два параллелепипеда, которые содержат примерно одинаковое количество пикселов.

Повторять предыдущую процедуру для каждого из параллелепипедов до тех пор, пока не сформируются  $N$  параллелепипедов, где  $N$  равно заданному размеру палитры.

Если на каком-то этапе окажется, что требуется разделить параллелепипед, содержащий лишь одно значение, то получившийся пустой параллелепипед следует присоединить к самому большому параллелепипеду.

Шаг 3. Заполнить палитру либо центральными значениями RGB параллелепипедов, либо средними арифметическими цветов пикселов, попавших в данные параллелепипеды, округлив их до целых.

Шаг 4. Для каждого пикселя определить, в какой параллелепипед он попал, и заменить его значения RGB параметрами параллелепипеда, вычисленными на шаге 3.

Результатом работы этого алгоритма является неравномерное квантование. Оно лучше предыдущего передает оттенки, при этом скорость работы данного алгоритма высока.

**Квантование изображений методами кластеризации.** Квантование можно выполнять методами, производящими кластеризацию цветовых значений пикселов изображения для построения палитры. В общем

случае *кластеризация* – это процесс разбиения объектов на группы (кластеры) на основе общих свойств объектов. Применимельно к квантованию изображений это означает разбиение всех оттенков заданного изображения (в модели RGB это будет куб) на кластеры так, что внутри каждого кластера находятся лишь близкие значения. В результате также получим неравномерное квантование.

Один из самых популярных алгоритмов кластеризации – это **кластеризация методом  $K$ -средних**. Алгоритм строит  $K$  кластеров, расположенных на максимально больших расстояниях друг от друга. Выбор числа  $K$  может базироваться на результатах предшествующих исследований или интуиции.

Общая идея алгоритма: заданное фиксированное число  $K$  кластеров наблюдения сопоставляется кластерам так, что средние в кластерах максимально отличаются друг от друга.

#### Алгоритм квантования цвета методом $K$ -средних

Исходные данные: цветное изображение в пространстве RGB, число кластеров  $K$ , максимальное число итераций  $N$ .

**Шаг 1.** Определить начальные центры кластеров. Выбор координат центров можно выполнить разными способами: случайно, выбрать значения оттенков R, G, B первых  $K$  пикселов и т.д.

Для цвета каждого пикселя найти ближайший кластер, вычислив расстояние между ним и центрами всех кластеров. В результате цвет каждого пикселя будет отнесен к одному из  $K$  кластеров.

**Шаг 2.** Вычислить новые координаты центров всех  $K$  кластеров, которыми считаются средние арифметические значений R, G, B всех пикселов, входящих в кластер.

Цвета всех пикселов изображения заново перераспределить по  $K$  кластерам, вычислив расстояния между ними и найденными центрами кластеров.

**Шаг 3.** Процесс вычисления центров кластеров и перераспределения цветов повторять до тех пор, пока не будет выполнено одно из условий:

1) координаты центров всех кластеров перестают изменяться, т. е. цвета принадлежат тем же кластерам, что и на предыдущей итерации;

2) число итераций достигает максимального заданного числа итераций  $N$ .

Недостатком данного метода является то, что он способен эффективно формировать лишь выпуклые кластеры.

Достоинство – высокая скорость работы. Применимельно к квантованию изображений данный метод показывает очень хорошие результаты.

## 2.3. Кодирование и типы цифровых изображений

В цифровой обработке изображений принято считать, что изображение представляется матрицей целых чисел размером  $N \times M$ , где значение каждого элемента отвечает определенному уровню квантования его энергетической характеристики, или яркости. Это так называемая пиксельная система координат.

В данном случае понятие пиксел (англ. pixel – сокращенно от picture cell – элемент изображения) имеет значение наименьшего логического элемента двухмерного цифрового изображения.

Таким образом, цифровое изображение можно представить в виде матрицы, элементы которой – пиксели – содержат целочисленные значения из некоторой палитры. Напомним, что эти значения безразмерны, поскольку имеют абстрактный характер.

По способу формирования изображения делятся на векторные и растровые. *Векторным* называется изображение, описанное в виде набора графических примитивов, оно рисуется линиями на графических устройствах вывода. *Растровое* изображение представляет собой двухмерный массив, элементы которого содержат информацию о цвете. Оно ориентировано на растровые устройства отображения. В цифровой обработке используются растровые изображения, поэтому векторные здесь не рассматриваем.

Растровые изображения можно разделить на четыре типа: бинарные, полутонаовые, палитровые, полноцветные. Их основная характеристика – глубина цвета. Этот термин означает количество бит, используемых для описания цвета одного пикселя, и в литературе он обозначается как bpp (от англ. bits per pixel).

Выбор типа изображения зависит от решаемой задачи. Основные типы цифровых изображений и их характеристики представлены в табл. 2.1.

Элементы *бинарного* (двухуровневого) изображения описываются одним битом, который принимает значения 0 (черный) и 1 (белый). Эти значения получаются в результате обработки (бинаризации) полутонаовых или цветных изображений. Бинарные изображения удобны при передаче изображений документов в факсимильной связи.

*Полутоновое* изображение состоит из пикселов, которые могут принимать значения от 0 до  $N$ , где  $N=2^k$ . Это один из наиболее распространенных типов изображений. В большинстве случаев используется значение  $k=8$  бит (т. е. один байт) на пикセル.

*Полноцветное* изображение содержит информацию о яркостях цветов (обычно R, G, B), т. е. его можно рассматривать как три полутонаовых изображения.

Таблица 2.1

## Основные типы цифровых изображений

Тип изображения	Основной диапазон значений яркости	Глубина цвета, бит	Количество оттенков
Бинарное	[0, 1]	1	2
Полутоновое	[0, 255]	8	4–256
Палитровое (индексированное)	[0, 255]	8	2–256
Полноцветное	$[0, 255]^3$	24	16 777 216

В палитровых (индексированных) изображениях значение пикселя, т. е. индекс, фактически является ссылкой на ячейку палитры, которая представляет собой таблицу значений RGB для этого индекса. Для описания таких изображений необходимо хранить палитру (таблицу 256 цветов) и, собственно, изображение в виде матрицы индексов (один байт на пикセル). При использовании неравномерных алгоритмов квантования цвета зрительная система человека практически не различает изображения, представленные в полноцветном и индексированном вариантах.

## 2.4. Сжатие изображений

Изображения занимают большой объем памяти, их компактное хранение и передача предполагают сжатие исходных данных. Например, при дистанционном зондировании Земли (ДЗЗ) с помощью спутников сжатие данных представляет серьезную проблему. По спутниковым каналам может передаваться до 2100 Мб/с. Поэтому алгоритмы сжатия прежде всего ориентированы на устранение избыточности двухмерных многоспектральных данных таким образом, чтобы иметь возможность в случае необходимости полностью восстановить исходное изображение.

Методы сжатия изображений можно разбить на две группы [1]:

- сжатие без потерь, позволяющее точно восстанавливать исходное изображение;
- сжатие с потерями (или сжатие с регулируемой степенью искажений), изображение восстанавливается приближенно.

Алгоритмы сжатия без потерь основаны на исключении статистической избыточности. Сжатие без потерь означает, что процесс обратим, т. е. значения яркости (цвета) можно восстановить при декодировании в первоначальном виде. К ним относятся:

- кодирование длин серий (алгоритм RLE);
- метод группового кодирования (алгоритм LZW);
- метод Хаффмана;
- арифметическое кодирование и др.

Методы сжатия с потерями основаны на том, что в изображении содержится информация, слабо воспринимаемая человеческим глазом. Такой вид избыточности называется *психофизиологической избыточностью*. Ее снижение состоит в уменьшении объема несущественной информации, которую человеческий глаз не воспринимает в силу психофизиологических особенностей. При этом часть информации безвозвратно теряется. Восстановить исходные значения яркости или цвета всех пикселов при декодировании невозможно. Например, глаз человека не различает цвет мелких деталей изображения. Следовательно, можно хранить только яркость деталей, а цветность отбросить и таким образом уменьшить объем файла, описывающего изображение. Этот метод применяется в телевидении. Разбиение непрерывного видеопотока на кадры также является примером исключения психофизиологической избыточности.

*Коэффициент сжатия* – основная характеристика алгоритма сжатия. Он определяется как отношение объема исходных несжатых данных к объему сжатых.

#### **2.4.1. Алгоритмы сжатия без потерь**

Одним из самых простых алгоритмов сжатия без потерь является алгоритм RLE (Run Length Encoding – групповое кодирование). Сжатие в RLE происходит за счет того, что в исходном изображении встречаются цепочки одинаковых байт, которые можно экономно кодировать. Различные модификации алгоритма RLE реализованы в утилитах, записывающих изображения в разных форматах (PCX, GIF, TIFF, PMP). Следует отметить, что многие современные форматы поддерживают запись данных (не только изображений) с использованием нескольких алгоритмов сжатия либо без них. Например, формат TIFF 6.0 может сохранять изображения с использованием алгоритмов сжатия RLE-PackBits, RLE-CCIT, LZW, Хаффмана с фиксированной таблицей кодов, JPEG-Lossless или без сжатия. Алгоритм LZW (от фамилий авторов Lempel – Ziv – Welch) является более сложным и универсальным алгоритмом (по отношению к RLE), сжимающим изображения за счет поиска, выделения и кодирования одинаковых подцепочек в потоке байтов. Алгоритм Хаффмана обеспечивает сжатие с помощью предварительного вычисления частоты появления одинаковых байтов в изображении.

Говоря о лучших на сегодня алгоритмах сжатия изображений без потерь, отметим, что в 1994 г. был принят стандарт ISO/JPEG-Lossless. Он ориентирован на полноцветные 24-битовые или 8-битовые полутононевые изображения. На 8-битовых тестовых изображениях нескольких категорий (космических, медицинских, фото- и компьютерно-графических, смешанных) средний коэффициент сжатия алгоритмом JPEG-Lossless равнялся двум.

## 2.4.2. Алгоритмы сжатия с потерями

Опишем один из наиболее распространенных на сегодня алгоритмов сжатия изображений – JPEG. Объединенная группа экспертов в области фотографии (The Joint Photographic Experts Group – JPEG) в 1993 г. в рамках Международной организации по стандартизации (ISO) разработала международный стандарт JPEG для сжатия полноцветных (24-битовых) изображений. Для него разработано четыре варианта сжатия [11]:

- последовательный на базе дискретного косинусного преобразования (ДКП);
- прогрессивный с записью коэффициентов ДКП, использующий пирамидальное представление изображений;
- последовательный без потерь (JPEG-Lossless);
- иерархический с потерями и без потерь соответственно.

Основные шаги алгоритма JPEG приведены на рис. 2.4.



Рис. 2.4. Схема преобразования данных в алгоритме сжатия 24-битового цветного изображения JPEG

Алгоритм оперирует блоками 8x8 пикселов; предполагается, что в них яркость и цвет меняются сравнительно плавно. К матрицам значений таких блоков применяется ДКП, после чего значимыми оказываются только первые коэффициенты. Система цветового восприятия человека слабо распознает некоторые частоты при цветовом изменении картинки, поэтому можно аппроксимировать коэффициенты ДКП без заметной потери качества восстановленного изображения, используя квантование коэффициентов. При таком преобразовании теряется часть информации, но достигаются большие коэффициенты сжатия.

### Алгоритм сжатия с потерями JPEG

Шаг 1. Преобразовать изображение из цветового пространства RGB в пространство YCrCb. Отметим, что стандарт JPEG не регламентирует выбор именно YCbCr, допуская и другие цветовые пространства, однако спецификация JFIF (JPEG File Interchange Format, ставшая в настоящее время стандартом де-факто) предполагает использование преобразования RGB→YCbCr. Перевод из RGB в YCrCb выполняется по формулам

$$\begin{aligned} Y &= 0,299 \cdot R + 0,587 \cdot G + 0,114 \cdot B; \\ Cr &= -0,1687 \cdot R - 0,3313 \cdot G + 0,5 \cdot B + 128; \\ Cb &= 0,5 \cdot R - 0,4187 \cdot G + 0,0813 \cdot B + 128, \end{aligned}$$

где  $Y$  – яркостная составляющая, а  $Cr$ ,  $Cb$  – компоненты, отвечающие за цвет (хроматический красный и хроматический синий).

Поскольку человеческий глаз менее чувствителен к вариациям цветовой компоненты, чем к яркости, можно выполнить прореживание массивов  $Cr$  и  $Cb$ . Параметры прореживания обозначаются отношением  $X:a:b$ , указывающим количество и порядок выбора яркостных  $Y$  и цветоразностных компонент  $Cr$  и  $Cb$ . Значения  $Y$  обычно не прореживаются, поэтому  $X = 4$ . Величина  $a$  – число выборок значений  $Cr$  и  $Cb$  в горизонтальном направлении в первой строке блока,  $b$  – число (дополнительных) выборок цветоразностных компонент ( $Cr$ ,  $Cb$ ) во второй строке (табл. 2.2).

Стандарт допускает и другие варианты прореживания.

Таблица 2.2

Варианты дискретизации цвета

Формула прореживания цвета	$Y$	$Cb$	$Cr$
4:4:4	Все	Все	Все
4:2:2	Все	Через один в строке	Через один в строке
4:2:0 (для MPEG-2)	Все	Через один в строке, среднее между соседними столбцами	Через один в строке, среднее между сосед- ними столбцами
4:1:1	Все	Через три в строке	Через три в строке

**Шаг 2.** Разбить исходное изображение на блоки  $8 \times 8$  элементов. Сформировать три рабочие матрицы ДКП для каждой компоненты  $Y$ ,  $Cr$ ,  $Cb$ .

Почему изображение разбивается на блоки  $8 \times 8$ ? В цифровой обработке данных часто используется степень двойки. Экспериментально установлено, что разбиение  $8 \times 8$  дает лучший результат сжатия, чем разбиение  $4 \times 4$ . Применение большего размера блока незначительно улучшает качество восстановленного изображения (при равном количестве сохраняемых далее коэффициентов), но увеличивает число вычислений. Например, вычисление ДКП при разбиении  $64 \times 64$  в два раза сложнее, чем для  $8 \times 8$ . Таким образом,  $8 \times 8$  – это экспериментально выбранный компромисс по критерию качество-сложность.

**Шаг 3.** Применить ДКП, вычисляя значения косинусного преобразования для каждого блока  $8 \times 8$  и подставляя вместо  $f(x, y)$  значения  $Y(x, y)$ ,  $Cr(x, y)$ ,  $Cb(x, y)$  по формуле

$$F(u, v) = \frac{1}{4} C(u) C(v) \sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cos\left[\frac{\pi(2x+1)u}{16}\right] \cos\left[\frac{\pi(2y+1)v}{16}\right]$$

для  $u = 0, \dots, 7$  и  $v = 0, \dots, 7$ ,

$$\text{где } C(k) = \begin{cases} 1/\sqrt{2} & \text{для } k = 0, \\ 1 & \text{в противном случае.} \end{cases}$$

В результате получаем матрицы  $Y[u, v]$ ,  $Cr[u, v]$ ,  $Cb[u, v]$ , в которых коэффициенты в левом верхнем углу соответствуют низкочастотной составляющей изображения, а в правом нижнем – высокочастотной, причем  $Y[0, 0]$  равно среднему значению в блоке 8x8.

На данном этапе сжатия не происходит, выполняется только преобразование данных.

#### Шаг 4. Квантование значений ДКП.

Выполнить квантование коэффициентов делением рабочей матрицы на матрицу квантования поэлементно. Для каждой компоненты ( $Y$ ,  $Cr$  и  $Cb$ ) задается своя матрица квантования (МК)  $Q[u, v]$  размером 8x8. Значения  $Q[u, v]$  определяют качество сжатия. Например, квантование для  $Y$  выполняется поэлементным делением одной матрицы на другую с округлением полученных значений до целых:

$$Y_Q[u, v] = \text{IntegerRound}\left(\frac{Y[u, v]}{Q[u, v]}\right).$$

Значения матрицы  $Q$  различны при разных коэффициентах качества (Quality Factor), которые могут задаваться пользователем. В разных приложениях этот коэффициент имеет разные диапазоны значений: от 1 до 100 (стандарт JPEG) или от 1 до 12 (пакет Adobe Photoshop). От него зависит качество изображения, восстановленного после сжатия. Отметим, что коэффициенты качества не определяют процент потери качества, это лишь относительные величины, т. е. если коэффициент качества равен 100, это не означает отсутствие потери данных. Ниже представлены рекомендованные стандартом матрицы квантования для коэффициента качества, равного 50:  $Q_l$  для  $Y$  и  $Q_c$  для компонент  $Cr$ ,  $Cb$ :

$$Q_l = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix};$$

$$Q_c = \begin{bmatrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{bmatrix}.$$

Производители разных цифровых фотоаппаратов используют разные матрицы квантования  $Q$  для сжатия изображений алгоритмом JPEG (см. <http://www.impulseadventure.com/photo/jpeg-quantization.html>). Описания двух матриц квантования, использованных при сжатии, хранятся в файлах с изображениями этих матриц.

Приведем пример такого квантования. Для блока значений (слева) коэффициенты ДКП имеют вид, представленный матрицей (справа).

139 144 149 153 155 155 155 155	235,6 -1,0 -12,1 -5,2 2,1 -1,7 -2,7 1,3
144 151 153 156 159 156 156 156	-22,6 -17,5 -6,2 -3,2 -2,9 -0,1 0,4 -1,2
150 155 160 163 158 156 156 156	-10,9 -9,3 -1,6 1,5 0,2 -0,9 -0,6 -0,1
159 161 162 160 160 159 159 159	-7,1 -1,9 0,2 1,5 0,9 -0,1 0,0 0,3
159 160 161 162 162 155 155 155	-0,6 -0,8 1,5 1,6 -0,1 -0,7 0,6 1,3
161 161 161 161 160 157 157 157	1,8 -0,2 1,6 -0,3 -0,8 1,5 1,0 -1,0
162 162 161 163 162 157 157 157	-1,3 -0,4 -0,3 -1,5 -0,5 1,7 1,1 -0,8
162 162 161 161 163 158 158 158	-2,6 1,6 -3,8 -1,8 1,9 1,2 -0,6 -0,4

Применив к ним квантование, с помощью матрицы  $Q_l$  получим

$$\begin{matrix} 15 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

При декодировании изображения восстанавливают коэффициенты ДКП умножением на матрицу квантования  $Q_l$  (слева) и применением обратного ДКП (справа).

240 0 -10 0 0 0 0 0	144 146 149 152 154 156 156 156
-24 -12 0 0 0 0 0 0	148 150 152 154 156 156 156 156
-14 -13 0 0 0 0 0 0	155 156 157 158 158 157 156 155
0 0 0 0 0 0 0 0	160 161 161 162 161 159 157 155
0 0 0 0 0 0 0 0	163 163 164 163 162 160 158 156
0 0 0 0 0 0 0 0	163 164 164 164 162 160 158 157
0 0 0 0 0 0 0 0	160 161 162 162 161 159 158 158
0 0 0 0 0 0 0 0	158 159 161 161 162 161 159 158

Заметны небольшие отличия в исходных и восстановленных значениях яркости.

На шаге 4 происходят *самые большие потери информации*. Выбирая матрицу квантования с большими коэффициентами, получаем больше нулей после квантования и, следовательно, большую степень сжатия.

Такое квантование порождает специфические эффекты. При большом сжатии потери низкочастотных параметров могут быть настолько велики, что восстановленное изображение выглядит разбитым на квадраты. При максимальном сжатии после восстановления все пиксели в блоках 8x8 имеют одинаковый цвет. Потери в высоких частотах могут проявиться в виде так называемого эффекта Гиббса, когда вокруг контуров с резким переходом цвета образуется своеобразный «нимб».

**Шаг 5.** Преобразовать матрицу квантованных коэффициентов 8x8 в 64-элементный вектор при помощи «зигзаг»-сканирования путем упорядочивания элементов с индексами [0,0], [0,1], [1,0], [2,0], [1,1], [0,2], [0,3], [1,2], ... [7,7] (рис. 2.5). В результате в начале вектора будут записаны коэффициенты, соответствующие низким частотам, а в конце – высоким. Последние значения вектора почти всегда равны нулю.

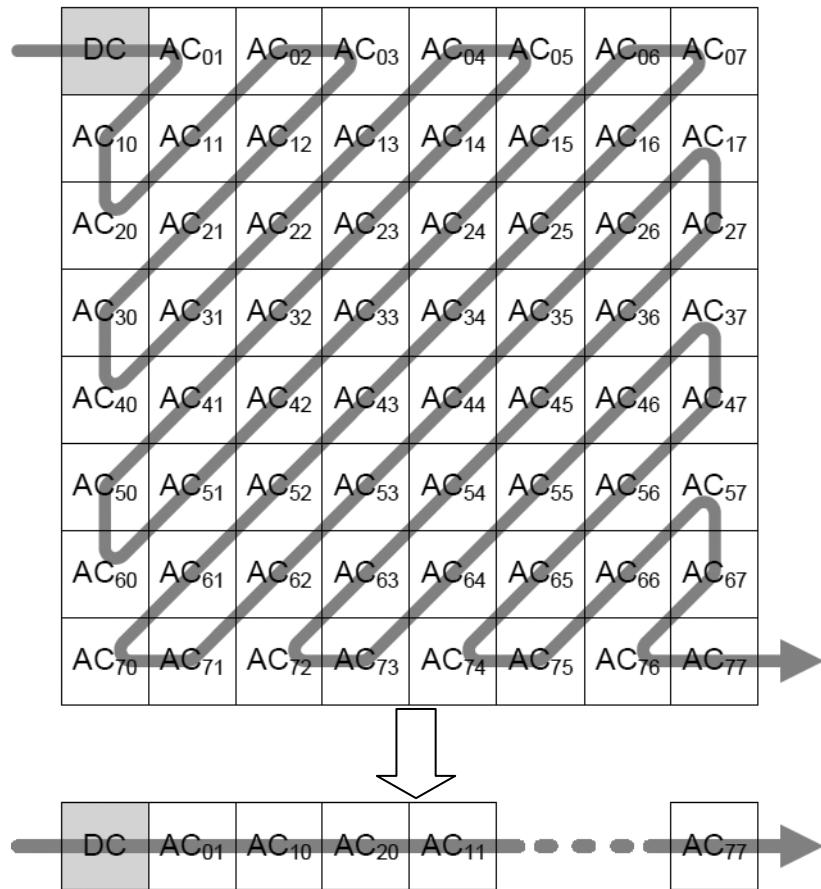


Рис. 2.5. Схема зигзагообразного преобразования матрицы в вектор

Первое значение вектора всегда равно средней яркости блока 8x8 и называется коэффициентом DC, а остальные – коэффициентами AC.

Шаг 6. Выполнить первичное сжатие полученного вектора с помощью алгоритма группового кодирования (RLE). При этом получаются пары чисел типа («пропустить», «число»), где параметр «пропустить» является счетчиком пропускаемых нулей, а «число» – значением, которое необходимо поставить в следующую ячейку. Так, вектор

42 3 0 0 0 -2 0 0 0 0 1 ...

будет свернут в пары (0,42) (0,3) (3,-2) (4,1).

Шаг 7. Выполнить сжатие полученных значений по алгоритму Хаффмана с фиксированной кодовой таблицей (табл. 2.3).

Идея этого алгоритма заключается в кодировании часто употребляемых чисел меньшим числом битов, а редко встречающихся – большим количеством. Вместо хранения фактических значений целочисленных параметров определяется минимальное число битов, описывающее эти значения, затем бинарными значениями кодируется представление этих величин.

Таблица 2.3

Таблица Хаффмана для кодирования коэффициентов в алгоритме JPEG

Диапазоны коэффициентов	Кол-во битов (категория)	Значения кодов
[-1, 1]	1	0, 1
[-3, -2], [2, 3]	2	00, 01, 10, 11
[-7, -4], [4, 7]	3	000, 001, 010, ..., 111
[-15, -8], [8, 15]	4	0000, 0001, ..., 1111
[-31, -16], [16, 31]	5	00000, ..., 11111
[-63, -32], [32, 63]	6	000000, ..., 111111
[-127, -64], [64, 127]	7	0000000, ..., 1111111
[-255, -128], [128, 255]	8	00000000, ..., 11111111
[-511, -256], [256, 511]	9	000000000, ..., 111111111
[-1023, -512], [512, 1023]	10	0000000000, ..., 1111111111
...	...	...
[-32 767, -16 384], [16 384, 32 767]	15	000000000000000, ..., 1111111111111

Рассмотрим на примере [26] задачу кодирования вектора (0,57); (0,45); (4,23); (1,-30); (0,-8); (2,1); (0,0) по алгоритму Хаффмана. Сначала опишем, как кодируются правые величины этих пар, кроме тех, которые являются специальными маркерами подобно (0,0) или (15,0).

Число 57 попадает в категорию 6 и кодируется шестью битами. Битовый код этого числа равен 111001, поэтому оно кодируется как (6,111001), а число 45 – как (6,101101). В результате вместо набора чисел, записанных в левом столбце, получаем набор кодов, записанных справа:

Кодируемое число	Категория, код
57	(6,111001)
45	(6,101101)
23	(5,10111)
-30	(5,00001)
-8	(4,0111)
1	(1,1)

Затем левые значения, указанные в скобках, будут заменены аналогичным парным кодом:

$$((0,6), 111001); ((0,6), 101101); ((4,5), 10111); \\ ((1,5), 00001); ((0,4), 0111); ((2,1), 1); (0,0).$$

Пары двух величин, заключенные во внутренние скобки, могут быть описаны одним байтом, так как каждое из двух значений может быть представлено четырьмя битами (счетчик предшествующих нулей и параметр категории всегда меньше 15). В этом байте старшая часть определяет число предшествующих нулей, а младшая – категорию новой величины, отличной от нуля. Такая пара рассматривается как единое число, которое кодируется в соответствии с таблицей Хаффмана (табл. 2.3):

Кодируемая пара значений	Пара как число	Код Хаффмана
(0,6)	6	111000
(4,5)	69	1111111110011001
(1,5)	21	11111110110
(0,4)	4	1011
(2,1)	33	11011
(0,0)	0 = EOB (конец блока)	1010

В результате кодируемый вектор записывается в виде набора битов:

$$111000 \ 111001 \ 111000 \ 101101 \ 111111110011001 \ 10111 \\ 11111110110 \ 00001 \ 1011 \ 0111 \ 11011 \ 1 \ 1010.$$

Отметим, что коэффициент DC соответствует самой низкой частоте в обрабатываем блоке 8x8 пикселов, он равен среднему арифметическому всех 64 значений блока. Обычно DC имеют большие значения. Согласно алгоритму в начале записывается абстрактное значение  $DC_0 = 0$ , а вместо значений последующих коэффициентов DC записываются величины  $Diff_i$  (разницы относительно предыдущего значения DC), вычисляемые по формуле  $Diff_i = DC_i - DC_{i-1}$ .

При сохранении изображения в файле указывается параметр качества  $P$ , задаваемый в условных единицах, например от 1 до 100. Большее число обычно соответствует лучшему качеству и большему размеру

файла. Однако даже при использовании наивысшего качества (соответствующего матрице квантования, состоящей из одних только единиц) восстановленное изображение может не совпадать с исходным, что связано с точностью вычисления ДКП, а также из-за округления значений Y, C<sub>b</sub>, C<sub>r</sub> и коэффициентов ДКП до целых значений.

Декодирование изображения из сжатого представления выполняется в обратном порядке.

Хотя алгоритм JPEG и является стандартом ISO, формат его файлов не был зафиксирован. Поэтому многие производители заменяют внутренние таблицы, рекомендованные стандартом, на свои собственные и создают свои форматы JPEG, не всегда совместимые между собой.

Алгоритм JPEG позволяет сжимать некоторые изображения в 10–15 раз без визуально заметных искажений. Он ориентирован на сжатие фотографий, содержащих сцены с плавными переходами яркости и цвета. Наибольшее распространение JPEG получил в сети Интернет. Он мало пригоден для сжатия чертежей и текста, имеющих резкий контраст между соседними пикселями, что приводит к появлению заметных артефактов. Подобные изображения лучше сохранять в форматах без потерь, таких как TIFF, GIF или PNG.

JPEG не должен применяться тогда, когда недопустимы даже минимальные потери, например при сжатии медицинских изображений. В таких случаях можно использовать предусмотренный стандартом JPEG вариант сжатия JPEG-Lossless. Он не использует ДКП, что обеспечивает точное совпадение восстановленного и исходного изображений, но его малая эффективность (коэффициент сжатия редко бывает более двух) и отсутствие поддержки со стороны разработчиков программного обеспечения не способствовали популярности JPEG-Lossless.

Стандарт JPEG предусматривает два основных варианта записи кодируемых данных в файл.

Наиболее распространенным, поддерживаемым большинством доступных кодеков, является последовательное представление данных, получаемое в результате последовательного обхода кодируемого изображения поблочно слева направо и сверху вниз. Над каждым кодируемым блоком изображения осуществляются описанные выше операции, а результаты кодирования помещаются в файл последовательно блок за блоком. Изображение восстанавливается последовательно блок за блоком, слева направо и сверху вниз (рис. 2.6, справа).

Второй вариант (прогрессивный JPEG) заключается в том, что сначала в файл записываются низкочастотные коэффициенты всех блоков, затем все A<sub>01</sub>, все A<sub>10</sub> и т. д. Прогрессивное представление данных позволяет увидеть грубый вариант всего изображения уже после декодирования части JPEG-файла (рис. 2.6, слева).



Рис. 2.6. Изображения, восстанавливаемые после сжатия прогрессивным (слева) и последовательным (справа) алгоритмами JPEG

В алгоритме JPEG 2000 вместо ДКП (применяемого в алгоритме JPEG) используется вейвлет-преобразование, которое основано на представлении сигнала в виде суперпозиции базовых функций со специальными свойствами, называемых *вейвлетами*. Сжатое этим алгоритмом, а затем восстановленное изображение получается более гладким и четким, а размер файла по сравнению с JPEG при одинаковом качестве восстановленного изображения оказывается меньшим. JPEG 2000 полностью свободен от главного недостатка JPEG: благодаря использованию вейвлетов, изображения, восстановленные после сильного сжатия, не содержат артефактов в виде блоков пикселов.

Алгоритм JPEG 2000, как и JPEG, поддерживает так называемое прогрессивное сжатие, позволяющее по мере загрузки увидеть сначала размытое, а затем все более четкое изображение.

#### *Основные преимущества JPEG 2000 по сравнению с JPEG*

1. Степень сжатия в среднем на 20 % больше (рис. 2.7). Большая степень сжатия достигается благодаря использованию дискретного вейвлет-преобразования к изображению целиком и более сложного энтропийного кодирования данных.

2. Масштабируемость фрагментов изображений позволяет бесшовное сжатие разных компонент изображения. Благодаря разбиению на блоки можно хранить изображения разных разрешений в одном кодовом потоке.

3. Произвольный доступ к областям интереса обеспечивает несколько механизмов доступа, поддерживает несколько степеней разбиения на части (области интереса).

4. Гибкий формат файла допускает хранение информации о цветовых пространствах, метаданных и информации для согласованного доступа в сетевых приложениях.

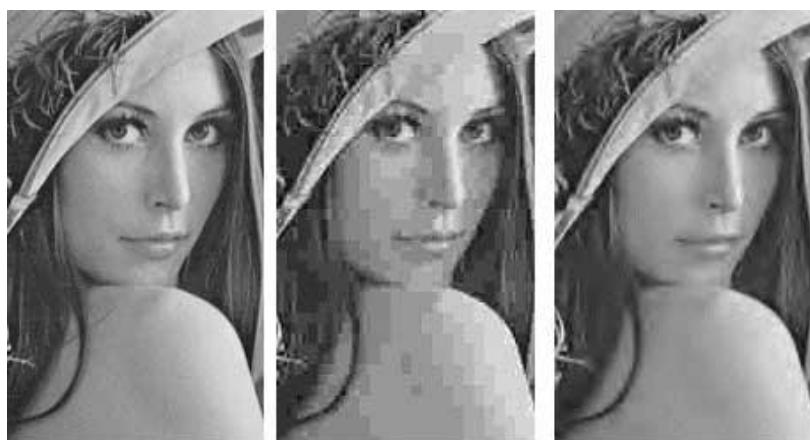


Рис. 2.7. Исходное изображение (слева), восстановленное после сжатия в 64 раза алгоритмами JPEG (по центру) и JPEG 2000 (справа)

Артефакты, возникающие при сжатии по алгоритму JPEG 2000 с высокой степенью компрессии, качественно мало отличаются от артефактов, возникающих при сжатии алгоритмом JPEG. В тех местах, где оригинальное изображение имело плавные цветовые переходы, они становятся еще более плавными (размытыми), а где были резкие переходы яркости или цвета (участки изображения с высокой контрастностью), возникает характерный артефакт в виде яркого контура, обрамляющего границу перехода, незначительно (на пару пикселов) отступающего от более темного участка перехода. Различия в артефактах: нет сетки в 8x8 пикселов; не искажаются цвета мелких деталей, сильно отличающихся по цвету от фона; артефакты, характерные для JPEG 2000, становятся заметными при больших, чем в случае JPEG, степенях сжатия.

Алгоритм *DjVu* (от франц. *déjà vu* – уже виденное) реализует сжатие изображений с потерями. Он разработан специально для сжатия отсканированных документов (книг, журналов, рукописей и т. п.), где обилие формул, схем, рисунков и рукописных символов делает трудоемким их полноценное распознавание. Также данный алгоритм является эф-

фективным, если необходимо передать: все нюансы оформления, например исторических документов, где важное значение имеют содержание, цвет и фактура бумаги; дефекты пергамента (трещинки, следы от складывания); исправления, кляксы, отпечатки пальцев, следы, оставленные другими предметами и т. д.

Для сжатия цветных изображений в DjVu применяется технология, разделяющая исходное изображение на три слоя: передний план, задний план и черно-белую (однобитовую) маску. Маска сохраняется с разрешением исходного файла и именно она содержит изображение текста и прочие четкие детали. Разрешение заднего плана, в котором остаются иллюстрации и текстура страницы, по умолчанию понижается для экономии места. Передний план содержит цветовую информацию о маске. Его разрешение обычно понижается еще сильнее. Затем задний и передний планы сжимаются с помощью вейвлет-преобразования, а маска – алгоритмом JB2.

Особенностью алгоритма JB2 является то, что он ищет на странице повторяющиеся символы и сохраняет их изображение только один раз. В многостраничных документах каждые несколько подряд идущих страниц пользуются общим «словарем» изображений символов.

Для сжатия большинства книг можно обойтись только двумя цветами. В этом случае используется всего один слой, что позволяет достичь рекордной степени сжатия. В типичной книге с черно-белыми иллюстрациями, отсканированной с разрешением 600 dpi, средний размер страницы составляет около 15 КБ, т. е. приблизительно в 100 раз меньше, чем объем исходного изображения. При наличии сложного заднего плана выигрыш объема составляет обычно 4–10 раз. Однако при стандартных настройках в DjVu используется сжатие данных с потерями, поэтому для особо важных документов применяются алгоритмы сжатия без потерь: PNG, JPEG 2000, TIFF и т. п. В DjVu можно использовать и сжатие данных без потерь с помощью утилиты CJV2.

В основе алгоритма DjVu лежат несколько базовых алгоритмов:

- алгоритм отделения текста от фона на отсканированном изображении;
- алгоритм вейвлет-сжатия фона IW44;
- алгоритм сжатия черно-белых изображений JB2;
- универсальный алгоритм сжатия ZIP;
- алгоритм распаковки «по запросу»;
- алгоритм «маскирования» изображений.

*Фрактальное сжатие изображений.* Некоторые недостатки в качестве восстановленного изображения после сжатия алгоритмом JPEG связаны с тем, что при сжатии не учитывается специфика изображенных объектов, т. е. не выявляются их структура и характерные участки. Учет структуры изображенного объекта лежит в основе фрактального метода

сжатия изображений, который предложил в 1988 г. Б. Мандельброт. Фрактал – это структура, выделенная при анализе изображения и обладающая схожей формой независимо от ее размеров. Например, в изображении кроны дерева, фрактал может представлять лист. Поэтому изображение можно формировать, собирая из объектов, называемых фракталами. В свою очередь, отдельный фрактал может быть описан некоторым стандартным образом [16]. Основу фрактального подхода составляет постулат о том, что существует набор аффинных коэффициентов, описывающих вращение, сжатие, расширение, искажение формы и сдвиг объектов, составляющих изображение.



Рис. 2.8. Лист папоротника (слева) и сгенерированное с помощью фракталов изображение (справа)

Наиболее распространенным примером изображения, построенного с помощью фракталов, является изображение папоротника (рис. 2.8). Оно состоит из четырех аффинных преобразований. Каждое преобразование кодируется считанными байтами, хотя исходное изображение может быть любого размера. Таким образом, фрактальное сжатие – это в первую очередь результат анализа изображения с целью поиска самоподобных объектов с их последующим кодированием.

## 2.5. Форматы представления цифровых изображений

Изображения требуется эффективно хранить и экспорттировать. Для этого созданы разные описания для хранения изображений в компьютерных файлах – форматы. Существует множество различных форматов графических файлов, т. е файлов, содержащих изображения. Многие из них стандартизированы.

*Формат* – это спецификация структуры данных, записанных в компьютерном файле. Тип формата часто указывается в расширении имени файла, отделенном точкой. Например, окончание имени «.tif» используют для обозначения файлов, содержащих изображение, описанное в формате TIFF.

Для многих форматов существуют опубликованные спецификации, где подробно описана структура файлов данного формата и то, как программы должны кодировать данные для записи в этот формат и декодировать их при чтении [7, 9].

Наиболее распространенные форматы и их особенности представлены в табл. 2.4.

Таблица 2.4

Форматы графических файлов, их описание и область применения

Формат	Особенности	Применение
RAW	Матрица цифровых данных, полученных с ПЗС-матрицы фотокамеры	Сжатие без потери качества
BMP	Хранит данные с глубиной цвета до 48 бит и максимальным размером 65 535×65 535 пикселов без сжатия	Поддерживается ОС Windows
GIF	Использует 256 оттенков цвета, эффективно сжимает однотонные цветные области, сохраняя детали изображения без потерь. Может содержать прозрачные области	Текст, логотипы, векторные иллюстрации, анимированные рисунки
TIF	Сохраняет фотографии в цветовых пространствах (RGB, CMYK, YCbCr, CIE Lab и др.) с глубиной цвета 8, 16, 32 и 64 бит	Полиграфия; GeoTIFF используется для хранения в одном файле нескольких спектральных спутниковых изображений
JPEG	Поддерживает до 16 млн цветов. Сжимает данные с потерями, может внести искажения в рисунок, особенно содержащий текст или тонкие линии	Фотографии. Не годится для рисунков, содержащих прозрачные участки, мелкие детали или текст
PNG-24	Поддерживает 16 млн цветов. Подобно JPEG сохраняет яркость и оттенки цветов в фотографиях, подобно GIF и PNG-8 сохраняет детали изображения (например, в векторных рисунках и логотипах)	Фотографии; рисунки, содержащие прозрачные участки; рисунки с большим количеством цветов и четкими краями изображений

Формат **TIFF (Tagged Image File Format)** изначально был создан в качестве универсального формата для хранения сканированных изображений с цветовыми каналами. Его главное достоинство – переносимость на разные платформы. На сегодняшний день TIFF является одним из самых распространенных и надежных, его поддерживают практически все программы, так или иначе связанные с графикой. В этом формате доступен весь диапазон цветовых моделей – от монохромной до RGB, CMYK и дополнительных цветов Pantone. Кроме того, TIFF позволяет в случае необходимости сохранять разнообразную дополнительную информацию об изображении (контуры, альфа-каналы и т. п.). В формате TIFF может быть использовано сжатие по алгоритму LZW.

TIFF – это лучший выбор при хранении изображений с высоким качеством, предназначенных для дальнейшего анализа и обработки, импорта растровой графики в векторные программы и издательские системы.

Независящий от аппаратного обеспечения формат **GIF (Graphics Interchange Format)** был разработан в 1987 г. (модификация GIF87a) фирмой CompuServe для передачи растровых изображений по сетям. В 1989 г. спецификация формата была доработана (модификация GIF89a), добавлены поддержка прозрачности и анимации изображений. Формат GIF использует LZW-сжатие и имеет ограниченную палитру в 256 цветов. Его можно назвать форматом хранения изображений без потерь. Однако это утверждение справедливо только в том случае, если исходное изображение имеет от 2 до 256 цветов. Если же количество цветов более 256, то часть из них будет изменена.

GIF используется как один из основных растровых форматов для представления графических данных в Интернете.

Формат **JPEG** описывает структуру данных в файле после сжатия изображений одноименным алгоритмом JPEG и предназначен для хранения полутоновых и цветных фотографий с высоким коэффициентом сжатия без заметной для глаза человека потери качества. При этом, как правило, изображение не может быть полностью восстановлено в исходном виде. При сохранении изображения в формате JPEG можно управлять параметром качества, измеряемым в относительных единицах, чаще всего от 0 (максимальное сжатие) до 100 (максимальное качество).

Формат JPEG следует использовать для хранения полноцветных изображений (в первую очередь фотографий), не предназначенных для дальнейшей обработки. Не стоит его применять для хранения синтезированных изображений с небольшим количеством цветов (менее 256) и с резкими переходами между цветами, так как это может привести к проявлению различимых глазом дефектов.

Формат **BMP** (**Windows Device Independent Bitmap**, от BitMaP – битовая карта) является растровым форматом ОС Windows. Он позволяет описывать структуру изображения как в индексированном виде (с палитрой до 256 цветов), так и полноцветные RGB-изображения (с глубиной цвета 24 бита). Возможно применение алгоритма сжатия RLE. Использование данного формата ограничено тем, что он не поддерживается графическими программами под управлением большинства операционных систем, кроме Windows.

Формат **DjVu** оптимизирован для хранения и передачи документов по сети таким образом, что страницу можно просматривать еще до завершения загрузки файла. DjVu-файл может содержать текстовый слой (это позволяет осуществлять полнотекстовый поиск по файлу), а также встроенное интерактивное оглавление и активные области – ссылки (это позволяет реализовать удобную навигацию в DjVu-книгах).

## 2.6. Видео и форматы видео

Цифровое видео базируется на технологиях записи, обработки, передачи, хранения, воспроизведения визуальных и аудиовизуальных данных в цифровом представлении. Не стоит путать понятие «цифровое видео» с цифровым телевидением. Цифровое телевидение определяет стандарты передачи видео- и аудиосигнала от передатчика к телеприемнику, используя при этом цифровую модуляцию, т. е. предполагает передачу цифрового видео на расстояние посредством спутниковых, наземных, мобильных или кабельных сетей.

Видеосигнал формируется из последовательности кадров – отдельных изображений. Системы регистрации видеосигнала бывают однократные и трехматричные. Для получения цветного видеозображения применяют специальную RGB-фильтрацию элементов изображения, чтобы на каждый элемент приходилось по три значения цвета – красного, зеленого и синего. Такой метод применяется в однократичных видеокамерах и фотоаппаратах с поддержкой видеозаписи.

Для профессиональной видеосъемки применяются трехматричные системы, где оптическое цветоделение производится при помощи дихроидной призмы. Каждое из цветоделенных изображений попадает на отдельную матрицу, с которойчитываются элементы кадра, затем формируется видеосигнал отдельной компоненты. После гамма-коррекции сигналов R, G, B производится их преобразование в модель YCbCr. Полученные компоненты Y, Cr, Cb квантуются с разрядностью 8 или 10 бит. Но для передачи компоненты Y при 8-битном кодировании из 256 доступных уровней используются только 220 (от 16-го до 235-го), а остальные – для сигналов синхронизации. При 10-битном кодировании

используется 877 уровней; для цветовых компонент Cr, Cb – 225 уровней в 8-битной системе и 897 уровней в 10-битной.

Для цифровых видеостандартов принято базовое соотношение 4:2:2, которое означает, что цветоразностные компоненты Cr, Cb передаются с пространственным разрешением в два раза меньшим, чем по яркостному сигналу Y, потому что человеческий глаз более чувствителен к изменению яркости, чем цвета. При этом частота дискретизации для яркостного сигнала Y устанавливается равной 13,5 МГц, что в два раза больше, чем для цветоразностных сигналов Cr и Cb – 6,75 МГц. Для сигналов телевидения высокой четкости установлены частоты дискретизации сигналов яркости (74,25 МГц) и цветности (37,125 МГц). Для европейского видеостандарта PAL размер кадра составляет 720x576 пикселов, для североамериканского стандарта NTSC – 720x480, для видео высокой четкости (HD 720 p) – 1280x720, а для стандарта HDTV (Full HD) – 1920x1080 точек. Минимальный показатель, при котором движение будет восприниматься плавным, – 16 кадров в секунду. В кино стандартная частота съемки и проекции составляет 24 кадра в секунду. Системы телевидения PAL и SECAM используют 25 кадров в секунду, а система NTSC – 30 кадров в секунду.

Стандарт разложения определяет параметры телевизионной развертки, применяемой для преобразования двухмерного изображения в одномерный поток данных. Развертка может быть прогрессивной (построчной) или чересстрочной. При прогрессивной развертке все горизонтальные строки изображения отображаются поочередно одна за другой, при чересстрочной – каждый кадр разбивается на два полукадра, каждый из которых содержит четные или нечетные строки.

Для размещения 60-секундного ролика с частотой 30 кадров в секунду, разрешением 720x576 пикселов и 16-битной глубиной цвета потребуется около полутора гигабайт без учета звуковой дорожки. Существует несколько десятков алгоритмов сжатия, которые используют разные варианты компрессии. Для кодирования видеосигнала используют форматы видео MPEG-2, MPEG-4, H.264, DV, AVC и др. При этом, применяя определенный алгоритм преобразования данных, можно сжать видео разными программными средствами, что дает на выходе различные результаты. Эти отличия определяются кодеком – программой, осуществляющей алгоритм сжатия исходных сигналов. Каждый кодек использует собственный алгоритм, который влияет на качество и скорость кодирования. Фильм длительностью 1,5 ч, записанный в формате DV, занимает 20 Гб, в MPEG-2 – 4,7 Гб, в MPEG-4 – 700 Мб.

## Глава 3. Регистрация цифровых изображений

Цифровые изображения могут быть получены посредством регистрации не только привычных глазу картин в видимом диапазоне электромагнитного спектра, но и посредством регистрации невидимых сигналов. Рассмотрим основные типы устройств регистрации и способы превращения сигналов в цифровые изображения.

### 3.1. Пиксел как регистрирующий сенсор

Принцип создания изображения одинаков и в цифровой камере, и в пленочной: обе обеспечивают фиксацию объекта съемки, используя энергию света, которая воздействует на светочувствительный материал. Элементарная частица электромагнитного излучения называется фотоном. Рассмотрим, как фотон становится электроном [8].

Устройства регистрации изображений в оптическом диапазоне спектра состоят из множества светочувствительных элементов, называемых пикселями. Пиксел состоит из р-подложки, покрытой прозрачным диэлектриком, на который нанесен светопропускающий электрод, формирующий потенциальную яму (рис. 3.1).

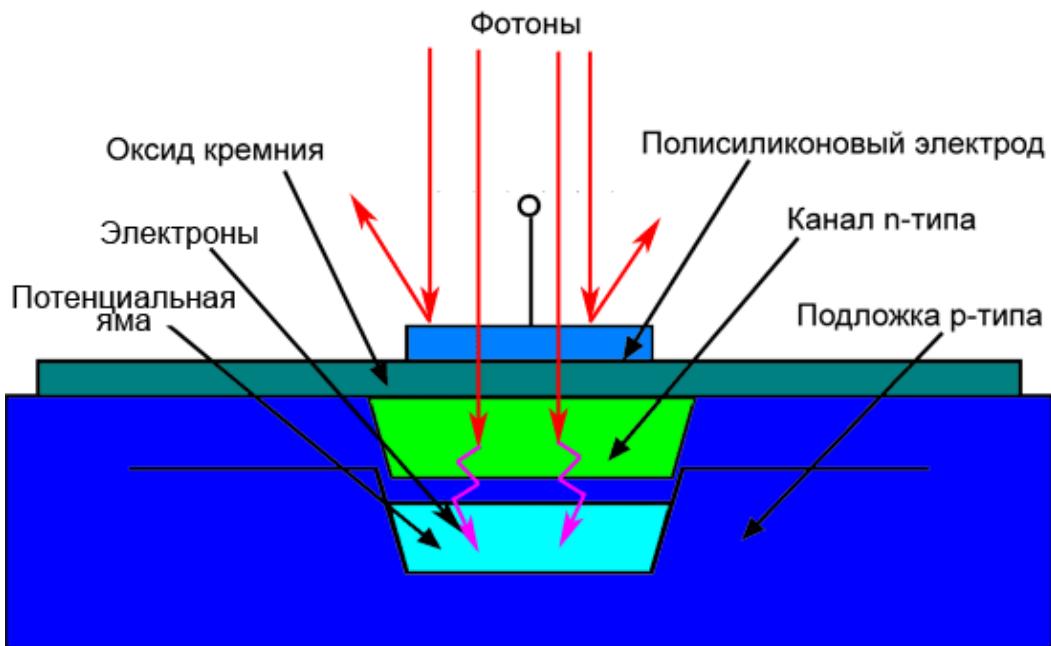


Рис. 3.1. Сечение пикселя ПЗС-матрицы

Для фотона, попавшего на пиксель, возможны три варианта развития событий: он срикошетит от поверхности, будет поглощен в толще полупроводника, пробьет насеквоздь рабочую зону полупроводника. Разработ-

чики стремятся создавать сенсоры, в которых потери от рикошета и пробивания были бы минимальны.

В приборах с зарядовой связью преобразование фотона в электрон происходит в результате поглощения светового кванта кристаллической решеткой полупроводника с выделением электронов. Это может быть либо пара «электрон + дырка», либо единичный электрон – последнее происходит при использовании примесей в полупроводнике. Образовавшиеся носители заряда до момента считывания необходимо сохранить. Для этого кремниевая подложка р-типа оснащается каналами из полупроводника п-типа, над которыми из поликристаллического кремния изготавливаются прозрачные для фотонов электроды. После подачи на такой электрод электрического потенциала в обедненной зоне под каналом п-типа создается потенциальная яма, назначение которой – хранить заряд, возникающий в результате внутреннего фотоэффекта. Чем больше фотонов упадет на светочувствительный элемент (пиксель) и превратится в электроны, тем выше будет заряд, накопленный ямой. Данный заряд получил название *фототок*, его значение довольно мало и после считывания требует обязательного усиления.

Не вся поверхность пикселя является светочувствительной, поэтому над ним часто устанавливается микролинза для увеличения количества накапливаемых фотонов (рис. 3.2). Размер одного пикселя равен 3–8 мкм.

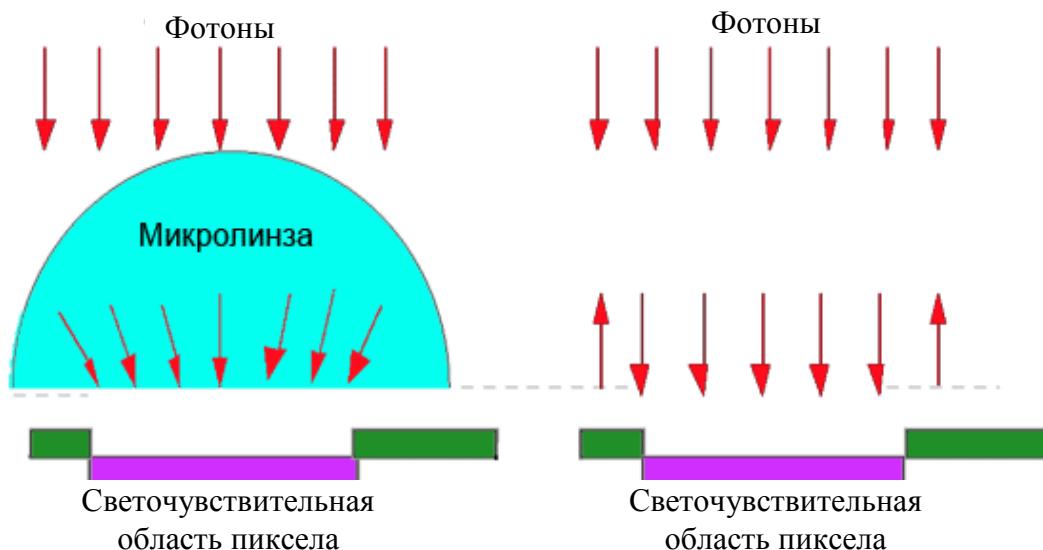


Рис. 3.2. Микролинза повышает светочувствительность пикселя

Считывание фототоков светочувствительных элементов осуществляется так называемыми *последовательными регистрами сдвига*, которые преобразуют строку зарядов на входе в серию импульсов на выходе. Данная серия представляет собой аналоговый сигнал.

В технике используются устройства разного типа, в которых реализованы два способа сбора фототоков с индивидуальных сенсоров:

ПЗС – приборы с зарядовой связью, за разработку которых в 2009 г. У. Бойл и Дж. Смит получили Нобелевскую премию;

КМОП – приборы на базе симметричных комплементарных металлооксидных полупроводников.

Преимуществами ПЗС-устройств являются:

- низкий уровень шумов;
- высокий коэффициент заполнения пикселов (до 100 %);
- высокая эффективность (отношение числа зарегистрированных фотонов к числу попавших на светочувствительную область сенсора) – 95 %, а человеческий глаз имеет квантовую эффективность порядка 1 %;
- высокий динамический диапазон (отношение тока насыщения к среднему квадратичному шумовому току, дБ);
- хорошая чувствительность в инфракрасном диапазоне спектра.

К недостаткам ПЗС-устройств относятся:

- сложная система считывания сигнала;
- высокий уровень энергопотребления (до 2–5 Вт);
- дороже КМОП в производстве.

Преимущества КМОП-устройств:

- высокое быстродействие (регистрация до 500 кадров в секунду);
- низкое энергопотребление (в 100 раз по сравнению с ПЗС);
- дешевле и проще ПЗС в производстве;
- перспективность технологии (на одном кристалле можно реализовать все необходимые дополнительные схемы: аналого-цифровые преобразователи, процессор, память, т. е. фактически законченную цифровую камеру на одном кристалле).

Недостатки КМОП-устройств:

- низкий коэффициент заполнения пикселов, что снижает чувствительность (эффективная поверхность пикселя до 75 %, остальное занимают транзисторы);
- высокий уровень шума (даже при отсутствии освещения через фотодиод течет значительный ток), борьба с которым усложняет технологию;
- невысокий динамический диапазон.

В ПЗС-сенсорах сигнал считывается последовательно из каждой ячейки, ряд за рядом. Поэтому сделать следующий снимок можно лишь тогда, когда предыдущий уже полностью сформирован. КМОП устроены иначе: сигнал из каждой ячейки считывается индивидуально. Для каждой ячейки заданы координаты в матрице и можно получить индивидуальный доступ к отдельной ячейке.

Пиксели можно объединять в строки (линейки) или матрицы. Каждый элемент улавливает разное количество фотонов, формирует анало-

говый сигнал, который можно преобразовать в массив, или матрицу цифровых значений. Таким образом формируют полутонаовые цифровые изображения.

Для получения цветных изображений используют сенсоры со светофильтрами. Наиболее распространенным вариантом является использование мозаики цветовых фильтров, где светочувствительный элемент накрыт фильтром, пропускающим фотоны, соответствующие диапазонам спектра красного, зеленого и синего цветов. Наиболее известным (но не единственным вариантом) является мозаика Байера, инженера фирмы Кодак (рис. 3.3). Из четырех соседних сенсоров два регистрируют зеленый цвет, остальные два – красный и синий (блок RGGB).

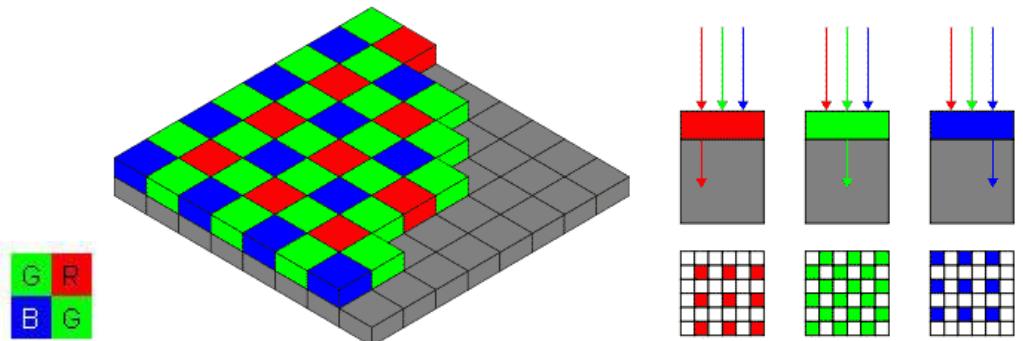


Рис. 3.3. Мозаика фильтров Байера

В результате матрица, описывающая изображение, состоит из мозаики пикселов трех основных цветов разной интенсивности. Зеленого цвета регистрируется больше, так как это соответствует особенностям человеческого зрения. Для восстановления недостающих цветов в каждом пикселе применяются специальные алгоритмы интерполяции мозаичного изображения (рис 3.4).

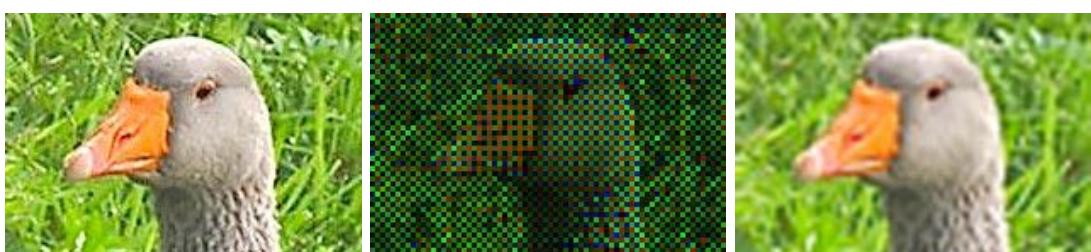


Рис. 3.4. Реальное изображение (слева), мозаика из фильтров Байера (в центре), интерполированное (справа)

В КМОП-матрице светофильтры не используются в принципе. Разделение светового потока на составляющие цвета происходит за счет того, что свет с разной длиной волны проникает в слой кремния на разную глубину. Например, волны синего цвета обладают наименьшей прони-

кающей способностью, а красного – наибольшей. Фотоэлемент каждого пикселя состоит из трех слоев (рис. 3.5). Толщина этих слоев рассчитана так, что они пропускают лучи с длинами волн определенного диапазона. При этом до нижнего слоя доходят фотоны с длиной волны, соответствующей красному цвету, до среднего – зеленому, до верхнего – синему. Образующиеся при этом электроны и «дырки» накапливаются в трех потенциальных ямах – по одной на каждый слой. Таким образом, для любого из пикселов матрицы есть данные по каждому из цветовых диапазонов. Это позволяет отказаться от мозаичных схем и связанных с ними потерю данных о цвете. Однако при распределении фотонов по слоям часть из них неизбежно поглощается при переходе из одного слоя в другой. В результате чувствительность матрицы ослабляется.

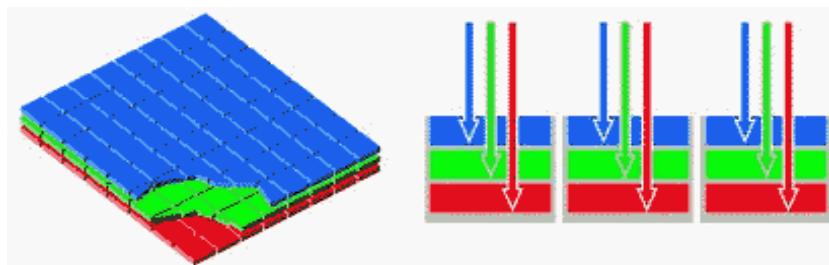


Рис. 3.5. Многослойные сенсоры

Отметим, что форма сенсора, называемого пикселом, может быть не только прямоугольной (рис. 3.6), но и L-образной, и шестиугольной (разработка Super CCD компании Fujifilm, 1999 г.).

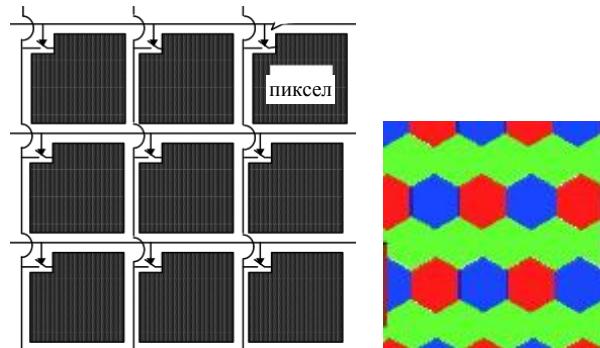


Рис. 3.6. Пиксели-сенсоры непрямоугольной формы

### 3.2. Сканер

*Сканер* – это устройство, которое при помощи аналого-цифрового преобразователя создает цифровое изображение некоторого документа. Для этого используются искусственное освещение и отражение в видимом диапазоне электромагнитного спектра. Процесс получения этого изображения называется сканированием. Сканеры бывают ручные, рулонные,

планшетные и проекционные. Разновидностью проекционных сканеров выступают слайд-сканеры, предназначенные для сканирования фотопленок. В высококачественной полиграфии используются барабанные сканеры, в качестве светочувствительного элемента у них применяется фотоэлектронный умножитель.

Главной особенностью сканера является формирование изображения ПЗС-сенсорами, выстроеными в линейку. В основном сканеры отличаются способом передвижения ПЗС-линейки или сканируемого документа друг относительно друга. Наиболее распространены планшетные сканеры.

Принцип работы однопроходного планшетного сканера (рис. 3.7) состоит в том, что вдоль сканируемого документа, расположенного на прозрачном неподвижном стекле, движется сканирующая каретка с источником света. Отраженный свет через оптическую систему сканера (состоящую из объектива и зеркал или призмы) попадает на ПЗС-сенсоры, расположенные линейно.

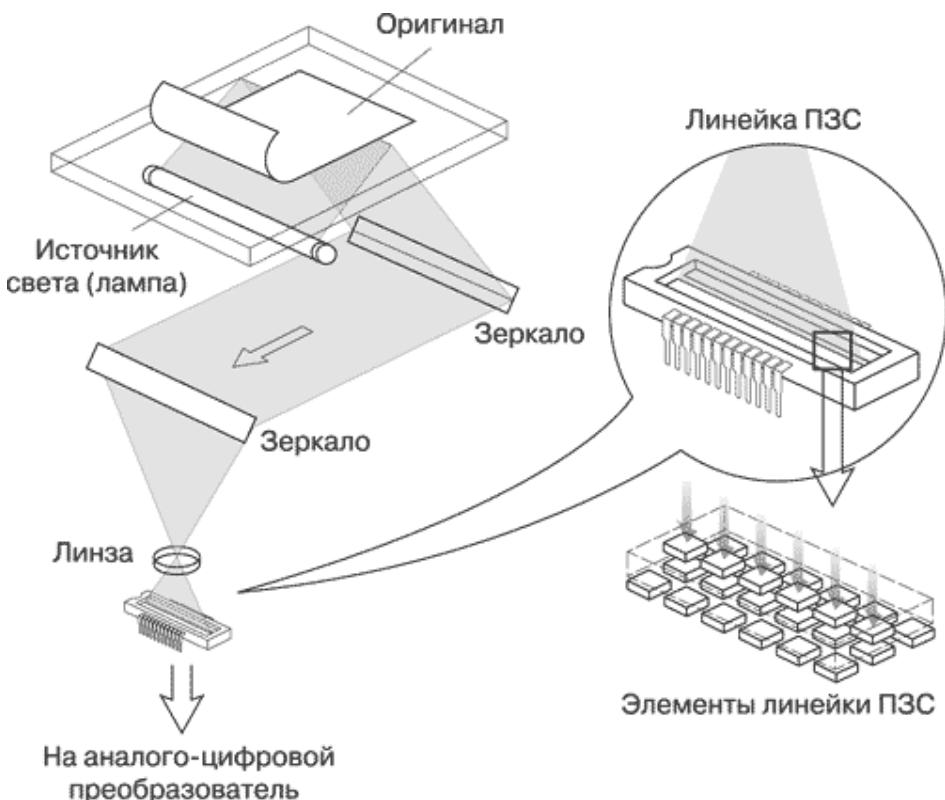


Рис. 3.7. Схема устройства планшетного сканера

При регистрации используется ПЗС-линейка, которая создает одну строку изображения. Для получения изображения в виде матрицы каретка смещается на фиксированное расстояние и регистрируется следующий фрагмент документа в виде строки пикселов.

Различают оптическое и механическое разрешение сканера. Например, если на нем указано разрешение 300x1200 dpi, то оптическое разрешение равно 300 dpi, а механическое (шаг смещения каретки) – 1200 dpi.

### 3.3. Цифровые фотоаппараты

Цифровой фотоаппарат по конструкции схож с обычным фотоаппаратом, но вместо фотопленки, на которой фиксируется изображение, у него имеются специальные датчики (фотоматрицы): матричные ПЗС или КМОП. Эти датчики преобразуют проецируемое на них объективом изображение в цифровую форму.

Две качественные характеристики, напрямую влияющие на результат съемки, – это динамический диапазон сенсора и разрядность представления цвета. Первая из этих характеристик отражает способность матрицы передавать световые оттенки, вторая относится не только к сенсору, но и аналого-цифровому преобразователю, переводящему электрические сигналы с выводов матрицы в цифровой код.

*Динамический диапазон* определяет количество оттенков серого (уровней яркости), которые способен различить сенсор цифровой камеры между абсолютно черным и абсолютно белым цветами. Чем шире динамический диапазон, тем выше сходство регистрируемого изображения с оригиналом.

Следует отметить, что до сих пор самым широким динамическим диапазоном обладает негативная фотопленка. Поэтому, несмотря на достижения цифровых технологий, при демонстрации фильмов в кинотеатрах еще используются пленочные проекторы.

Существует несколько вариантов конструкции цифровых фотоаппаратов: камеры с задней разверткой (рис. 3.8), трехкадровые камеры (рис. 3.9) и однокадровые камеры с одной или тремя фотоматрицами (рис. 3.10, 3.11) [13]. В настоящее время в основном применяются однокадровые фотокамеры.

Камера с задней разверткой производит сканирование в плоскости изображения. Принцип работы такой камеры похож на технологию, реализованную в сканерах с построчным считыванием информации. Сканирующая головка, содержащая линейку светочувствительных ПЗС, перемещается с небольшим шагом вдоль задней фокальной плоскости камеры поперек изображения, регистрируя за каждый шаг одну строку пикселов. Камеры, в которых применяется такой принцип, позволяют получать изображения с высоким разрешением, но время экспозиции может достигать нескольких минут, что делает технику задней развертки *не-пригодной для съемки движущихся объектов* или при работе со вспышкой. Во время сканирования затвор камеры остается открытый, поэтому

необходимо применять постоянное освещение, так как ни вспышка, ни стробоскоп в данном случае не годятся.

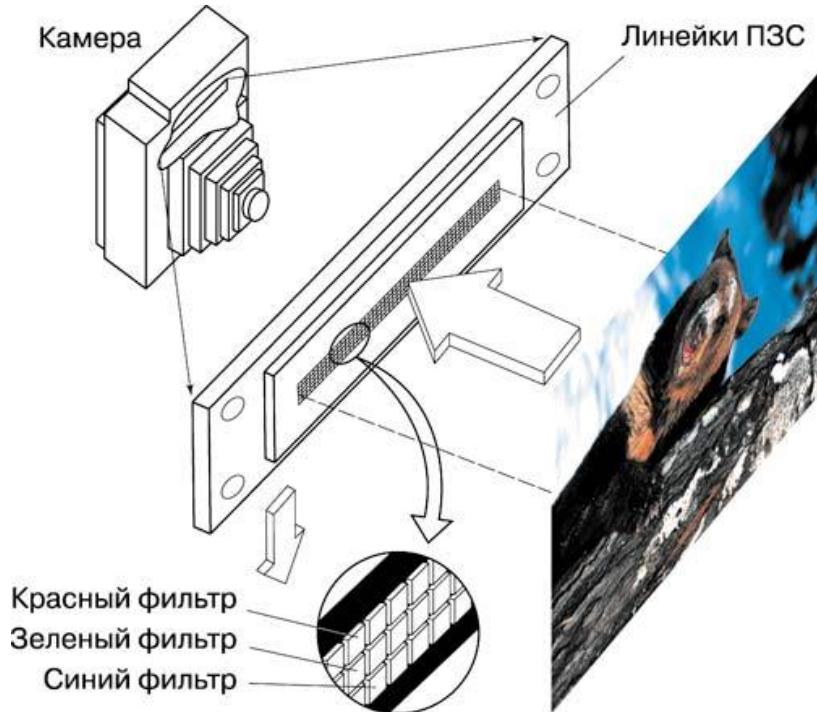


Рис. 3.8. Принцип действия камеры с задней разверткой

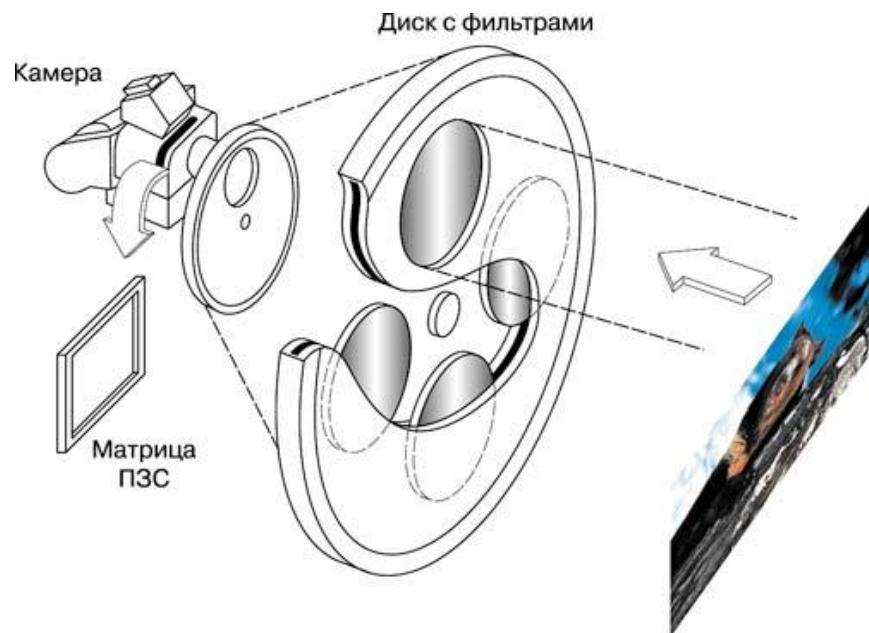


Рис. 3.9. Принцип действия трехкадровой камеры

Трехкадровая камера предназначена для регистрации цветных изображений неподвижных объектов (рис. 3.9). В качестве светочувствитель-

ного датчика используется двухмерная матрица ПЗС. Плоские двухмерные матрицы имеют гораздо меньшее разрешение, чем линейные. Каждый элемент матрицы формирует один элемент (пиксель) изображения. Экспозиция производится с такой скоростью, что можно пользоваться освещением от обычной вспышки. Для регистрации цветного изображения нужно сделать три отдельных снимка через три светофильтра (красный, зеленый и синий). Между экспонированием диск с фильтрами поворачивается таким образом, чтобы в момент съемки перед матрицей находился красный, зеленый или синий светофильтры.

Технология, реализованная в однокадровых камерах с одной матрицей, обеспечивает высокую скорость цифрования изображения, но характеризуется более низким разрешением и худшей цветопередачей, чем «многоснимочная» технология. Такие камеры называются также камерами с вычислением цветов.

Как и в трехкадровой, в однокадровой камере с одной матрицей (рис. 3.10) применяется плоская матрица, но данные о цвете регистрируются не через отдельные фильтры, а через нанесенный на поверхность матрицы ПЗС пленочный фильтр, состоящий из красных, зеленых и синих элементов. Данные о каждом элементе изображения регистрируются только в одном из трех цветов, например в красном. Для добавления к нему надлежащих долей зеленого и синего программа обработки интерполирует данные о цветах соседних точек. Поскольку требуется всего одна экспозиция, однокадровые камеры обеспечивают съемку движущихся объектов.

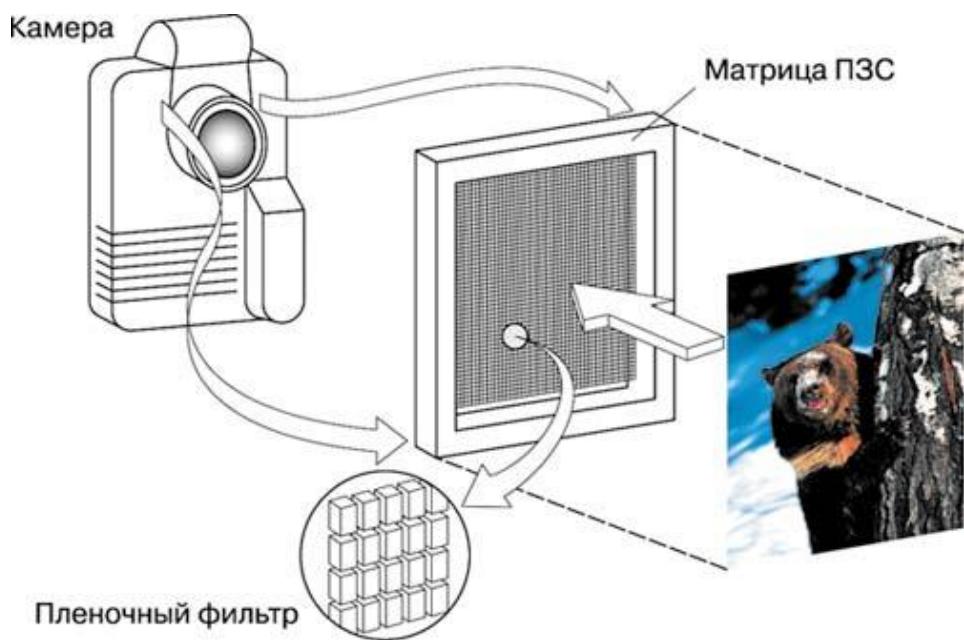


Рис. 3.10. Принцип действия однокадровой камеры с одной матрицей

Принцип действия однокадровой камеры с тремя матрицами (камеры для однокадровой цветной съемки) заключается в оптическом разделении приходящего света призмой на красную, зеленую и синюю составляющие, причем каждая из них направляется на свою матрицу (рис. 3.11). В одних моделях каждая матрица регистрирует свой цвет, в других объединяются плоская матрица, на поверхность которой нанесены красный и синий пленочные фильтры, и две дополнительные матрицы с зелеными фильтрами. Во втором случае интерполяция производится только по двум цветам, что приводит к повышению качества зафиксированного изображения. Недостаток этого способа регистрации – относительно невысокое разрешение.

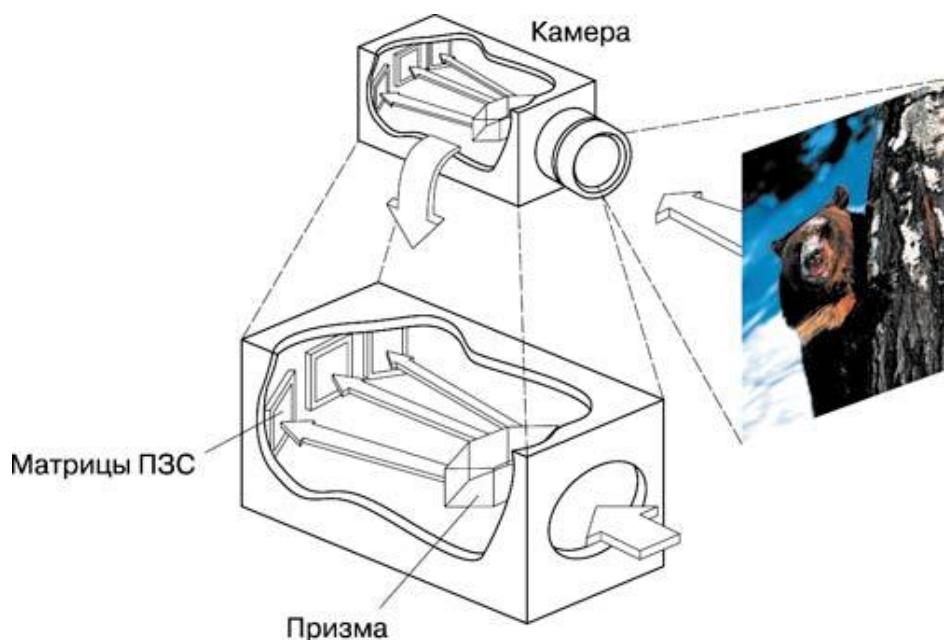


Рис. 3.11. Принцип действия однокадровой камеры с тремя матрицами

При использовании трехслойных (X3)-фотоматриц (рис. 3.12) компании Foveon цветodelение на основные цвета RGB проводится в толще полупроводникового материала послойно с применением физических свойств кремния, заключающихся в том, что с увеличением длины световых волн растет и глубина их проникновения в кремний (рис. 3.12, а).

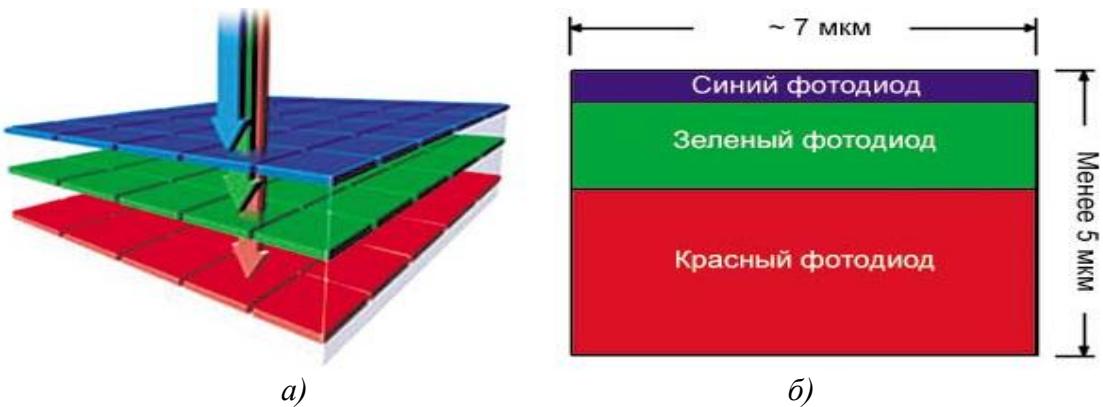


Рис. 3.12. Трехслойная фотоматрица: а) принцип действия; б) структура слоев

Фотодиоды, созданные чередованием зон проникновения света, размещаются один под другим на определенных глубинах для улавливания фотонов синего, зеленого и красного цветов. Синяя часть спектра поглощается верхним слоем (толщина 0,2 мкм), зеленая – средним (0,4 мкм), а красная – нижним (более 2 мкм). Толщина каждого слоя выбрана по результатам экспериментальных исследований глубины проникновения квантов соответствующего спектрального диапазона в кремний. Слои, в которых происходит фотоэффект, разделены дополнительными тонкими зонами низколегированного кремния и имеют отдельные выводы сигнала. Таким образом получается датчик, регистрирующий информацию о трех цветовых компонентах изображения в одной точке, точно соответствующей координатам формируемого пикселя.

Достоинствами ХЗ-фотоматриц является то, что они не требуют установки цветных фильтров, а следовательно, и повышения соотношения сигнал-шум, так как фильтры поглощают примерно две трети светового сигнала и имеют потенциально более высокое разрешение.

Недостатками ХЗ-фотоматриц можно назвать невысокую точность цветопередачи, так как в наибольшей степени она определяется свойствами кремния, а также относительно высокий уровень цифрового шума, оттого, что часть фотонов поглощается в «не своей» области.

Для получения качественной фотографии важнее не количество мегапикселов фотоаппарата, а другие параметры: размеры активной области матрицы, ее чувствительность, динамический диапазон, точность цветопередачи. Наибольшее значение имеют первые три параметра, причем в ПЗС-матрице все они неразрывно связаны друг с другом.

На рис. 3.13 показано соотношение размеров кадра пленки 24x36 мм и ПЗС-матриц.

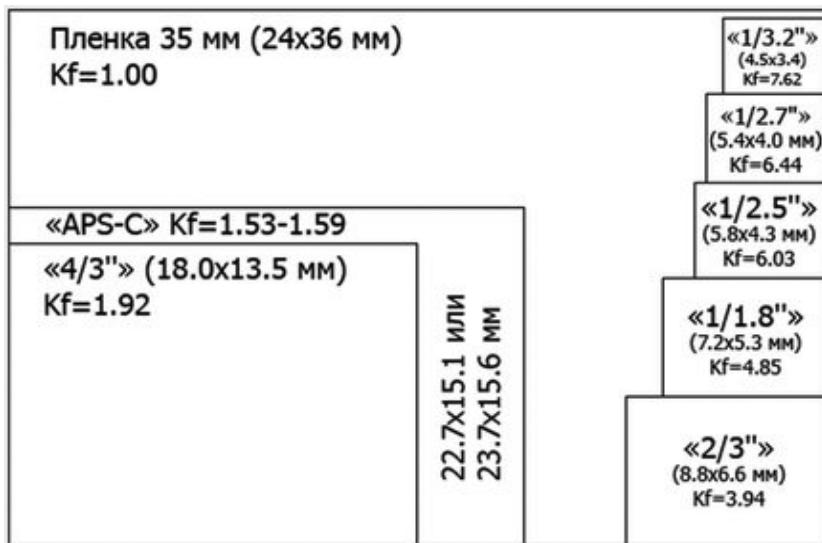


Рис. 3.13. Соотношение размеров кадра пленки 24x36 мм и ПЗС-матриц

Указаны размеры матриц в дюймах и мм, а Kf – это коэффициент, показывающий во сколько раз диагональ матрицы меньше диагонали кадра пленки. Физический размер матрицы напрямую связан с количеством попадающего на матрицу света, поэтому чем больше матрица, тем качественней будут фотографии в условиях плохой освещенности.

Размер матрицы влияет на глубину резкости: меньше матрица – большая глубина, но при этом, например, портретные фотографии выглядят плоскими. Увеличение размера матрицы влечет за собой увеличение размера и стоимости фотоаппарата.

Отметим, что пока цифровые фотоаппараты не превосходят плечевые. Они не могут снимать в условиях низкой освещенности без применения источников искусственного света, а профессиональная пленка светочувствительностью в 3200 единиц ISO вытянет снимок даже при свете обычной свечи.

### 3.4. Получение изображений с помощью ультразвука

*Ультразвуковое исследование (УЗИ)* – это неинвазивное исследование живого организма с помощью ультразвуковых волн (рис. 3.14, 3.15). Физической основой УЗИ является пьезоэлектрический эффект.

При деформации монокристаллов некоторых химических соединений под воздействием ультразвуковых волн на поверхности этих кристаллов возникают противоположные по знаку электрические заряды, т. е. прямой пьезоэлектрический эффект. При подаче на них переменного электрического заряда в кристаллах возникают механические колебания с излучением ультразвуковых волн. Один и тот же пьезоэлемент может быть попеременно приемником и источником ультразвуковых волн.

В системах медицинской ультразвуковой диагностики обычно используют частоты от 2 до 10 МГц. Разрешающая способность современных ультразвуковых аппаратов достигает 1–3 мм. Любая среда, в том числе и ткани организма, препятствует распространению ультразвука, т. е. обладает различным акустическим сопротивлением, величина которого зависит от их плотности и скорости распространения звуковых волн. Достигнув границы двух сред с различным акустическим сопротивлением, пучок ультразвуковых волн претерпевает изменения: одна его часть продолжает распространяться в новой среде, частично поглощаясь ею, другая – отражается. Коэффициент отражения зависит от разности величин акустического сопротивления граничащих друг с другом тканей: чем больше это различие, тем больше отражение и амплитуда зарегистрированного сигнала, тем светлее и ярче он будет выглядеть на экране аппарата. Полным отражателем является граница между тканями и воздухом.

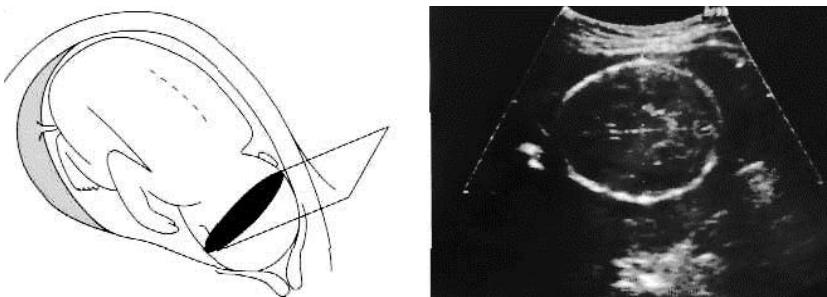


Рис. 3.14. Измерение размера головки плода: *а*) схема; *б*) изображение



Рис. 3.15. Изображение плода на 20-й неделе беременности, полученное методом УЗИ 3Д

Так как на границе перехода ультразвука из воздуха на кожу происходит отражение 99,99 % колебаний, то при ультразвуковом сканировании пациента необходимо смазывание поверхности кожи водным желе, которое выполняет роль переходной среды. Отражение зависит от угла падения луча (наибольшее при перпендикулярном направлении) и частоты ультразвуковых колебаний (при более высокой частоте большая часть отражается). Для исследования органов брюшной полости, полу-

сти малого таза используется частота 2,5–3,5 МГц, для щитовидной железы – 7,5 МГц.

Генератор работает в импульсном режиме, посылая около 1 000 импульсов в секунду. В промежутках между генерираием ультразвуковых волн пьезодатчик фиксирует отраженные сигналы. Отраженные эхосигналы поступают в усилитель и специальные системы реконструкции, после чего появляются на экране монитора в виде отражения среза тела, окрашенного в различные оттенки черно-белого цвета, т. е. полуточнового 2D-изображения (рис. 3.14). Оптимальным является наличие не менее 64 оттенков серого.

В настоящее время с помощью УЗИ можно получить 3D-изображения (статичные трехмерные изображения) в результате объемного сканирования (рис. 3.15). УЗИ 4D – это движущиеся изображение в формате 3D, которое можно наблюдать в режиме реального времени. В качестве четвертого измерения выступает время.

### **3.5. Рентгеновские изображения**

Рентгеновские лучи охватывают диапазон электромагнитного спектра с длиной волн от 1 до 0,01 нм. Глаз человека их не видит, но они способны проникать через твердые предметы.

Вильгельм Рентген за открытие излучений этого типа в 1901 г. стал первым Нобелевским лауреатом по физике. Идея получения рентгеновского изображения основана на различном ослаблении излучения при прохождении через ткани с разной плотностью и последующей регистрацией. В результате получается изображение всех тканей, спроектированное на плоскость (рис. 3.16).

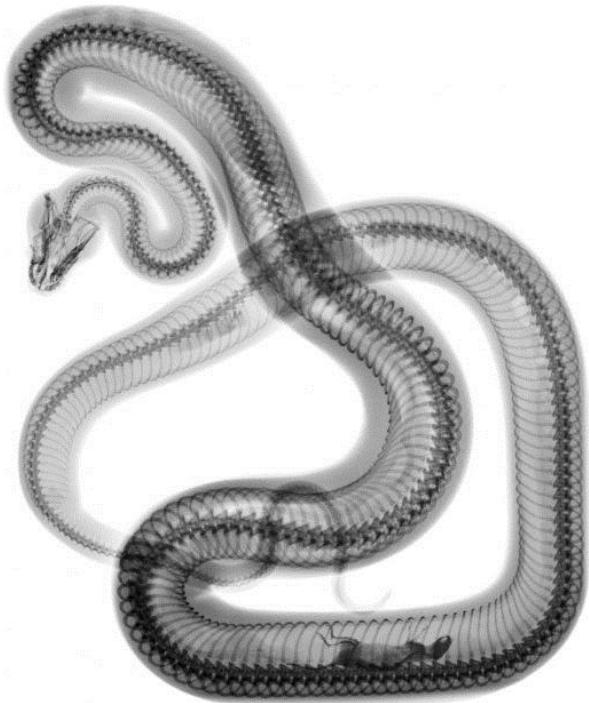


Рис. 3.16. Пример рентгеновского цифрового снимка змеи, проглотившей мышь

В современных цифровых аппаратах регистрация выходного излучения производится на специальную электронную матрицу. В ней накапливается заряд, создаваемый при поглощении рентгеновских лучей и его построчной передаче на усилители заряда во время сканирования. Преобразование поступающего рентгеновского излучения в заряд в аморфном кремнии с целью электронного считывания может быть реализовано тремя методами: прямым, методом фотопроводимости и сцинтилляционным. Каждый метод имеет свои преимущества и недостатки. Во всех методах заряд перед считыванием накапливается в течение периода кадра.

В *прямом методе* рентгеновские лучи попадают на диод из кремния, где создаются пары «дырка – электрон». Устройства, использующие прямой метод преобразования, изготавливаются из кристаллического кремния, но на практике применяются только решетки из одной или двух линий, при этом даже такие решетки отличаются дороговизной.

В *методе фотопроводимости* используются материалы с более высокой, чем у кремния, способностью поглощать рентгеновское излучение. Они наносятся на множество проводящих пластин для сбора заряда, поставляемых вместе с накопительным конденсатором, а также создают пары «дырка – электрон» при поглощении рентгеновских лучей. Однако генерируемый заряд хранится за пределами слоя во избежание перекрестных искажений. Приложенное поле отделяет заряд и направляет его на расположенную непосредственно внизу пластину для сбора заря-

да с целью сохранения резкости изображения. В настоящее время селен является единственным используемым в производстве фотопроводящим материалом. Он имеет относительно низкую способность поглощать рентгеновское излучение и требует примерно 50 электронвольт для создания пары «дырка – электрон». Это ограничивает как минимально необходимую дозу, так и величину генерируемого сигнала.

В *сцинтилляционном методе* используется состав, который поглощает рентгеновские лучи и преобразует их энергию в видимый свет. Эффективный сцинтиллятор производит много фотонов света для каждого поступающего рентгеновского фотона. Сцинтилляторы могут представлять собой гранулы (например, люминофоры) или кристаллы (например, йодид цезия).

Практически все современные системы рентгеновской визуализации используют модели с детекторами ПЗС. Качество полученного рентгеновского изображения определяется тремя основными параметрами: напряжением, подаваемым на рентгеновскую трубку, силой тока и временем работы трубки. В зависимости от исследуемых анатомических образований, массы и размеров объекта исследования эти параметры могут существенно изменяться.

Обычно разрешающая способность достигает 0,5 мм (одна пара различаемых линий на миллиметр соответствует двум пикселам на миллиметр), но можно получить и более высокое разрешение – до 26 пар линий на миллиметр, что соответствует 0,02 мм. Лучший на сегодняшний день ПЗС-детектор рентгеновских излучений фирмы Nova Imaging (США) имеет до 16 мегапикселов и позволяет получать цифровое изображение области 43x43 см.

Недостатки рентгенографии:

- информативность классической рентгенографии значительно ниже таких методов визуализации, как компьютерная томография и магнитно-резонансная томография. Обычные рентгеновские изображения отражают проекционное наслаждение сложных анатомических структур, т. е. их суммарную рентгеновскую тень в отличие от послойных серий изображений, получаемых томографическими методами;
- без применения контрастирующих веществ рентгенография практически неинформативна для анализа изменений в мягких тканях.

### **3.6. Томографические изображения**

*Томография* – это метод получения послойного изображения внутренней структуры некоторого объекта. Томографию применяют при производстве деталей ответственного применения (для авиации, космоса, автомобилестроения), в электронике и микроэлектронике, в материа-

ловедении, но в основном в медицине. Томографические изображения получают с помощью устройств двух типов: на базе рентгеновского излучения и ядерного магнитного резонанса.

### ***3.6.1. Магнитно-резонансная томография***

*Магнитно-резонансная томография (МРТ)* – это томографический метод исследования внутренних органов и тканей с использованием физического явления ядерного магнитного резонанса (ЯМР). Он основан на измерении электромагнитного отклика ядер атомов водорода на возбуждение их определенной комбинацией электромагнитных волн в постоянном магнитном поле высокой напряженности (рис. 3.17).

Пол Лотербур в 1973 г. опубликовал идею создания изображений с помощью МРТ. Но в СССР способ и устройство для ЯМР-томографии этого типа еще в 1960 г. предложил В.А. Иванов (термин «ЯМР-томография» был заменен на МРТ в 1986 г.). За изобретение метода МРТ Питер Мэнсфилд и Пол Лотербур получили в 2003 г. Нобелевскую премию в области медицины.

МРТ позволяет визуализировать с высоким качеством головной, спинной мозг и другие внутренние органы. Эффект ЯМР возникает при взаимодействии между протонами биологических тканей, постоянным или переменным магнитным полем и энергией радиочастотных импульсов, испускаемых катушкой, помещенной около исследуемой части тела. Под действием радиочастотных импульсов протоны атомов водорода временно переходят на более высокий энергетический уровень. Возвращение протонов к равновесному состоянию сопровождается выделением энергии в виде импульсов определенной (резонансной) частоты. Эту энергию можно измерить при помощи приемной катушки. Для формирования изображения полученные сигналы обрабатываются с помощью преобразования Фурье.

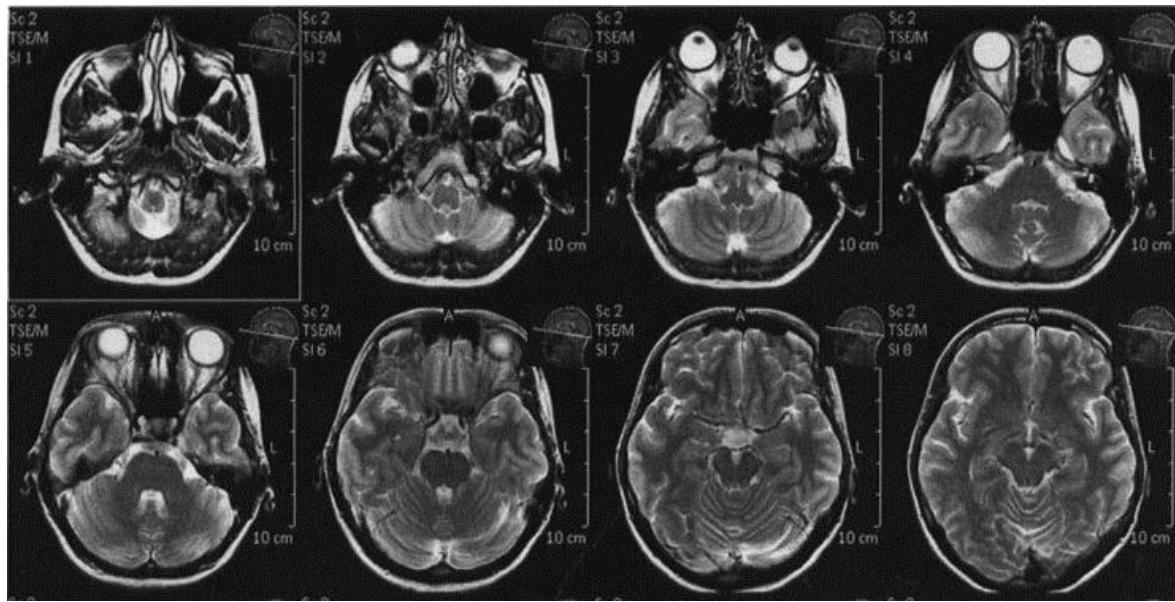


Рис. 3.17. МРТ-изображения головы человека

МРТ позволяет получать изображение в разных плоскостях, не изменяя положения больного. Изображение в каждой плоскости формируется за 5–10 мин. В отличие от компьютерной томографии, изменение положения больного во время исследования искажает все изображения.

### 3.6.2. Компьютерная томография

Идея компьютерной томографии (КТ) заключается в компьютерной реконструкции изображения поперечного среза тела на основании анализа поглощения рентгеновского излучения. За разработку метода рентгеновской компьютерной томографии А. Маккормаку и Г. Хаунсфилду в 1979 г. была присуждена Нобелевская премия.

Трубка испускает тонкий веерообразный пучок рентгеновских лучей перпендикулярно длинной оси тела. Этот пучок может быть достаточно широким и охватывать весь диаметр тела. Толщину пучка можно регулировать от 1 до 10 мм, при этом изменяется толщина исследуемого среза ткани. Пропускаемый через пациента пучок рентгеновских лучей фиксируется не пленкой, а системой специальных детекторов. Рентгеновские электроны генерируют в детекторах электрические сигналы. Чем сильнее интенсивность достигшего детектора первичного луча, тем сильнее электрический сигнал.

Фиксируя интенсивность пропущенного излучения, можно рассчитать ослабление первичного луча. КТ-детекторы примерно в 100 раз чувствительнее рентгеновской пленки при определении различий в интенсивности излучения. Излучения регистрируются по шкале Хаунсфилда (HU) от –1024 до +3071.

Для получения компьютерной томограммы (среза) на выбранном уровне необходимо выполнить следующие операции: 1) формирование рентгеновского луча требуемой ширины (коллимирование); 2) сканирование пучком рентгеновского излучения, осуществляемого движением (вращательным и поступательным) вокруг неподвижного объекта устройства «излучатель – детекторы»; 3) измерение излучения и определение его ослабления с последующим преобразованием результатов в цифровую форму; 4) компьютерный синтез томограммы по совокупности данных измерения, относящихся к выбранному слою; 5) построение изображения исследуемого слоя и отображение его на экране монитора.

Воздух или газ (вне зависимости, где они находятся) имеют значения около  $-1000$  HU и выглядят черными, кости ( $+400$  HU и выше) – белыми. Другие органы, ткани и участки патологических изменений отображаются оттенками серого между этими двумя крайностями (от 0 до  $+300$ ). Поскольку отобразить и различить  $2^{10}$  оттенков нельзя, при визуализации данных КТ выбирается диапазон определенных значений (окно) для анализа нужных тканей (рис. 3.18).

Качество изображения растет при увеличении числа детекторов и количества регистрируемых проекций за один оборот трубки и при увеличении первичной матрицы. Однако увеличение количества регистрируемых проекций ведет к повышению лучевой нагрузки, применение большей первичной матрицы – к увеличению времени обработки среза или необходимости устанавливать дополнительные специальные процессоры видеозображения.

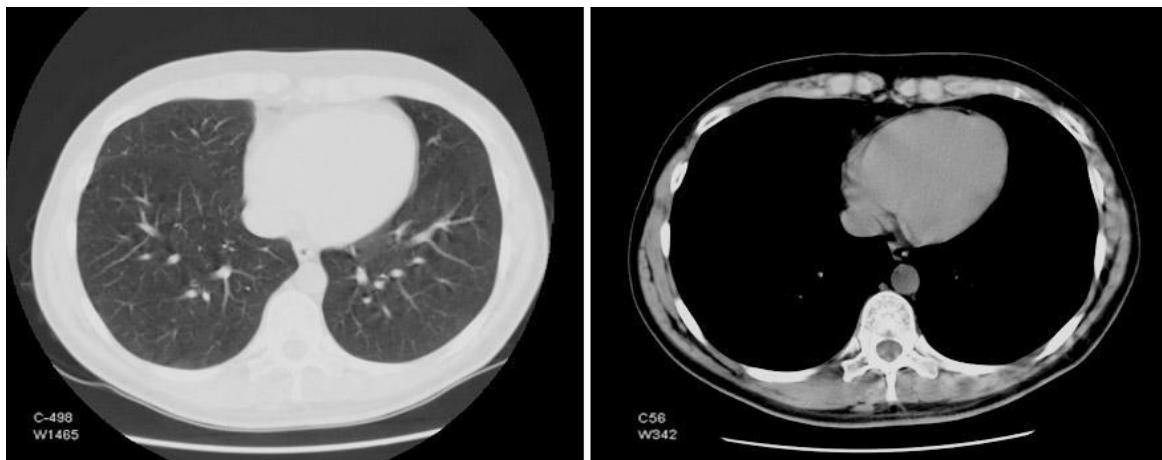


Рис. 3.18. Томограмма грудной клетки в легочном (слева) и мягкотканом (справа) окнах шкалы Хаунсфилда

Современный томограф позволяет получать срезы толщиной 1–2, 5 и 10 мм со скоростью сканирования слоя 1–3 с, полное исследование головного мозга занимает 2–3 мин.

### ***3.6.3. Отличия изображений компьютерной томографии от магнитно-резонансной***

На обычном рентгеновском снимке лучше всего различимы кости, поэтому КТ используется для исследования позвоночника, основания черепа и височных костей. Кроме того, на КТ-изображении хорошо видно свежее кровотечение, поэтому этот метод применяют при исследованиях больных с травмами головы, грудной клетки, брюшной и тазовых полостей, а также инсультов в ранней стадии. Использование контрастного вещества позволяет получить качественное изображение сосудов, почек и кишечника.

На снимках МРТ кости тоже видны, но лучше видны мягкие ткани (хрящи, мышцы, связки, мозг и т. д.). МРТ чувствительнее КТ на ранних стадиях поражения центральной нервной системы. Поэтому данный метод показан при подозрении на разрыв связок и для исключения грыжи межпозвонкового диска, а также широко используют в нейрохирургии и неврологии (старые травмы головного мозга, инсульты в поздней стадии, подозрения на опухоли спинного и головного мозга и т. д.).

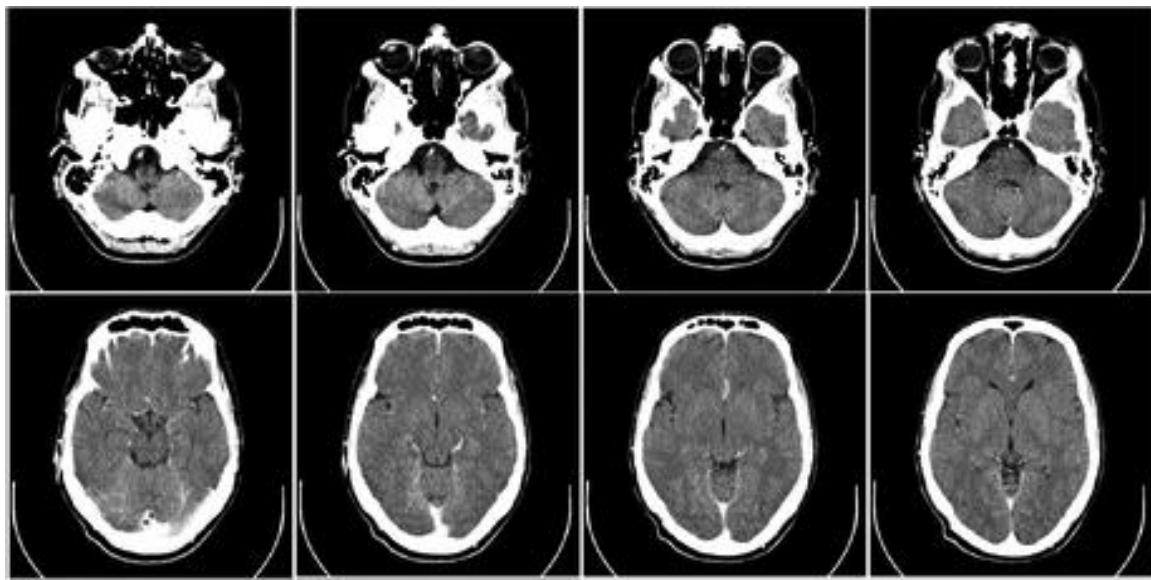


Рис. 3.19. КТ-изображения головы человека

Разницу в изображении разных тканей методами МРТ и КТ можно увидеть, сравнив рис. 3.17 и 3.19.

## **3.7. Изображения дистанционного зондирования Земли**

Под дистанционным зондированием понимают получение информации о состоянии поверхности Земли по измерениям на расстоянии без непосредственного контакта датчиков с поверхностью. Датчики уста-

навливаются на спутниках, самолетах и беспилотных летательных аппаратах. Диапазон измеряемых электромагнитных волн – от долей микрометра (видимое оптическое излучение) до метров (радиоволны). Методы ДЗЗ могут быть *пассивные*, т. е. использующие естественное отраженное или вторичное тепловое излучение объектов на поверхности Земли, обусловленное солнечной радиацией, и *активные* – использующие отражение, инициированное искусственным источником, установленном на летательном аппарате. Получение информации для ДЗЗ основывается на том, что объекты разных типов (почва, вода, растительность, здания, техника и др.) по-разному отражают и поглощают излучение в разных диапазонах электромагнитного спектра.

В настоящее время для ДЗЗ используются разные системы, формирующие изображение исследуемой местности. Их можно разделить на несколько классов, различающихся по спектральному диапазону используемого электромагнитного излучения, по типу приемника регистрируемого излучения и методу зондирования (активный или пассивный):

- фотографические системы;
- сканирующие системы видимого и инфракрасного диапазона (телеизионные, сканирующие радиометры и многоспектральные сканеры);
- телевизионные оптические системы;
- радиолокационные системы бокового обзора;
- сканирующие СВЧ радиометры.

Атмосфера поглощает часть лучей электромагнитного спектра, но в ней есть несколько окон прозрачности (рис. 3.20), которые пропускают электромагнитные волны с минимальными искажениями. Чаще всего в методах ДЗЗ используется окно прозрачности, соответствующее оптическому диапазону, объединяющему видимую (380–720 нм), ближнюю инфракрасную (720–1300 нм) и среднюю инфракрасную (1300–3000 нм) области.

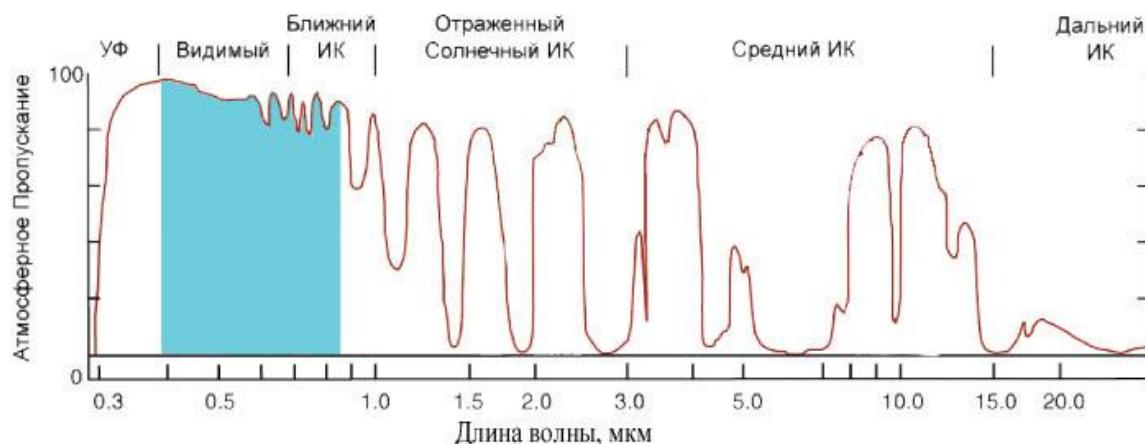


Рис. 3.20. Степень прозрачности атмосферы для разных длин волн спектра

Полученные снимки классифицируются по пространственному разрешению (размеру участка местности, соответствующему одному пикселю):

- низкому (300–1000 м);
- среднему (50–200 м);
- относительно высокому (20–40 м);
- высокому (10–20 м);
- очень высокому (1–10 м);
- сверхвысокому (0,3–0,9 м).

Рассмотрим основные типы изображений, получаемые средствами ДЗЗ.

### ***3.7.1. Мультиспектральные изображения***

Мультиспектральные камеры – самые распространенные виды сенсоров, установленных на спутниках ДЗЗ. Отраженные волны проходят сквозь светофильтры, выделяющие определенный участок электромагнитного спектра, затем через фокусирующие линзы, дисперсирующее устройство и попадают на ПЗС-линейку, где регистрируются в виде полуточнового изображения. Число фильтров определяет количество каналов мультиспектрального изображения (табл. 3.1). Как правило, все изображения записываются в один файл в формате Geotiff.

Таблица 3.1  
Спутники со съемочными системами высокого разрешения

Спутник	Pan, мкм	B, мкм	G, мкм	R, мкм	NIR, мкм	Другие	Разрешение, м	Бит
WorldView-3	.45-.80	.4-.5 фиолет. и .45-51	.51-.58 и .585-.625 желтый	.63-.69 и .705-.745	.77-.895 и .86-1.04	Нет	.31; 1.24	11
WorldView-2	То же	То же	То же	То же	То же	Нет	.5; 2	11
GeoEye-2	.45-.90	.45-.52	.52-.60	.625-.695	.76-.90	Нет	.25; 1.65	11
Ikonos	То же	То же	.51-.60	.63-.70	То же	Нет	.1; 4	11
QuickBird	То же	То же	.52-.60	.63-.69	То же	Нет	.61; 2.44	11
OrbView-3	То же	То же	.52-.60	.625-.695	То же	Нет	.1; 4	11
Spot-6 и 7	.45-.745	То же	.53-.59	.625-.695	.76-.89	Нет	2; 6	12
СИЧ-2 (Украина)	.50-.90	нет	.50-.60	.61-.68	.80-.89	1.55-.7 мкм	7.8; 39.5	8
БелКА и Канопус-В	.45-.85	.46-.52	.52-.60	.63-.69	.75-.86	Нет	2.1; 10.5	8

Как видно из табл. 3.1, в каждом канале регистрируется полуточновое изображение с большим динамическим диапазоном значений яркости. Следует отметить два факта: чем шире спектральный диапазон отражаемого сигнала, тем выше разрешение (пример: панхроматический режим имеет разрешение в 2–4 раза выше спектральных); спектральные

диапазоны могут не пересекаться и наоборот, все зависит от используемых типов фильтров.

### 3.7.2. Гиперспектральные изображения

Ключевым моментом при построении гиперспектральных систем является способ формирования трехмерной структуры данных на двух- (ПЗС-матрицу) или одномерный массив (ПЗС-линейку) фотоприемников. Устройство получения таких изображений называется гиперспектрометром. Наиболее распространенными на сегодняшний день являются гиперспектрометры, которые в каждый момент времени регистрируют узкую полосу поверхности под собой (рис. 3.21). Первый гиперспектрометр Ais-1 разработало NASA (США) в 1982 г.

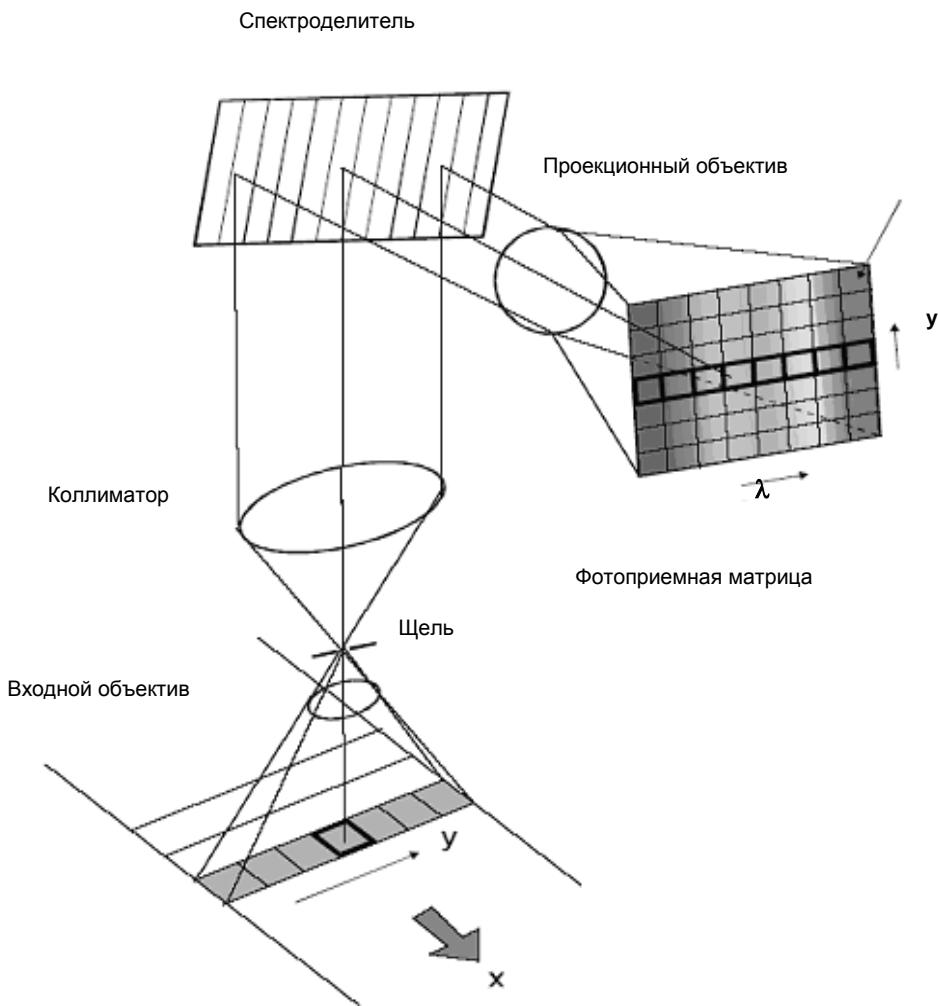


Рис. 3.21. Принципиальная схема гиперспектрометра

Формирование изображения узкого участка поверхности поперек направления полета производится посредством щели на задней фокальной плоскости входного объектива.

*Гиперкубом* называют трехмерный массив данных, два измерения которого соответствуют пространственным координатам ( $x, y$ ), третье – спектральной координате  $\lambda$  (рис. 3.22). Его получают при съемке кадра путем фиксации одной из координат (либо пространственной, например  $x$ , либо спектральной  $\lambda$ ). За один кадр фотоприемная матрица формирует двухмерный срез гиперкуба с фиксированной третьей координатой.

После коллимирующего объектива отраженный сигнал в параллельных лучах попадает на призму. В призме происходит его разложение на волны разной длины, а затем они проецируются на фотоприемную матрицу. Таким образом на матрице формируется срез гиперкуба для фиксированной пространственной координаты  $x$ . Большинство спектрометров регистрируют 100–200 каналов шириной (спектральным разрешением) по 0,4–2,5 нм каждый в диапазоне 400–2500 нм без промежутков и наложений.

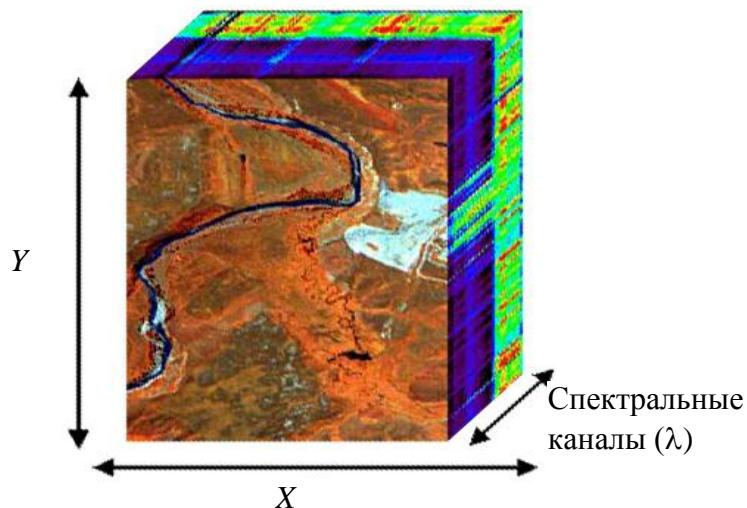


Рис. 3.22. Гиперспектрометр сканирует пространство вдоль оси  $X$  (или  $\lambda$ )

Мультиспектральные и гиперспектральные снимки могут покрывать одинаковый спектральный диапазон, но отличаются способами регистрации изображений. Отметим разницу между мультиспектральными и гиперспектральными изображениями. Мультиспектральное является набором из полутоновых изображений одного участка местности, зарегистрированных в нескольких узких диапазонах спектра, выделенных путем фильтрации. Гиперспектральное получают путем разложения с помощью призмы солнечного излучения, отраженного от участка Земли, соответствующего одному пикселу с координатами ( $x, y$ ), на линейку сенсоров, каждый из которых регистрирует сигналы в узком диапазоне спектра. Все значения яркости, зарегистрированные линейкой, присваиваются одному пикселу.

Например датчик, формирующий 20 изображений, создает гиперспектральное изображение, если оно представляет собой отражение в диапазоне от 500 до 700 нм, пропущенное через призму и зарегистрированное 20 сенсорами, каждый из которых охватывает часть этого диапазона, равную 10 нм в ширину. В то время как датчик, регистрирующий 20 дискретных полос в этом же диапазоне, которые получены путем фильтрации лучей, т. е. пропуска и регистрации волн только одной полосы – мультиспектральное. Полученные результаты будут отличаться, поскольку при выделении узких диапазонов спектра используются физически разные способы, а фильтрацией невозможно полностью разделить волны разной длины.

### 3.7.3. Радиолокационные изображения

Радиолокационная съемка основана на способности объектов по-разному отражать узконаправленные на них радиоимпульсы сантиметрового диапазона (0,3–100 см).

Свойства материалов влияют на интенсивность отраженного сигнала. Коэффициент диэлектрической проницаемости характеризует способность материала поглощать электрическую энергию и зависит от используемой частоты (рис. 3.23). Различие значений коэффициентов для разных материалов обуславливает возможность их выявления при помощи радарной съемки (табл. 3.2). Регистрировать сигналы радарного диапазона и получать изображения можно в условиях облачности и в темное время суток.

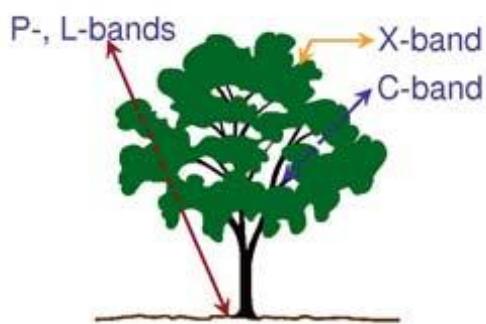


Рис. 3.23. При разных частотах сигнал отражается от разных поверхностей

Нагретые объекты излучают энергию и в радиодиапазоне, хотя радиотепловое излучение по интенсивности значительно уступает тепловому излучению в инфракрасном диапазоне. Радиотепловое излучение, как и инфратепловое, формируется поверхностным излучательным слоем, толщина которого тем больше, чем длиннее волна излучения.

Таблица 3.2

Частоты и длины волн, используемые в радиолокации

Название	Частота, МГц	Длина волны, см
Ka-band	40,000–26,000	0,8–1,1
K-band	26,500–18,500	1,1–1,7
Ku-band	18,500–12,500	1,7–2,4
X-band	12,500–8,000	2,4–3,8
C-band	8,000–4,000	3,8–7,5
S-band	4,000–2,000	7,5–15,0
L-band	2,000–1,000	15,0–30,0
P-band	1,000–300	30,0–100,0

Интенсивность радиотеплового излучения характеризуется так называемой *радиояркостной температурой* – произведением абсолютной температуры и коэффициента излучения. Величина коэффициента излучения в радиодиапазоне значительно варьирует в зависимости от электрических свойств (а значит, от влажности и солености), кристаллической структуры и характера поверхности объекта. Наибольшее его значение имеет растительность (0,9), затем следуют почвы (0,8), водные поверхности (0,3) и, наконец, металлические предметы. Таким образом, различия в радиояркостной температуре земных объектов определяются их излучательной способностью, а не физической температурой. Летом наибольшую радиояркостную температуру имеет растительный покров (300 К) и меньшую – водные поверхности (100 К). Но самые «холодные» в радиодиапазоне – это металлические предметы.

Активное зондирование земной поверхности основано на измерении вариаций рассеянного излучения. Информация о поверхности представлена отраженной электромагнитной волной. Радиолокатор испускает волну. Она характеризуется длиной и частотой (равной отношению скорости распространения в среде к длине волны, в вакууме скорость равна скорости света). Кроме направления волны важна ее поляризация. Конструктивно радиолокационные системы могут осуществлять излучение и прием на вертикальной и горизонтальной поляризациях сигнала (рис. 3.24). Принятый в результате сигнал зависит от типа поляризации при излучении и при приеме, его тип обозначается двумя буквами:

- вертикальная поляризация (тип ВВ, или VV);
- горизонтальная (тип ГГ, или GG);
- перекрестная (тип ВГ и ГВ, или VG и GV).

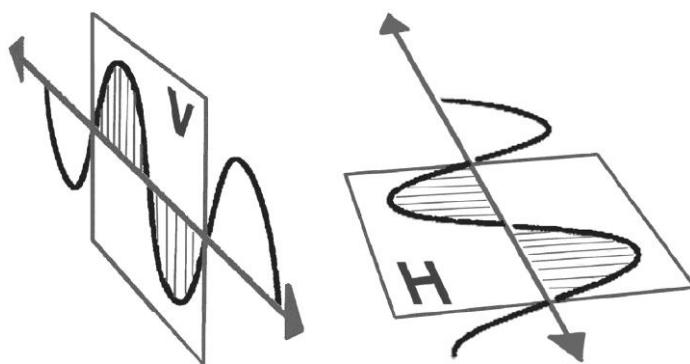


Рис. 3.24. Вектор электрического поля при вертикальной V (слева) и горизонтальной H (справа) поляризациях

Мощность сигнала, попадающего в приемную антенну, зависит от типа поляризации излучения и взаимного направления поляризаций при излучении и приеме. Коэффициент рассеяния водной поверхности при вертикальной поляризации выше, чем при горизонтальной и перекрестных поляризациях. Вертикальная поляризация при излучении и приеме (тип VV) применяется при анализе процессов на водной поверхности. Горизонтальная поляризация (тип GG) – для разделения льда и воды. Перекрестные поляризации используют для выделения на морской поверхности различных объектов (кораблей, трещин во льду и т. п.).

На рис. 3.25, а отчетливо видны проявления поверхностных атмосферных и морских процессов, на рис. 3.25, б – отражения судов. Вертикально поляризованный волна взаимодействует с вертикальными стеблями слоя растений, горизонтально поляризованный проникает сквозь слой. Комбинирование разных видов поляризации может улучшать детальность классификации, например позволяет определять различные классы растительности.

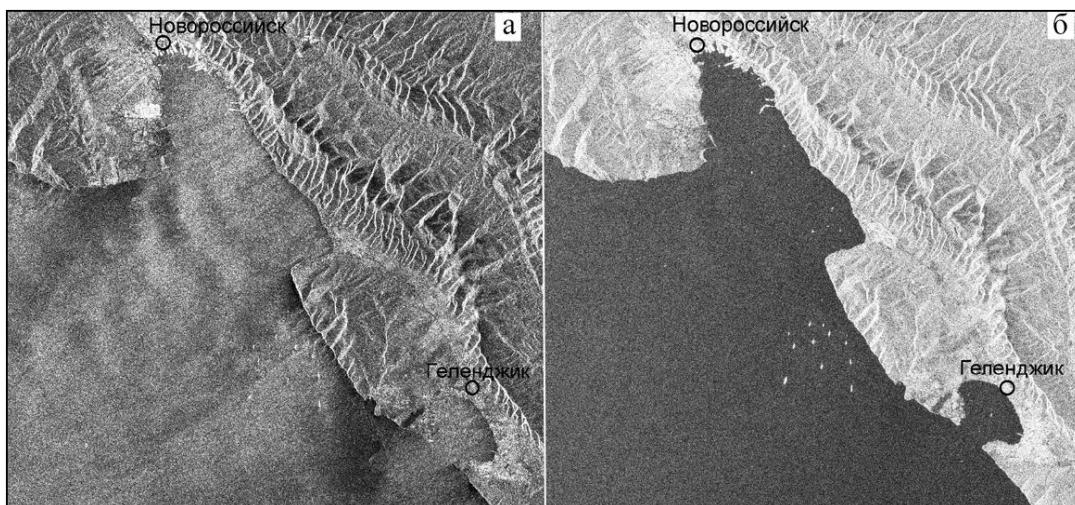


Рис. 3.25. Примеры изображений, полученных методом синтеза апертуры со спутника Envisat ASAR: поляризация типа VV (а), типа VG (б)

Разрешение при радиолокации зависит от размера антенны и длины волны испускаемого сигнала. Радиолокаторы, установленные на самолетах при расстоянии до земли порядка 10 км и длине волны 2 см, имеют разрешение около 15 м. Если подобную аппаратуру использовать на спутнике, то расстояние до Земли составит 500–1000 км и разрешение возрастет до 1000–2000 м.

Для повышения разрешения используют *метод синтеза апертуры*, т. е. искусственное увеличение размера антенны. В этом методе производится когерентная (с сохранением фазы) обработка рассеянных сигналов в течение фиксированного промежутка времени, за который спутник пролетает некоторое расстояние (рис. 3.26). Это расстояние играет роль длины синтезированной антенны. Например, при длине волны 5 см и дальности до Земли 1000 км разрешение составит 5 м. Подобные системы называют радиолокаторами с синтезированной апертурой (PCA).

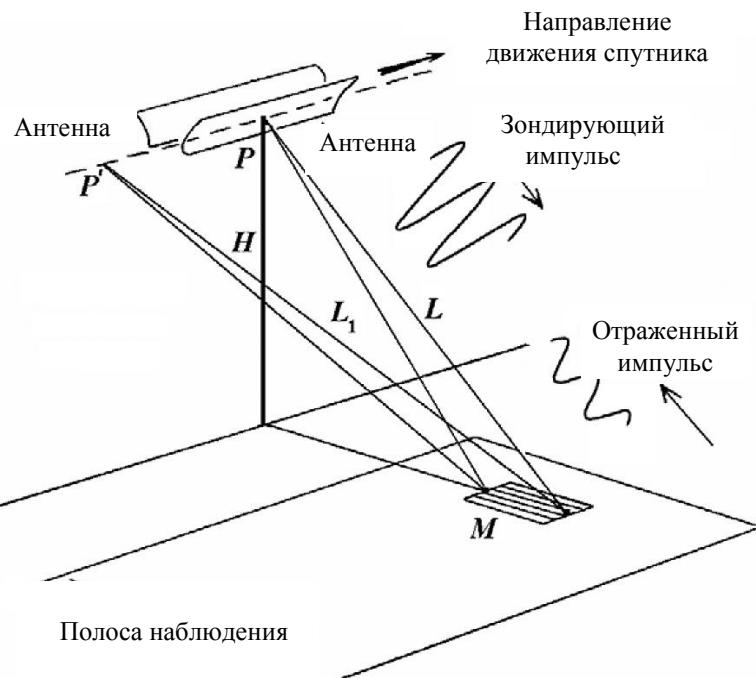


Рис. 3.26. Идея метода синтезированной апертуры

На рис. 3.27 приведены для сравнения изображения одной местности, зарегистрированные в разных спектральных диапазонах.

На левом изображении можно заметить спекл-шум, представляющий усиления яркости светлых объектов в виде светлых вертикальных крестов. Этот вид шума появляется только на радарных изображениях в результате случайной интерференции и наложения отраженных сигналов от разных объектов. Один из методов уменьшения этого шума – фильтрация, например медианная (см. разд. 7.5).

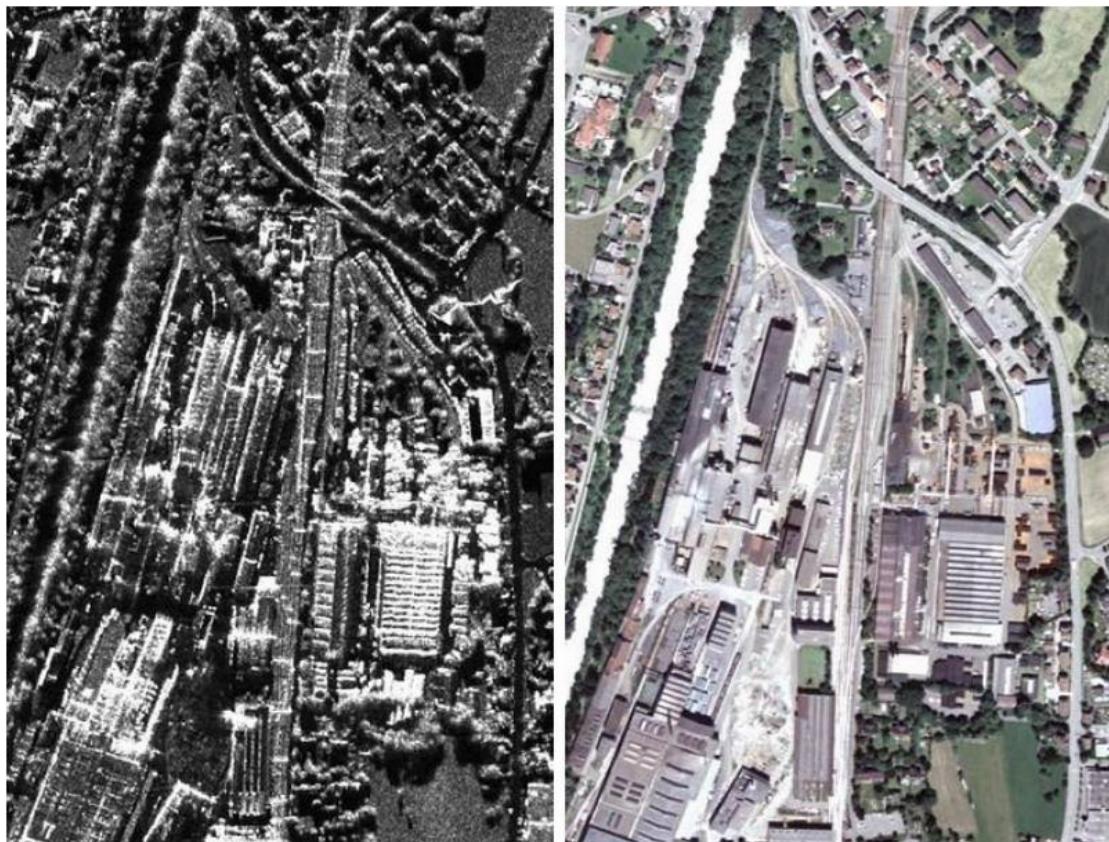


Рис. 3.27. Примеры изображений, полученных методом синтеза апертуры (слева)  
и в видимом диапазоне (справа)

При регистрации отраженных сигналов от гористой местности может фиксироваться ее искаженное изображение в силу разницы между пройденным путями посланного и отраженного сигнала (рис. 3.28 и 3.29).



Рис. 3.28. Искажение наклонной дальности меняет пропорции объектов  
(справа реальные пропорции)

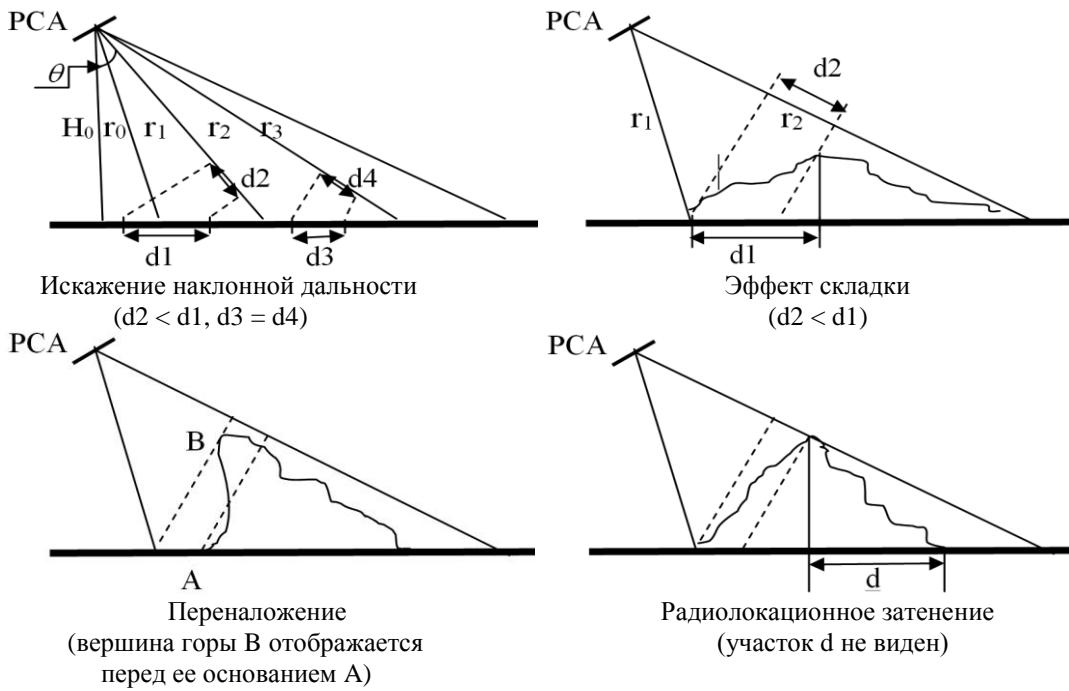


Рис. 3.29. Влияние геометрии съемки и рельефа на формирование радарных изображений

Радары способны измерять не только амплитуду, но и точный момент колебаний (фазу) отраженных волн. Системы, регистрирующие амплитуды и фазы сигналов, называются когерентными. Что дает знание фазы колебаний? Предположим, что со спутника, находящегося на орбите высотой 800 км посыпается к Земле радиосигнал с частотой 6 ГГц. Длина волны при этом составляет 5 см. Сигнал достигает поверхности Земли и отражается назад, проделав общий путь в 1600 км, что соответствует очень большому, но целому числу длин волн. Фаза колебаний при этом остается неизменной. Однако если расстояние до Земли превышает 800 км хотя бы на один сантиметр, то волна должна проделать путь на два сантиметра больше, что составляет 40 % от длины волны. В результате, когда фаза отраженной волны достигнет спутника, она будет отставать на 40 % периода колебаний, что можно легко зарегистрировать. Таким образом, измерение фазы дает возможность оценить расстояние до цели с сантиметровой или даже миллиметровой точностью.

В случае, если два последовательных радарных снимка получены при совершенно одинаковых положениях спутника, то не должно быть никакой разности фаз для любой пары соответствующих точек. Если сцена на Земле изменилась в промежутке между съемками хотя бы незначительно, то фазы некоторых пикселов будут отличаться. Таким образом, на основе двух радарных снимков можно получить интерферо-

граммой, которая позволяет отслеживать смещения земной поверхности в несколько миллиметров.

С учетом изложенного очень важно правильно подбирать тип поляризации и длину волны для конкретного проекта, чтобы улучшить характеристики результата ДЗЗ.

## Выводы по части I

Изображение – это двухмерный зарегистрированный сигнал, отраженный в определенном участке электромагнитного спектра. Исключением является рентгеновское излучение – регистрируется сигнал, прошедший через некоторое тело.

Изображение формируется путем регистрации сигнала. Сигналы разной длины регистрируются сенсорами, которые еще называют пикселями. Пиксели размещаются линейно или в виде прямоугольного массива, поэтому регистрирующие устройства называют ПЗС-линейками и ПЗС-матрицами.

*Цифровое изображение* – это дискретная функция двух пространственных переменных  $f(x, y)$ , определенная в конечной области плоскости  $XY$ , принимающая множество неотрицательных целых значений из диапазона  $[0, 2^n - 1]$ , где  $n$  – целое. То есть  $f(x, y) \geq 0$ ,  $0 \leq x \leq a$ ,  $0 \leq y \leq b$ , где  $f(x, y)$  – целое число, обозначающее номер диапазона некоторой шкалы, соответствующий яркости изображения в пикселе с координатами  $(x, y)$ ;  $a$  – ширина;  $b$  – высота изображения. Функция  $f$  представлена таблично в виде прямоугольной матрицы. Элементы матрицы тоже называются пикселями (от англ. pixel – picture element).

На логическом уровне представления бинарных, полутооновых и индексированных изображений задаются двухмерной матрицей, цветные и многоспектральные изображения – трехмерной матрицей. Для хранения и передачи изображения могут записываться в файлы с использованием разных форматов, часто подвергаясь сжатию (не путать алгоритм сжатия и формат, например JPEG). При этом данные располагаются в файле последовательно, т. е. фактически в виде одномерного массива.

После считывания и декомпрессии файла, содержащего изображение, восстанавливается матрица значений яркости. Если изображение бинарное или полутооновое, то матрица двухмерная. Если изображение цветное или многоспектральное, то она трехмерная, но каждый слой, соответствующий отражению в определенном диапазоне электромагнитного спектра, представлен как полутооновое изображение.

Такие матрицы представляют цифровые изображения, которые являются исходными данными для методов обработки, описанных в следующей части.

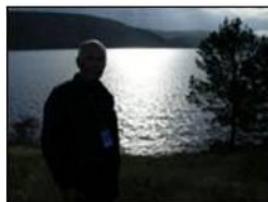
## ЧАСТЬ II. ОБРАБОТКА ЦИФРОВЫХ ИЗОБРАЖЕНИЙ

### Глава 4. Классификация методов цифровой обработки изображений

*Цифровая обработка изображений* – это любое изменение данных, представленных в виде цифровых изображений. Результатом обработки также является цифровое изображение (рис. 4.1). Цель обработки изображений есть преобразование изображений, ориентированное на *визуальное восприятие человеком* (в основном повышение качества, коррекция цвета и контраста, исправление мелких помех или искажение изображения) и их *автоматизированный анализ* (например, фильтрация, математическая морфология, сегментация на области определенных классов, выделение объектов и т. п.).



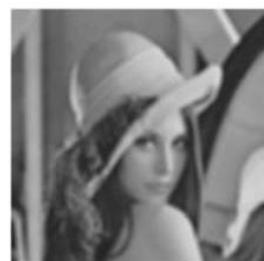
Темное или слабоконтрастное



Неправильные цвета



Шумное



Размытое



Неравномерно освещённое

Рис. 4.1. Примеры изображений, требующих корректировки

Среди задач обработки изображений можно выделить две основные: улучшение и восстановление изображений. Главная цель *улучшения* заключается в таком изменении изображения, чтобы результат оказался наиболее подходящим с точки зрения *конкретного* применения. Общей теории улучшения изображений не существует. Например, метод, оптимальный для улучшения рентгеновских изображений, как правило, не подходит для обработки цветных фотографий и наоборот.

Если изображение обрабатывается для визуальной интерпретации, то пользователь будет судить, насколько хорош примененный метод. Визуальное оценивание качества изображения – это субъективный процесс, а понятие хорошего изображения невозможно формализовать или оценить численно. Когда целью является обработка изображения для компьютерного анализа, задача оценивания проще. Например, при решении задачи распознавания символов лучшим будет тот метод обработки изображения, благодаря которому будут получены более точные результаты машинного распознавания.

При *восстановлении* изображения также выполняется повышение его качества, но известен тип преобразования, исказившего изображение (например, смазано изображение автомобиля на фотографии). Несмотря на пересечение областей применения методов обоих классов, улучшение изображений ориентировано на субъективное восприятие результата, в то время как восстановление может быть оценено более объективно. При восстановлении выполняется реконструкция изображения, которое было искажено в процессе регистрации (например, смазано), посредством использования априорной информации о явлении, вызвавшем ухудшение изображения. Поэтому методы восстановления основаны на моделировании процессов искажения, применения обратных процедур для воссоздания исходного изображения.

Таким образом, улучшение изображений выполняется как для визуального восприятия результата человеком, так и для автоматического анализа и распознавания изображений. В чем же разница? Человеческий глаз не отличит на полутоновом изображении пиксели, имеющие значения яркости 10, 14 и 18, а при автоматическом анализе придется, например, решать, образуют эти пиксели одну однородную область или нет.

Методы обработки изображений можно классифицировать разными способами. В частности, можно выделить группы методов в зависимости от их предназначения.

### *1. Исправление дефектов изображения, обусловленных:*

- шумом (случайными погрешностями цвета в отдельных пикселях изображения);
- недостаточной или избыточной яркостью;

- недостаточной контрастностью или узким динамическим диапазоном изображения;
- неправильным цветовым тоном;
- нерезкостью (расфокусировкой);
- искажениями из-за пыли, царапин на сканируемом документе;
- дисторсией (искривлением);
- необходимостью ретуши фотопортретов: устранение прыщей, царапин, шрамов, синяков, пор, веснушек или уменьшение их количества, разглаживание морщин; отбеливание зубов; замена цвета волос, глаз; коррекция формы фигуры и т. п.

*2. Структурное редактирование изображений:*

- а) кадрирование;
- б) создание панорам;
- в) устранение ненужных деталей изображения;
- г) фотомонтаж – создание из нескольких изображений нового изображения;
- д) вставка чертежей, надписей, символов, указателей и пр.;
- е) применение спецэффектов, фильтров, теней, текстур.

*3. Подготовка фотографий к публикации* (в печати, интернете): разные устройства вывода (монитор, принтер, офсетная печатная машина и т. п.) имеют разные возможности цветового охвата (не любой цвет можно воспроизвести). Выполняется коррекция цвета для печати на бумаге, определяется количество краски каждого цвета.

Для устранения искажений, внесенных в процессе регистрации изображений, используются методы геометрической и яркостной коррекции. При обработке космических снимков применяют атмосферную коррекцию изображений. Отдельное место занимают методы обработки и анализа изображений с помощью математической морфологии.

Методы обработки условно разделяют на три уровня:

- низкий (не используются дополнительные знания, модели и информация об объектах, представленных на изображении);
- средний (частично используются дополнительные сведения);
- высокий (широко используются дополнительные сведения и контекст изображения, т. е. информация об изображенных объектах).

Согласно вычислительной сложности, т. е. с учетом количества пикселов, участвующих в вычислении значения яркости одного пикселя нового изображения, методы делят на три класса:

*точечные* (в вычислениях используется значение одного пикселя исходного изображения);

*локальные* (в вычислениях используются значения нескольких соседних пикселов в заданной окрестности);

*глобальные* (при вычислении значения пикселя используются все пиксели исходного изображения).

Вычислительная сложность алгоритмов пропорциональна количеству пикселов, участвующих в вычислении нового значения яркости одного пикселя.

Напомним, что линейный функционал – это линейное отображение множества функций в множество чисел со свойствами

$$F[f + g] = F[f] + F[g], \quad F[cf] = cF[f], \quad (4.1)$$

где  $F$  – линейный функционал;  $f$  и  $g$  – функции из его области определения;  $c$  – константа.

Все методы обработки изображений можно разделить на линейные и нелинейные. Линейные методы допускают упрощение (и, следовательно, ускорение) вычислений, а также реализацию с помощью операции свертки (описана в разд. 7.2).

Обработка изображений может осуществляться рекурсивными и нерекурсивными методами. Рекурсивные методы используют результат обработки части соседних пикселов, нерекурсивные – не используют. В обработке изображений большинство методов являются нерекурсивными.

Многие методы обработки изображений адаптированы из разных областей науки: обработки многомерных сигналов, математики, физики, фотографии, полиграфии, картографии и др.

## Глава 5. Геометрические преобразования изображений

Геометрические преобразования заключаются в пространственном изменении расположения совокупности пикселов из одной двухмерной системы координат в другую. При этом изображение удобно рассматривать в виде пикселов, заданных в прямоугольной системе координат. При выполнении геометрических преобразований исходное множество пикселов с целочисленными координатами  $(x, y)$  преобразуется в новое множество (сетка раstra трансформируется пространственно) с координатами  $(x', y')$  и сохранением яркости в этих пикселях. Поскольку цифровое изображение представляет собой дискретное множество, кратко опишем основы геометрических преобразований для непрерывных множеств, а затем адаптируем их для растровых изображений.

При изучении геометрических преобразований плоских изображений (2D) предполагается, что имеем евклидово пространство с ортонормированной декартовой системой координат с взаимно ортогональными осями. Тогда каждому пикселу изображения ставится в соответствие пара

$(x, y)$  декартовых координат. Их можно интерпретировать как двухмерный вектор  $\mathbf{X}$ , представленный отрезком из точки  $(0, 0)$  до точки  $(x, y)$ .

Двухмерные преобразования на плоскости можно представить как движение точек, соответствующих множеству пикселов. Выделим линейные преобразования, описываемые матрицами, т. е. преобразования, при которых новые координаты точки линейно зависят от исходных координат этой точки следующим образом:  $\mathbf{x}' = \mathbf{T}\mathbf{x}$ .

Для удобства матричного описания и комбинирования геометрических преобразований в 2D-плоскости используется формализм «однородные координаты» [20]. Точки плоскости описываются трехмерными векторами вида  $(x, y, 1)$ , а преобразование – в матричном виде

$$\begin{bmatrix} \bar{x}' \\ \bar{y}' \\ w' \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix} \begin{bmatrix} \bar{x} \\ \bar{y} \\ w \end{bmatrix}.$$

К линейным преобразованиям можно отнести кусочно-линейные, при которых изображение разбивается на части, а затем к каждой из них применяются разные линейные преобразования.

Нелинейные преобразования – это все преобразования, которые не удовлетворяют свойствам линейности (4.1), например полиномиальные (выполняемые посредством полиномиальных функций), синусоидальные и т. п.

## 5.1. Аффинные преобразования

Основные типы аффинных преобразований и их матричные представления [20] показаны на рис. 5.1.

В целом любое аффинное преобразование на плоскости описывается уравнениями  $x' = ax + bx + c$ ,  $y' = dx + ey + f$  или представляется в матричном виде:

$$\mathbf{x}' = \mathbf{Ax} + \mathbf{c}, \quad \mathbf{x}' = \begin{bmatrix} x' \\ y' \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} a & b \\ d & e \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} c \\ f \end{bmatrix}.$$

В однородных координатах это преобразование можно описать одной матрицей  $T$  или следующим образом:

$$\begin{bmatrix} x' & y' & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \cdot \begin{bmatrix} t_{11} & t_{21} & 0 \\ t_{12} & t_{22} & 0 \\ x_0' & y_0' & 1 \end{bmatrix}.$$

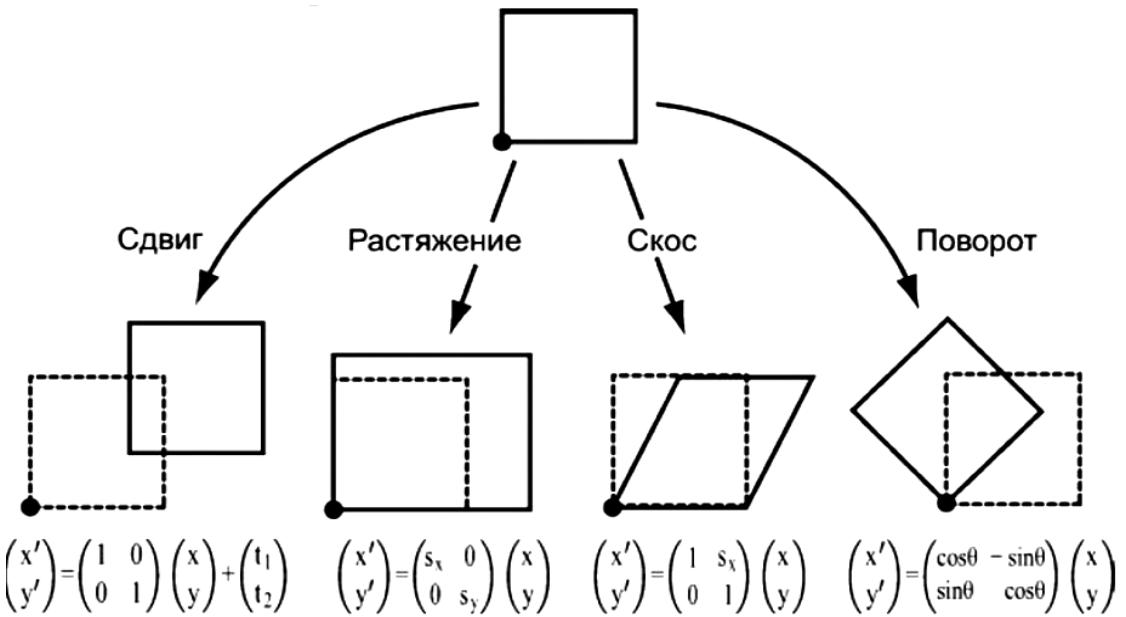


Рис. 5.1. Базовые преобразования аффинного типа

Опишем параметры базовых аффинных преобразований.

*Сдвиг изображения* задается соотношениями

$$x' = x + c, \quad y' = y + f.$$

Матрица сдвига имеет вид

$$\mathbf{T}_{cd} = \begin{bmatrix} 1 & 0 & c \\ 0 & 1 & f \\ 0 & 0 & 1 \end{bmatrix}.$$

*Отражение изображения относительно оси OY* задается формулами

$$x' = x, \quad y' = -y.$$

Матрица отражения имеет вид

$$\mathbf{T}_{om} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

*Масштабирование (растяжение) изображения* задается в виде

$$x' = \alpha x, \quad \alpha > 0,$$

$$y' = \beta y, \quad \beta > 0.$$

В однородных координатах матрица преобразования имеет вид

$$\mathbf{T}_{ma} = \begin{bmatrix} \alpha & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Необязательно  $\alpha = \beta$ . Если  $\alpha$  и  $\beta$  меньше единицы, изображение уменьшается, если больше единицы – увеличивается. При увеличении изображения количество пикселов возрастает, появляются новые с неопределенным значением яркости. Их значения вычисляются путем интерполяции, т. е. локальными методами, описанными ниже.

*Скос изображения* вдоль оси  $OX$  описывается формулами

$$x' = x + sy, \quad y' = y.$$

Матрица преобразования в однородных координатах имеет вид

$$\mathbf{T}_{ck} = \begin{bmatrix} 1 & s & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

*Поворот изображения* вокруг начала координат на угол  $\varphi$  описывается формулами

$$x' = x \cos \varphi - y \sin \varphi, \quad y' = x \sin \varphi + y \cos \varphi.$$

Матрица вращения в однородных координатах имеет вид

$$\mathbf{T}_{no} = \begin{bmatrix} \cos \varphi & -\sin \varphi & 0 \\ \sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Отметим, что при повороте на  $90^\circ \cos \varphi = 0, \sin \varphi = 1$ , поэтому

$$x' = -y, \quad y' = x.$$

При повороте изображения координаты пикселя могут принять нецелые значения, в таком случае требуется их округлить до целых. Это создает в повернутом изображении пиксели с неопределенной яркостью (рис. 5.2) [15].

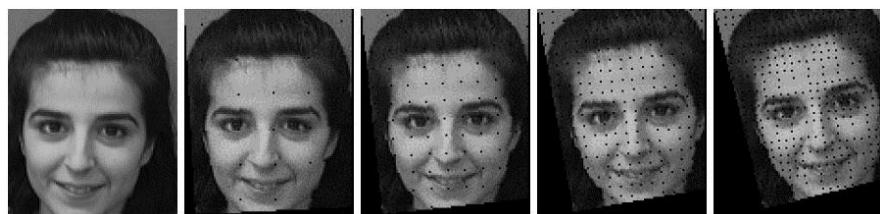


Рис. 5.2. Изображение повернуто на 3, 6, 10 и  $14^\circ$  вокруг верхнего левого угла  
(в черных пикселях яркость не определена)

Пикселам с неопределенной яркостью можно присвоить значение яркости ближайшего пикселя. Альтернативой является поворот посредством выполнения трех операций скоса вдоль оси  $OX$ , затем  $OY$  и еще раз вдоль  $OX$ . Матрица вращения описывается в виде произведения трех матриц скоса

$$T(\alpha) = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} = \begin{bmatrix} 1 & -\operatorname{tg} \alpha / 2 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ \sin \alpha & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & -\operatorname{tg} \alpha / 2 \\ 0 & 1 \end{bmatrix}.$$

После каждой операции скоса выполняется округление координат до целых значений.

Произвольное аффинное преобразование можно записать в виде композиции последовательно выполняемых базовых преобразований. Суперпозиция аффинных преобразований также является аффинным преобразованием. Преимущество суперпозиции: сложное преобразование изображения можно описать шестью параметрами, вычисленными один раз, и применить к координатам всех пикселов.

В непрерывной геометрии любое аффинное преобразование имеет обратное преобразование, которое также является аффинным. Произведение прямого и обратного дает единичное преобразование, оставляющее все точки на месте.

Перечислим некоторые свойства аффинных преобразований:

- прямая линия переходит в прямую;
- параллельные прямые переходят в параллельные;
- сохраняются отношения длин отрезков, лежащих на одной прямой или параллельных прямых;
- сохраняются отношения площадей фигур.

Множество аффинных преобразований образуют аффинную группу. Отметим, что аффинная группа является подгруппой проективной группы преобразований, а евклидова – частным случаем аффинной. Эти группы формируют следующую иерархию:

евклидово преобразование  $\subset$  аффинное  $\subset$  проективное преобразование.

## 5.2. Проективные преобразования

Общими линейными преобразованиями  $T$  в однородных координатах описывается группа проективных преобразований

$$\begin{bmatrix} \bar{x}' \\ \bar{y}' \\ 1 \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}. \quad (5.1)$$

При представлении в обычных координатах соотношение (5.1) будет иметь нелинейный вид:

$$x' = \frac{t_{11}x + t_{12}y + t_{13}}{t_{31}x + t_{32}y + t_{33}}, \quad y' = \frac{t_{21}x + t_{22}y + t_{23}}{t_{31}x + t_{32}y + t_{33}}. \quad (5.2)$$

Проективные преобразования в общем случае не сохраняют параллельности линий. Свойством, сохраняющимся при проективном преобразовании, является коллинеарность точек: три точки, лежащие на одной прямой (коллинеарные), после преобразования остаются на одной прямой.

Проективное преобразование связано с отображением трехмерной визуальной информации на двухмерную плоскость. С математической точки зрения удобно рассматривать мир как трехмерное проективное пространство  $P^3$ , а плоскость изображения – включенной в проективное пространство размерности два –  $P^2$ . Точки на трехмерной сцене и на изображении представляются в проективных пространствах как векторы в однородных координатах. Проективное преобразование из  $P^3$  в  $P^2$  (перспективная проекция), отображающее евклидову точку сцены  $\mathbf{p} = (X, Y, Z)^t$  в точку изображения  $\mathbf{x} = (x, y)^t$  и выраженное в однородных координатах, задается уравнениями (5.1) или (5.2).

На рис. 5.3, б, в показаны изображения, полученные проективными преобразованиями соответственно:

$$x' = \frac{1,1x + 0,35y}{0,00075x + 0,0005y + 1}, \quad y' = \frac{0,2x + 1,1y}{0,00075x + 0,0005y + 1}$$

и

$$x' = \frac{1,1x + 0,2y}{0,00075x + 0,00005y + 1}, \quad y' = \frac{0,1x + 0,9y}{0,00075x + 0,00005y + 1}.$$

Однородные координаты векторов проективного пространства  $(\bar{X}, \bar{Y}, \bar{Z}, W)^t$  и проективной плоскости  $(\bar{x}, \bar{y}, w)^t$  соотносятся с неоднородными (евклидовыми) координатами векторов  $\mathbf{p}$  и  $\mathbf{x}$  следующим образом:

$$\mathbf{p} = (\bar{X}/W, \bar{Y}/W, \bar{Z}/W)^t \text{ и } \mathbf{x} = (\bar{x}/w, \bar{y}/w)^t.$$

Изображение объектов на снимке, сформированном регистрирующей камерой, связано с проектированием при помощи пучка прямых, поскольку каждая 2D-точка является проекцией множества 3D-точек вдоль некоторого луча проектирования в плоскость снимка.

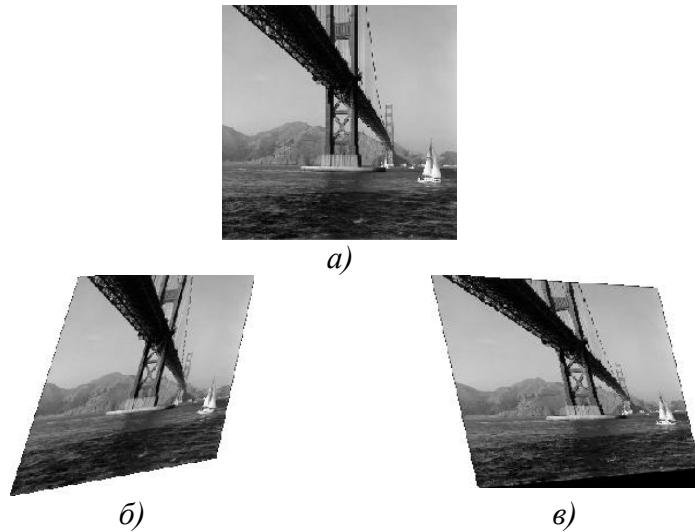


Рис. 5.3. Исходное изображение (а); два варианта его проективного преобразования (б, в)

При рассмотрении геометрических преобразований используется идеальная модель камеры. В действительности формирование изображений сопровождается различными нелинейными искажениями (типа оптической дисторсии линзы, описанных в подразд. 5.5.2). Приведение снимков к реальному представлению можно выполнить нелинейной функцией геометрического преобразования. Для этого обычно используют полиномиальную аппроксимирующую функцию преобразования

$$x' = a_0 + a_1x + a_2y + a_3x^2 + a_4xy + a_5y^2 + \dots + a_p y^m;$$

$$y' = b_0 + b_1x + b_2y + b_3x^2 + b_4xy + b_5y^2 + \dots + b_p y^m,$$

где  $(x, y)$  – координаты точек идеального изображения;  $(x', y')$  – соответствующие им координаты на его проекции.

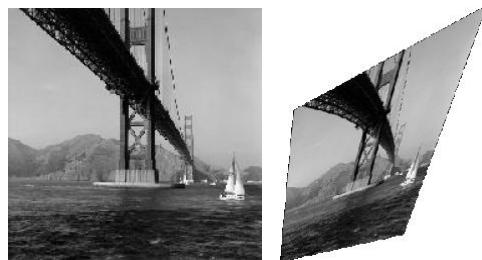


Рис. 5.4. Пример билинейного геометрического преобразования снимка

Полиномиальная аппроксимирующая функция преобразования для снимка на рис. 5.4 имеет вид

$$x' = 0,1x + 0,9y + 0,002xy, \quad y' = 0,2x + 1,1y + 0,0022xy. \quad (5.3)$$

### **5.3. Уточнение координат и значений яркости пикселов после преобразования**

После поворота или увеличения изображения часто возникает задача уточнения новых координат пикселов (рис. 5.5) в целочисленной системе координат. Она решается путем округления координат пикселов до целых значений и двухмерной интерполяции значений яркости в неопределенных пикселях. Наиболее простой является интерполяция методом ближайшего соседа, где значения координат округляются до ближайших точек дискретного растра.

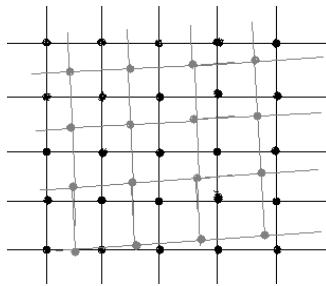


Рис. 5.5. При повороте координаты пикселов (выделены серым тоном) не совпадают с дискретной целочисленной системой координат

Возможны два варианта определения яркости пикселов при геометрических преобразованиях:

*прямой* – для каждого пикселя вычисляются новые координаты, они округляются до ближайших целых значений, затем неопределенным пикселям нового изображения присваивается яркость ближайшего после геометрического преобразования пикселя с неокругленными координатами;

*обратный* – для каждого пикселя преобразованного изображения координаты подвергаются обратному преобразованию в систему координат исходного изображения, координаты округляются до ближайших целых значений и берется яркость пикселя с такими координатами.

### **5.4. Геометрическая коррекция изображений**

#### **5.4.1. Построение мозаики из изображений**

Как можно использовать геометрические преобразования, рассмотрим на примере построения мозаики из двух изображений. Мозаика, или сшивка, – это объединение двух или более изображений в единое изображение. Пусть два изображения получены путем сканирования по частям одного большого рисунка (рис. 5.6). Обязательное условие: на обо-

их изображениях частично присутствуют одни и те же объекты, т. е. изображения имеют области перекрытия.

Системы координат двух изображений будут различаться из-за разного ракурса съемки, вращения камеры и движения самого фотографируемого объекта. Для изображений, представленных на рис. 5.6, освещение, расстояние до камеры, оптическая система одинаковые, поэтому основной задачей обработки таких изображений является приведение их в общую систему координат. Если изображения регистрировались в разное время, необходимо учесть изменения яркости, возникающие из-за изменения освещения, сезонные и суточные изменения, использование другой оптической системы.

Рассмотрим задачу нахождения такого пространственного преобразования, которое позволяет определить пиксели обоих изображений, представленных на рис. 5.6, в единой системе координат таким образом, чтобы точки, соответствующие одинаковым объектам на двух изображениях, совпали. В качестве общей системы координат можно использовать систему левого изображения, тогда требуется найти преобразование координат всех пикселов второго изображения  $(x, y)$  в общую систему координат  $(x', y')$ . Для упрощения задачи считаем, что в процессе регистрации не произошло искривления прямых линий, а лишь аффинные трансформации (поворот, сдвиг). Аффинные преобразования являются подмножеством полиномиальных преобразований первого порядка и описываются двумя уравнениями:

$$\begin{aligned} x' &= a_0 + a_1x + a_2y, \\ y' &= b_0 + b_1x + b_2y. \end{aligned} \quad (5.4)$$

На обоих изображениях (рис. 5.7) можно выбрать интерактивно или найти с помощью специальных методов детекции пиксели, соответствующие одинаковым объектам. Обозначим через  $(x_i, y_i)$  координаты таких пикселов на правом изображении в системе координат этого изображения,  $(x'_i, y'_i)$  – координаты в системе координат левого изображения. Эти координаты известны (точки определены), коэффициенты  $a_0–a_2$  и  $b_0–b_2$  неизвестны. Для решения задачи вычисления шести коэффициентов  $a_0–a_2$ ,  $b_0–b_2$  аффинного преобразования минимальное количество общих точек равно  $n=3$ . Если задать больше пикселов, то необходимо минимизировать ошибки их задания, которые неизбежны в силу неточного задания и округления координат до целых значений.

Таким образом, имеем шесть неизвестных, для их поиска требуется три пары пикселов с известными координатами до и после трансформации  $(x_1, y_1); (x_2, y_2); (x_3, y_3)$  и  $(x'_1, y'_1); (x'_2, y'_2); (x'_3, y'_3)$ . Подста-

вим координаты  $(x, y)$  в уравнения (5.4), получим три пары уравнений с неизвестными  $a_i$  и  $b_i$ :

$$\begin{aligned} x'_1 &= a_0 + a_1 x_1 + a_2 y_1, & y'_1 &= b_0 + b_1 x_1 + b_2 y_1, \\ x'_2 &= a_0 + a_1 x_2 + a_2 y_2, & y'_2 &= b_0 + b_1 x_2 + b_2 y_2, \\ x'_3 &= a_0 + a_1 x_3 + a_2 y_3, & y'_3 &= b_0 + b_1 x_3 + b_2 y_3. \end{aligned}$$



Рис. 5.6. Пример сканирования документа по частям



Рис. 5.7. Пример выбора пар соответствующих пикселов

В матричной форме это выглядит следующим образом:

$$\begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}, \quad \begin{bmatrix} y'_1 \\ y'_2 \\ y'_3 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix}.$$

Для вычисления коэффициентов  $a_i$  и  $b_i$  каждая часть матричного уравнения должна быть умножена на обратную матрицу. Например, для  $a_i$

$$\begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}^{-1} \begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix} \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}^{-1} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}.$$

В матричной форме коэффициенты  $a_i$  и  $b_i$  вычисляются по формулам

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}^{-1} \begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \end{bmatrix}, \quad \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}^{-1} \begin{bmatrix} y'_1 \\ y'_2 \\ y'_3 \end{bmatrix}.$$

Полученный набор коэффициентов подставляется в исходные уравнения (5.4), и новые координаты вычисляются для каждого пикселя второго изображения. Осталось определить, какую часть второго изображения необходимо присоединить к первому.

К полученному изображению можно присоединять справа и снизу другие изображения. Результирующее изображение представлено на рис. 5.8.



Рис. 5.8. Пример мозаики, составленной из двух изображений



Рис. 5.9. Пример сшивки фотографий с учетом геометрических искажений

При съемке больших объектов можно их фотографировать по частям, а затем выполнить сшивку полученных изображений. Пример обработки панорамных фотографий приведен на рис. 5.9. Таким же образом была создана панорама Праги из 2600 снимков. В результате получено изображение размером 260 000x130 000 пикселов. Если его распечатать с разрешением 150 dpi, получится фотография длиной около 40 м.

#### 5.4.2. Коррекция проективных искажений

Проективное преобразование описывается уравнениями

$$x' = \frac{a_{01}x + a_{10}y + a_{00}}{c_{01}x + c_{10}y + 1}, \quad y' = \frac{b_{01}x + b_{10}y + b_{00}}{c_{01}x + c_{10}y + 1},$$

где  $x$  и  $y$  – точки идеального изображения;  $x'$  и  $y'$  – точки проективно искаженного изображения.

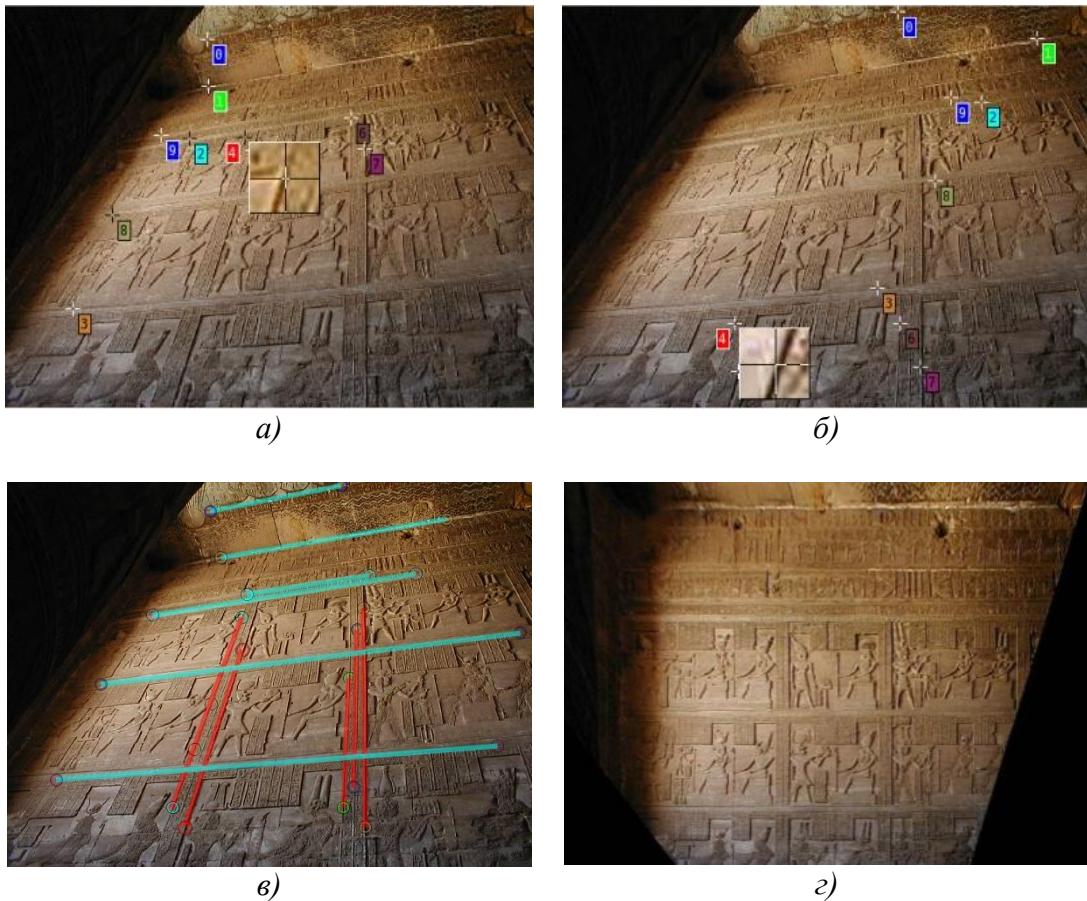


Рис. 5.10. Пример перспективной коррекции изображения: *а*) заданы левые концы горизонтальных отрезков и верхние вертикальных; *б*) заданы правые концы горизонтальных отрезков и нижние вертикальных (концы одного отрезка помечены одинаковыми цифрами); *в*) указанные отрезки отображены; *г*) изображение откорректировано

При коррекции перспективных искажений исходное изображение одно (рис. 5.10). Интерактивно задаются концы отрезков, которые должны быть вертикальными и горизонтальными на откорректированном изображении. Для определения восьми неизвестных коэффициентов минимально необходимо задать восемь точек. Посредством решения системы линейных уравнений вычисляются неизвестные параметры преобразования  $a_{ij}$ ,  $b_{ij}$  и  $c_{ij}$ . Все пиксели исходного изображения трансформируются, и отмеченные отрезки становятся вертикальными и горизонтальными.

## 5.5. Нелинейная геометрическая коррекция

### 5.5.1. Общий подход к вычислению параметров преобразования

Не всегда задачу геометрической коррекции можно решить с помощью линейных преобразований. Нелинейные преобразования (второго и более порядка) помогают корректировать нелинейные искажения изображения, возникшие в процессе его регистрации.



Рис. 5.11. Примеры нелинейных искажений

Преобразования второго порядка могут быть использованы для преобразования данных из географической системы координат (широта, долгота) в прямоугольную, больших областей (для учета кривизны Земли), для точной привязки искаженных по той или иной причине (из-за искажений линз камеры, плохо отсканированных материалов) данных и т. д.

Способ решения задачи трансформации аналогичен предыдущему и сводится к нахождению коэффициентов системы уравнений. В случае полиномиального преобразования второго порядка степени системы уравнений выглядит следующим образом:

$$\begin{aligned} x' &= a_0 + a_1x + a_2y + a_3x^2 + a_4xy + a_5y^2, \\ y' &= b_0 + b_1x + b_2y + b_3x^2 + b_4xy + b_5y^2, \end{aligned} \quad (5.5)$$

где  $x$ ,  $y$  – координаты точек в одной системе координат (известны);  $x'$ ,  $y'$  – координаты этих точек в другой системе координат (известны);  $a_0-a_5$ ,  $b_0-b_5$  – коэффициенты преобразования (неизвестны).

В данном случае необходимо знать координаты хотя бы шести пар соответствующих точек до и после трансформации:  $(x_1, y_1); (x_2, y_2); (x_3, y_3); (x_4, y_4); (x_5, y_5); (x_6, y_6)$  и  $(x'_1, y'_1); (x'_2, y'_2); (x'_3, y'_3); (x'_4, y'_4); (x'_5, y'_5); (x'_6, y'_6)$ .

Подставим их в (5.5), перегруппируем и вычислим 12 неизвестных  $a_i$  и  $b_i$ :

$$\begin{aligned} x'_1 &= a_0 + a_1 x_1 + a_2 y_1 + a_3 x_1^2 + a_4 x_1 y_1 + a_5 y_1^2, \\ &\dots \\ x'_6 &= a_0 + a_1 x_6 + a_2 y_6 + a_3 x_6^2 + a_4 x_6 y_6 + a_5 y_6^2, \\ y'_1 &= b_0 + b_1 x_1 + b_2 y_1 + b_3 x_1^2 + b_4 x_1 y_1 + b_5 y_1^2, \\ &\dots \\ y'_6 &= b_0 + b_1 x_6 + b_2 y_6 + b_3 x_6^2 + b_4 x_6 y_6 + b_5 y_6^2. \end{aligned}$$

Запишем эти уравнения для поиска  $a_i$  в матричной форме:

$$\begin{bmatrix} x'_1 \\ \dots \\ \dots \\ \dots \\ \dots \\ x'_6 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 & x_1^2 & x_1 y_1 & y_1^2 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & x_6 & y_6 & x_6^2 & x_6 y_6 & y_6^2 \end{bmatrix} \begin{bmatrix} a_1 \\ \dots \\ \dots \\ \dots \\ \dots \\ a_6 \end{bmatrix}$$

В матричной форме коэффициенты  $a_0-a_5, b_0-b_5$  вычисляются по формулам

$$\begin{bmatrix} a_1 \\ \dots \\ \dots \\ \dots \\ \dots \\ a_6 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 & x_1^2 & x_1 y_1 & y_1^2 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & x_6 & y_6 & x_6^2 & x_6 y_6 & y_6^2 \end{bmatrix}^{-1} \begin{bmatrix} x'_1 \\ \dots \\ \dots \\ \dots \\ \dots \\ x'_6 \end{bmatrix},$$

$$\begin{bmatrix} b_1 \\ \dots \\ \dots \\ \dots \\ \dots \\ b_6 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 & x_1^2 & x_1 y_1 & y_1^2 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & x_6 & y_6 & x_6^2 & x_6 y_6 & y_6^2 \end{bmatrix}^{-1} \begin{bmatrix} y'_1 \\ \dots \\ \dots \\ \dots \\ \dots \\ y'_6 \end{bmatrix}.$$

Полученный набор коэффициентов подставляется в исходные уравнения (5.5) для пересчета координат всех пикселов. Сначала нужно

определить размер нового изображения, найдя минимум и максимум координат углов изображения в новой системе координат. Новые координаты вычисляются для каждого пикселя исходного изображения.

Преобразования более высоких порядков могут быть использованы для корректировки более сложных типов искажений, например при сшивке изображений горной местности, снятых с самолета.

Для вычисления параметров преобразований высших порядков требуется большее количество контрольных точек. Минимальное количество пар соответствующих точек для преобразования  $t$ -го порядка вычисляется по формуле

$$\frac{((t+1)(t+2))}{2}.$$

### 5.5.2. Коррекция дисторсии

При фотографировании на изображении может возникнуть искажение, которое называется дисторсией. *Дисторсия* – это оптическое искажение, выражющееся в искривлении прямых линий. Световые лучи проходят через центр линзы и сходятся в точке, расположенной дальше от линзы, чем лучи, которые проходят через ее края (рис. 5.12).

Дисторсия не нарушает резкость и яркость изображения, но вносит искажение в его форму. Прямые линии изображаются кривыми, кроме тех, которые лежат в одной плоскости с оптической осью. Например, изображение квадрата, центр которого пересекает оптическая ось, имеет вид подушки (подушкообразная дисторсия) при положительной дисторсии у широкоугольных объективов при съемке на максимальном фокусном расстоянии и вид бочки (бочкообразная дисторсия) при отрицательной дисторсии при съемке телеобъективом на минимальном фокусном расстоянии (рис. 5.13). Это наиболее распространенные варианты дисторсии оптических систем фотоаппаратов с осевой симметрией.

Дисторсия объектива может быть скорректирована оптически за счет особой конструкции объектива, использования асферических линз либо программно. Как правило, такая дисторсия имеет осевую симметрию, это значит, что горизонтальная и вертикальная оси, проходящие через центр изображения, не искажены.

Если искривление присутствует в цифровом изображении, его можно исправить программно методом, описанным ниже.

Будем считать, что центр системы координат находится в центре изображения. Для осесимметричных оптических систем дисторсия может быть только нечетных порядков: третьего, пятого, седьмого и т. д. Пусть  $\mathbf{r} = (x, y)$  – вектор, задающий две координаты в плоскости, расположенной вблизи центра изображения.

ложенной перпендикулярно оптической оси (координата  $Z$  вдоль оси сюда не входит). Для идеального изображения все лучи, вышедшие из точки с координатами  $\mathbf{r}$  и прошедшие через оптическую систему, попадут в точку изображения с координатами  $\mathbf{R}$ , которые определяются по формуле  $\mathbf{R} = b_0\mathbf{r}$ , где  $b_0$  – коэффициент линейного увеличения. Для одной и той же оптической системы дисторсия зависит от расстояния до объекта, а следовательно, и от коэффициента  $b_0$ . У длиннофокусных объективов дисторсия меньше, чем у нормальных, у широкоугольных – больше. Дисторсия также незначительно зависит от длины отраженной волны.

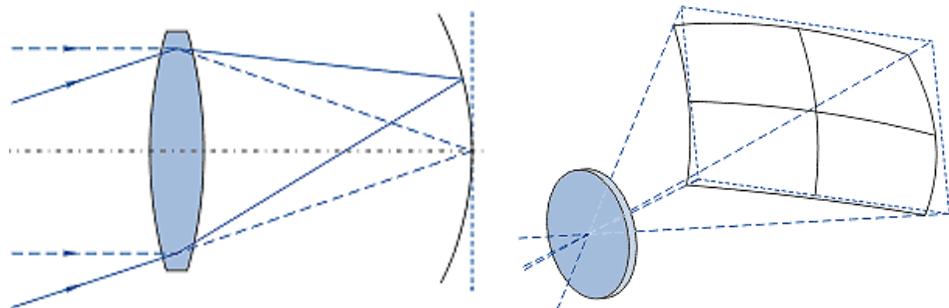


Рис. 5.12. Оптические искажения при регистрации изображения порождают дисторсию

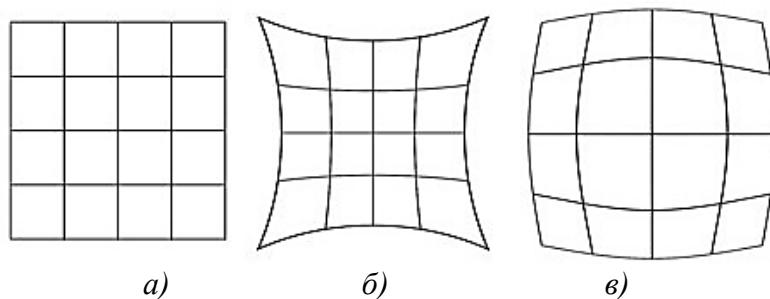


Рис. 5.13. Основные варианты дисторсии фотообъективов: *a)* дисторсия отсутствует; *б)* подушкообразная дисторсия; *в)* бочкообразная

Если присутствует дисторсия третьего порядка, тогда лучи, вышедшие из точки с координатами  $\mathbf{r}$ , попадут в точку изображения с координатами  $\mathbf{R} = b_0\mathbf{r} + F_3r^2 \cdot \mathbf{r}$ , где  $r$  – длина вектора  $\mathbf{r}$ ;  $F_3$  – коэффициент дисторсии третьего порядка, который вносит наибольший вклад в искажение формы изображения. Если  $F_3$  имеет тот же знак, что и  $b_0$ , возникает подушкообразное искажение, в противном случае – бочкообразное.

Для вычисления параметров корректирующего преобразования используют изображение регулярной сетки и ее искривленные изображения (рис. 5.14). На них выбираются пары соответствующих точек; вы-

числяются векторы, соединяющие эти точки с началом координат; полученные параметры подставляются в уравнение дисторсии и решается система линейных уравнений для определения неизвестных коэффициентов  $b_0$  и  $F_3$ .

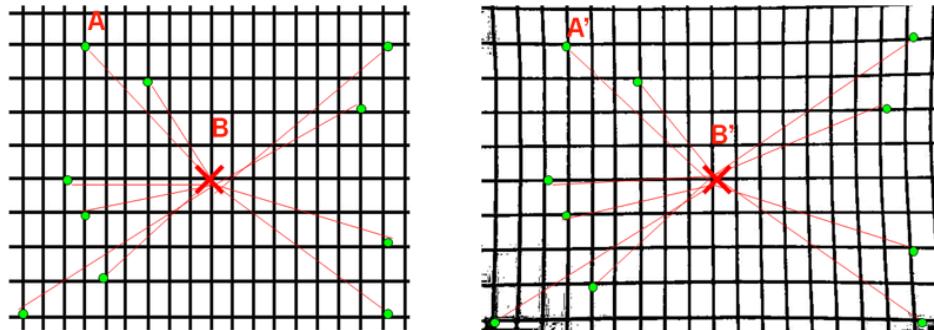


Рис. 5.14. Изображение регулярной сетки (слева) и ее снимок (справа), сделанный мобильным телефоном Nokia 6103 с парами соответствующих точек А и В

Примеры исправления дисторсии реальных изображений приведены на рис. 5.15.



Рис. 5.15. Примеры двух видов дисторсии и исправленное изображение

# Глава 6. Преобразования изображений точечного типа

## 6.1. Логические операции над изображениями

Яркость бинарных изображений задается числами 0 и 1. Эти изображения можно интерпретировать как множества черных пикселов на белом фоне. К ним можно применять операции теории множеств AND, OR, NOT, XOR (рис. 6.1). Пусть  $L_1$  и  $L_2$  – два бинарных изображения равных размеров. Операции применяются к пикселям с одинаковыми координатами. В табл. 6.1 показано, как изменяются значения яркости при применении этих логических операций. Операция NOT создает негативное изображение, ее можно описать функцией  $L = 1 - L_1$ .

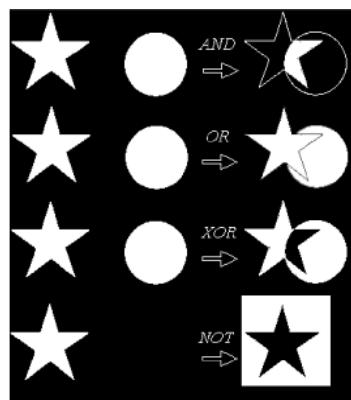


Рис. 6.1. Примеры логических операций над бинарными изображениями

Таблица 6.1

Результаты логических операций

$L_1$	$L_2$	$L_1 \text{AND } L_2$	$L_1 \text{OR } L_2$	$\text{NOT } L_1$	$L_1 \text{XOR } L_2$
0	0	0	0	1	0
0	1	0	1	1	1
1	0	0	1	0	1
1	1	1	1	0	0

## 6.2. Линейные преобразования яркости полутоновых изображений

Яркостными преобразованиями изображения  $I(i, j)$  называются преобразования двухмерных функций яркости, описываемые формулой  $I'(i, j) = f(I(i, j))$ . Преобразования яркости изображений относятся к точечному типу, если значение яркости пикселя после преобразования зависит от яркости одной точки (пикселя) исходного изображения и не зависит от ее местоположения. Пусть  $x = I(i, j)$  – функция яркости исход-

ного изображения,  $y = J(i, j)$  – функция яркости преобразованного изображения. Поэлементная обработка означает, что изменение яркости можно описать функцией  $y = f(x)$  независимо от координат пикселя.

**Препарирование изображений** [20]. С помощью яркостного среза изображения можно выделить области изображения с яркостью из определенного интервала. При этом остальным областям можно присвоить черный цвет и получить бинарное изображение (рис. 6.2, *a*) или оставить неизменными (рис. 6.2, *б*). Перемещая выделенный интервал по шкале яркости и изменяя его ширину, можно детально исследовать содержание изображения. При этом пиксели, имеющие яркость, не попавшую в этот интервал, отображаются черными, с яркостью 0 (рис. 6.2, *a*), либо линейно контрастируются (рис. 6.2, *б*). Такие преобразования используются, если интересующие интервалы яркости можно определить (вычислить или задать интерактивно).

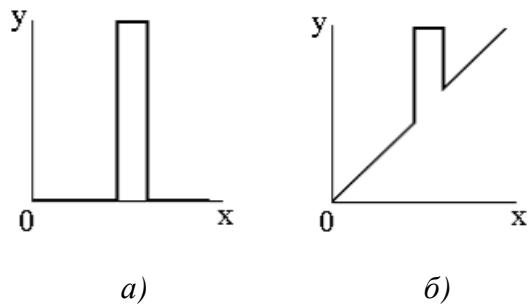


Рис. 6.2. Два варианта графического задания функции яркостного среза

Аналитически функция среза для первого варианта (рис. 6.2, *а*) задается следующим образом (вместо значения яркости 1 используется 255 для отображения на экране):

$$y = \begin{cases} 255, & \text{если } T_1 < x < T_2, \\ 0, & \text{иначе.} \end{cases}$$

*Пилообразное контрастирование* – это линейное контрастирование, применяемое к одному или нескольким диапазонам яркости (рис. 6.3). Оно также позволяет повысить детальность изображения в выбранном интервале яркости.

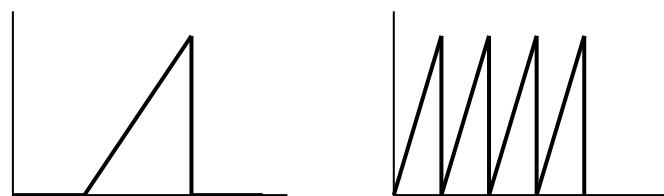


Рис. 6.3. Пилообразные варианты яркостного среза

**Бинаризация изображений.** Простейшим методом препарирования изображений является *бинаризация*. Она преобразует полутонаовое изображение в бинарное (черно-белое). Преобразование имеет единственный параметр – порог, относительно которого яркость пикселов меняется на черную или белую. Функция бинаризации с глобальным (т. е. единым для всех пикселов) порогом имеет вид

$$y = \begin{cases} 255, & \text{если } x > T, \\ 0, & \text{иначе,} \end{cases}$$

а ее графическое представление показано на рис. 6.4. Это преобразование используют при выделении темных объектов на светлом фоне или наоборот (рис. 6.5).

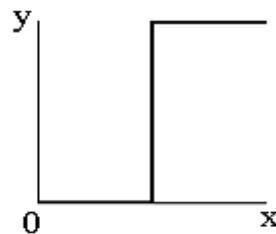


Рис. 6.4. Функция бинаризации с глобальным порогом



Рис. 6.5. Пример бинаризации изображения

Опишем три варианта вычисления порога. Наиболее простой из них заключается в нахождении минимального  $g_{\min}$  и максимального  $g_{\max}$  значений яркости изображения и вычисления среднего значения между ними:

$$t = (g_{\max} - g_{\min})/2.$$

Метод Отсу использует гистограмму распределения значений яркости изображения и ищет оптимальный порог  $t$ , разделяющий два класса пикселов (темные и светлые) и уменьшающий внутриклассовую дисперсию, которая определяется как взвешенная сумма дисперсий двух классов:

$$\sigma_{\omega}^2(t) = \omega_1(t)\sigma_1^2(t) + \omega_2(t)\sigma_2^2(t),$$

где веса  $\omega_i$  – это вероятности двух классов, разделенных порогом  $t$ ;  $\sigma_i^2$  – дисперсия этих классов. Минимизация дисперсии *внутри* класса равно-

сильна максимизации дисперсии между классами, которая выражается вероятностью  $\omega_i$  среднего арифметического  $i$ -го класса  $\mu_i$ :

$$\sigma_b^2(t) = \sigma^2 - \sigma_\omega^2(t) = \omega_1(t)\omega_2(t)[\mu_1(t) - \mu_2(t)]^2,$$

где  $\sigma^2$  – дисперсия всего изображения.

**Алгоритм вычисления порога бинаризации методом Отсу**

**Шаг 1.** Вычислить гистограмму яркостей и вероятность для каждого уровня интенсивности  $p_i = n_i/N$ , где  $N$  – число пикселов в изображении;  $n_i$  – число пикселов с уровнем яркости  $i$ . Диапазон яркостей делится на два класса порогом  $k$ , где  $k$  – целое значение от 0 до  $L$  (обычно  $L = 255$ ).

**Шаг 2.** Вычислить начальные значения для  $\omega_i(0)$  и  $\mu_i(0)$ :

$$\begin{aligned}\omega_0(k) &= \sum_{i=1}^k p_i, & \omega_1(k) &= \sum_{i=k+1}^L p_i = 1 - \omega_0(k), \\ \mu_0(k) &= \sum_{i=1}^k \frac{ip_i}{\omega_0}, & \mu_1(k) &= \sum_{i=k+1}^L \frac{ip_i}{\omega_1}.\end{aligned}$$

**Шаг 3.** Для каждого значения порога от  $t = 1$  до максимальной яркости (равной 255):

- 1) обновить  $\omega_i$  и  $\mu_i$ ;
- 2) вычислить  $\sigma_b^2(t)$ ;
- 3) если  $\sigma_b^2(t)$  больше, чем имеющееся, запомнить  $\sigma_b^2$  и значение порога  $t$ .

**Шаг 4.** Оптимальный порог  $t$  соответствует максимуму  $\sigma_b^2(t)$ .

**Алгоритм вычисления порога на основе градиента яркости**

**Шаг 1.** Вычислить модуль градиента яркости для каждого пикселя  $(m, n)$  изображения  $f$ :

$$G(m, n) = \max \{ |G_m(m, n)|, |G_n(m, n)| \},$$

$$\begin{aligned}G_m(m, n) &= f(m+1, n) - f(m-1, n), \\ \text{где } G_n(m, n) &= f(m, n+1) - f(m, n-1).\end{aligned}$$

**Шаг 2.** Вычислить значение порога  $t$ :

$$t = \frac{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) G(m, n)}{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} G(m, n)}.$$

**Линейное растяжение контраста изображений** предназначено для улучшения согласования динамического диапазона изображения и экрана, на котором выполняется визуализация. Если для цифрового представления яркости каждого пикселя изображения отводится 1 байт (8 бит), то входное или выходное изображение может принимать одно из 256 значений в диапазоне от 0 до 255. Пусть минимальная и максимальная яркости исходного изображения равны  $x_{\min}$  и  $x_{\max}$  соответственно. Если  $x_{\min} \gg 0$  и  $x_{\max} \ll 255$ , т. е. динамический диапазон узок, то изображение выглядит серым и малоконтрастным.

При линейном растяжении контраста используется преобразование яркости типа

$$y = a*x + b,$$

где  $a$  и  $b$  определяются желаемыми значениями минимальной и максимальной яркостей результирующего изображения (обычно  $y_{\min} = 0$  для  $x_{\min}$  и  $y_{\max} = 255$  для  $x_{\max}$ ). С учетом этого преобразование яркости принимает вид

$$y = 255 * (x - x_{\min}) / (x_{\max} - x_{\min}). \quad (6.1)$$

Графическое представление этой функции и пример линейного растяжения контраста представлены на рис. 6.6 и 6.7 соответственно.

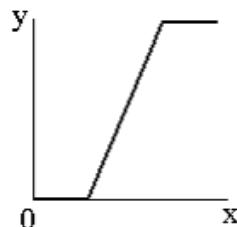


Рис. 6.6. Графическое представление функции линейного контрастирования изображения



a)



б)

Рис. 6.7. Светлое изображение (a) с диапазоном яркостей  $x_{\min}=180$ ,  $x_{\max}=240$  преобразуется посредством линейного растяжения контраста (б)

Частным случаем этого линейного изменения яркости является преобразование изображения в негативное представление:  $y = 255 - x$ .

Контрастирование можно выполнить кусочно-линейным способом. Пусть заданы  $x_1 < x_2 < \dots < x_n$  – границы диапазонов изменения яркости (рис. 6.8).

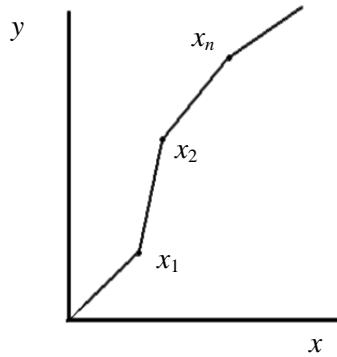


Рис. 6.8. Графическое представление функции кусочно-линейного контрастирования изображения

Тогда кусочно-линейная функция изменения яркости имеет вид

$$y = \begin{cases} a_0x + b_0, & \text{при } x < x_1, \\ a_1x + b_1, & \text{при } x_1 < x < x_2, \\ \dots \\ a_nx + b_n, & \text{при } x_n < x. \end{cases}$$

### 6.3. Арифметические операции над изображениями

Арифметические преобразования изображений относятся к точечному типу.

Умножение яркости изображения на константу:

$$y(i, j) = ax(i, j),$$

где  $a > 0$  – константа. Если  $y(i, j) > 255$ , заменить на  $y = 255$ .

Увеличение яркости на константу  $c$ :

$$z(i, j) = x(i, j) + c.$$

Сложение либо вычитание равных по размеру изображений (рис. 6.9):

$$z(i, j) = ax(i, j) \pm (1 - a)y(i, j),$$

где  $x, y, z$  – цифровые изображения одного размера,  $0 < a < 1$  – константа.

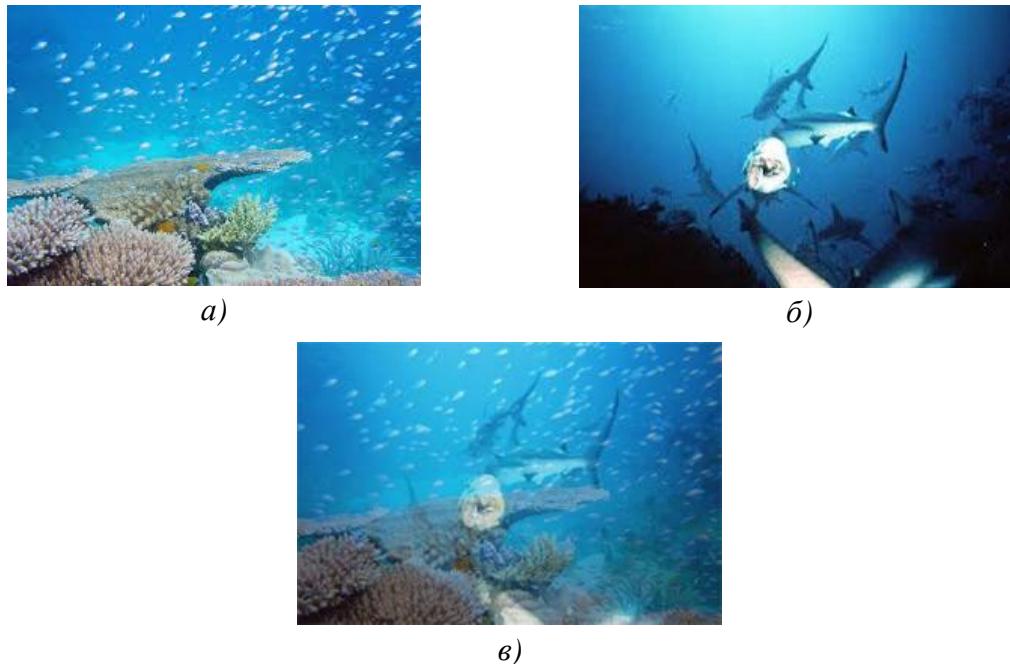


Рис. 6.9. Пример сложения двух изображений:  
а), б) исходные изображения; в) результат сложения

Пространственное маскирование изображения (рис. 6.10):

$$y(i, j) = a(i, j)x(i, j),$$

где  $a(i, j)$  – матрица (маска) из нулей и единиц того же размера, что изображение  $x$ . Операция представляет собой поэлементное произведение двух матриц.



Рис. 6.10. Пример маскирования

## 6.4. Нелинейная коррекция яркости изображений

*Гамма-коррекция* – это операция коррекции яркости изображения  $I$ , используется для одинакового представления изображений на мониторах разных конструкций с нелинейной яркостной характеристикой. Обычно используется степенная функция вида

$$I = c \cdot I^\gamma.$$

Исторически введение  $\gamma$ -коррекции было обусловлено тем, что у электронно-лучевой трубы зависимость между количеством испускаемых фотонов и напряжением на катоде близка к степенной формуле. В результате это вошло в стандарт и для современных жидкокристаллических мониторов, где зависимость между напряжением и яркостью более сложная. Стандартное значение параметра  $\gamma$  для видеоизображений NTSC равно 2,2. Для большинства мониторов значение  $\gamma$  составляет от 1,8 до 2,4.

При обработке цифровых изображений с диапазоном яркости от 0 до 255 это преобразование выполняется по формуле

$$I = 255 \cdot (I/255)^\gamma.$$

Если изображение выглядит темным или блеклым,  $\gamma$ -коррекция используется для коррекции этих недостатков.

Аналогично выполняется *логарифмическое изменение яркости* ( $c$  – задаваемая константа,  $b \leq 1$ ):

$$I = c \cdot \log(I + b).$$

Преобразование, называемое *соляризацией изображения*, изменяет яркость с помощью параболической функции:

$$y = k \cdot x(x_{\max} - x),$$

где  $x_{\max}$  – максимальное значение яркости исходного изображения (обычно 255), а  $k$  – константа, управляющая динамическим диапазоном преобразованного изображения. Функция преобразования является квадратичной параболой, ее график при  $k = 1/64$  приведен на рис. 6.11.

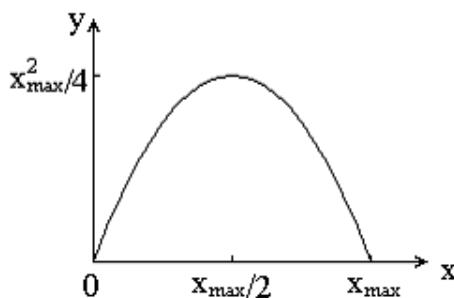


Рис. 6.11. Функция изменения яркости при соляризации

Смысл преобразования состоит в том, что участки исходного изображения с яркостью, близкой к белому цвету, после обработки принимают значения яркости ближе к черному. При этом темные участки остаются темными. Значения, близкие к белому, приобретают участки, изначально имевшие средние уровни яркости (рис. 6.12).



Рис. 6.12. Исходное изображение (а) и результат его соляризации (б)

Как видно из рис. 6.12, преобразование повышает четкость темных и светлых областей: улучшены изображения глаз, повышен контраст на переходе «лицо – волосы».

**Алгоритм коррекции яркости путем приведения средней яркости к заданному значению**

Пусть  $L_{ij}$  – яркость пикселя с координатами  $i, j$ .

**Шаг 1.** Задать требуемое значение средней яркости  $K$ .  $L = L/255$ ,  $K = K/255$ .

**Шаг 2.** Вычислить минимальное  $L_{\min}$ , максимальное  $L_{\max}$  и среднее арифметическое значения яркости исходного изображения ( $Av$ ):

$$Av = \text{mean}(L) = \frac{1}{N} \sum_{i,j} L_{ij}.$$

**Шаг 3.** Определить параметры преобразования значений яркости так, чтобы значение средней яркости стало равным  $K$ .

$$K = \frac{1}{N} \sum_{i,j} F(L_{ij}).$$

Для этого изменить значения яркости всех пикселов изображения, например, нелинейно

$$F(L_{ij}) = \left( \frac{L_{ij} - L_{\min}}{L_{\max} - L_{\min}} \right)^{\alpha}, \quad (6.2)$$

получим

$$K = \frac{1}{N} \sum_{i,j} F(L_{ij}) = \frac{1}{N} \sum_{i,j} \left( \frac{L_{ij} - L_{\min}}{L_{\max} - L_{\min}} \right)^{\alpha}.$$

Подобрать значение  $\alpha$  так, чтобы среднеарифметическое значение яркости пикселов изображения стало равно  $K$ .

**Шаг 4.** Выполнить преобразование яркости всех пикселов исходного изображения по формуле (6.2), дополнительно умножив на 255.

Примеры преобразования изображения к средним значениям яркости приведены на рис. 6.13.

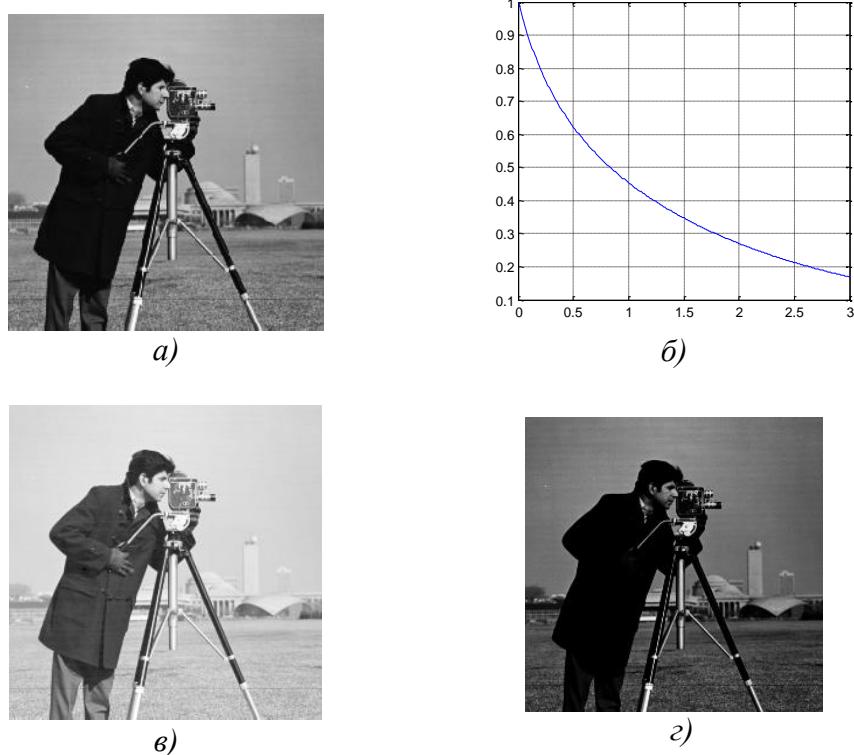


Рис. 6.13. Преобразование к заданному значению средней яркости:  
а) исходное изображение со средней яркостью 119; б) график функции  
 $K = f(a)$  при  $\alpha = 1,85$ ; в) преобразованное изображение со средней  
яркостью 177; г) преобразованное изображение со средней яркостью 77

**Табличный метод изменения яркости.** В преобразованиях точечного типа яркость пикселя меняется согласно некоторой формуле, одинаковой для всех пикселов изображения. Пусть размер изображения равен  $N \times N$ , где  $N \gg 256$ . При нелинейных изменениях яркости значений потребуется  $N^2$  раз вычислить новые значения пикселов.

От этих недостатков избавлен табличный метод замены яркости (Look-UpTable – LUT). Его суть состоит в том, что предварительно вычисляются значения функции изменения яркости  $f(x)$  для всех  $x$  от 0 до 255 и создается таблица новых значений  $y = f(x)$  (табл. 6.2). Затем яркость меняется в соответствии с этой таблицей, при этом исходная яркость  $x$  используется как индекс в полученной таблице, а  $y$  – как новое значение яркости.

Таблица 6.2

Табличное задание функции  $y = f(x)$ 

$x$	$x_1$	$x_2$	...	$x_{n-1}$	$x_n$
$y = f(x)$	$y_1$	$y_2$	...	$y_{n-1}$	$y_n$

Если исходное изображение имеет диапазон яркостей  $[0, \dots, 255]$ , то таблица LUT состоит из 256 элементов. Например, для преобразования соляризации по формуле  $y = 4x(255 - x)/255$  таблица описывается одномерным массивом:

$$0,4,8,12,16,20,23,27,\dots,198,199,201,203,\dots,254,255,255,255,\dots,12,8,4,0.$$

Новые значения яркости при таком подходе вычисляются 256 раз при формировании таблицы. Если вычислять ее по формуле для каждого пикселя, то потребуется  $N^2$  вычислений, где  $N$  может равняться нескольким тысячам. Поэтому *табличное изменение яркости является очень быстрым*.

**Интерактивное изменение функции яркости изображения.** Опишем метод преобразования яркости, аналогичный реализованному в графическом редакторе Photoshop (функция «кривые»). Изначально график изменения яркости описывается тождественной функцией  $y = x$ , где  $x$  – яркость исходного, а  $y$  – результирующего изображения (рис. 6.14). Значения  $x$  и  $y$  изменяются в диапазоне от 0 до 255.

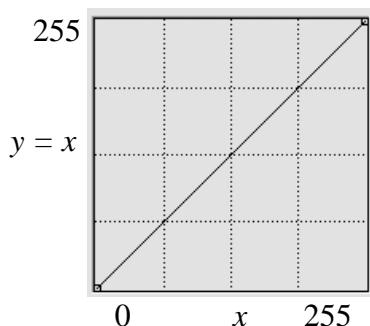


Рис. 6.14. Исходный график функции изменения яркости

Интерактивно изменяем положение некоторой точки этого графика, например  $(x_1, y_1)$ , где  $x_1 = y_1$ , на новое с координатами  $(x_{11}, y_{11})$ , при этом  $x_{11} \neq y_{11}$  (рис. 6.15).

Кривая, задающая функцию изменения яркости, проходит через точки  $P_0, P_1, P_n$  (рис. 6.15), где  $n = 2$ ,  $P_0 = (0,0)$ ,  $P_1 = (x_{11}, y_{11})$ ,  $P_n = (255, 255)$ . При  $n = 2$  это кривая второго порядка (в общем случае парабола).

Можно изменить форму графика, задав еще одну точку  $P_2 = (x_{22}, y_{22})$  между точками  $P_0$  и  $P_n$  (рис. 6.16). В этом случае кривая должна проходить через конечные  $P_0 = (0,0)$ ,  $P_n = (255, 255)$  и заданные точки  $P_1, P_2$ , при этом  $n = 3$ . Чтобы вычислить параметры такой функции, кривая

описывается полиномом степени на единицу больше числа заданных (промежуточных) точек. В данном примере – степени  $n = 3$ :

$$a_0x^n + a_1x^{n-1} + \dots + a_{n-1}x + a_n = y. \quad (6.3)$$

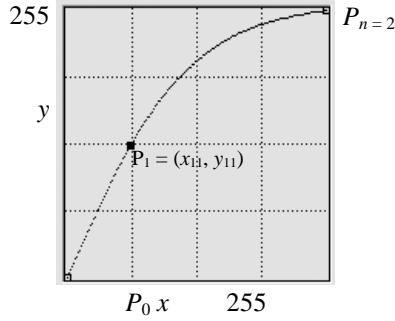


Рис. 6.15. Параболическая функция изменения яркости

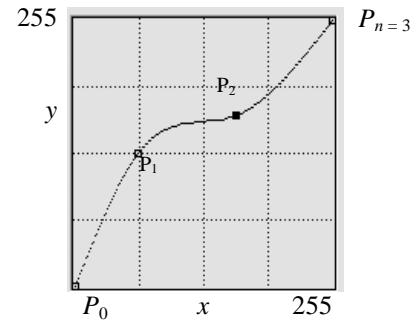


Рис. 6.16. Функция изменения яркости третьего порядка

Поскольку  $P_0 = (0,0)$ , т. е. при  $x = 0$  из уравнения (6.3) имеем  $y = 0$ , то  $a_0 = 0$ . Остальные  $a_i$  неизвестны, их количество равно  $n$ . Чтобы их найти, в уравнение (6.3) нужно подставить координаты точек  $P_i = (x_i, y_i)$ . Так,  $P_3 = (255, 255)$  означает, что при  $x=255$  имеем  $y=255$ . Получим систему из  $n$  линейных уравнений с неизвестными  $a_i$ :

$$\begin{cases} a_0x_1^n + a_1x_1^{n-1} + \dots + a_{n-1}x_1 = y_1, \\ a_0x_2^n + a_1x_2^{n-1} + \dots + a_{n-1}x_2 = y_2, \\ \dots \\ a_0x_{n-1}^n + a_1x_{n-1}^{n-1} + \dots + a_{n-1}x_{n-1} = y_{n-1}, \\ a_0255^n + a_1255^{n-1} + \dots + a_{n-1}255 = 255. \end{cases} \quad (6.4)$$

В матричном виде система (6.4) записывается как

$$X \cdot A = Y,$$

где  $Y = (y_1, y_2, \dots, y_{n-1}, 255)$  – вектор-столбец констант;  $A = (a_0, a_1, a_2, \dots, a_{n-1})$  – вектор-столбец неизвестных;  $X$  – квадратная матрица констант,

$$X = \begin{bmatrix} x_1^n & x_1^{n-1} & \dots & x_1 \\ x_2^n & x_2^{n-1} & \dots & x_2 \\ \dots & & & \\ x_{n-1}^n & x_{n-1}^{n-1} & \dots & x_{n-1} \\ 255^n & 255^{n-1} & \dots & 255 \end{bmatrix}.$$

Система имеет решение, если ее определитель не равен нулю. Это условие выполняется, если все точки  $P_i$  имеют разные координаты. Тогда неизвестные  $a_i$  находим, вычислив обратную к  $X$  матрицу  $X^{-1}$ :

$$A = X^{-1} \cdot Y.$$

Вычислив коэффициенты  $a_i$ , подставляем их в уравнение (6.3) и строим таблицу LUT для всех  $0 \leq x \leq 255$ , получаем новые значения яркости  $y$ . Если  $y < 0$ , заменяем на 0, если  $y > 255$ , заменяем на 255. LUT готова, ее графическое представление показано на рис. 6.16. Согласно полученной таблице заменяем все значения яркости в исходном изображении в цикле по строкам и столбцам.

## 6.5. Преобразования яркости на базе гистограммы изображения

*Гистограмма* – это функция распределения яркостей полутонового изображения. Пусть  $L$  – максимально возможное значение яркости (обычно 255). Вычислить гистограмму означает построить массив  $H$  размером в 256 элементов, значения которых равны количеству пикселов с яркостью, равной номеру элемента  $H(i)$ , т. е.  $i$ . Вычисления выполняются проходом по всем пикселям изображения и добавлением единицы к значению соответствующего элемента массива  $H$ .

В гистограмме типичного изображения яркость пикселов распределена неравномерно (рис. 6.17). В результате часто на темных участках изображений детали оказываются плохо различимыми.

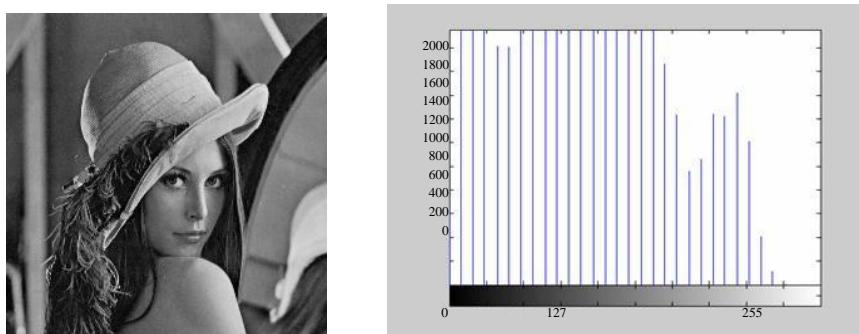


Рис. 6.17. Полутоновое изображение и его гистограмма

Разработан ряд методов улучшения изображений на базе изменения гистограммы. Наиболее простым из них является эквализация (линейное выравнивание) гистограммы. Это преобразование стремится изменить распределение яркостей изображения, делая график гистограммы максимально близким к горизонтальной прямой (рис. 6.18).

Алгоритм эквализации (линейного выравнивания) гистограммы

Шаг 1. Вычислить гистограмму исходного изображения  $f(x, y)$ , записать ее в массив  $H$ . Вычислить число пикселов  $N$ .

**Шаг 2.** Массив  $H$  нормализовать так, чтобы сумма всех элементов стала равна 255:  $H(j) = \frac{255}{N} H(j)$ .

**Шаг 3.** Вычислить кумулятивную гистограмму, суммирующую распределение яркости от 0 до  $i$

$$Sum(i) = \sum_0^i H(j).$$

**Шаг 4.** Вычислить новые значения яркости каждого пикселя с координатами  $(x, y)$  по формуле

$$g(x, y) = Sum(f(x, y)).$$

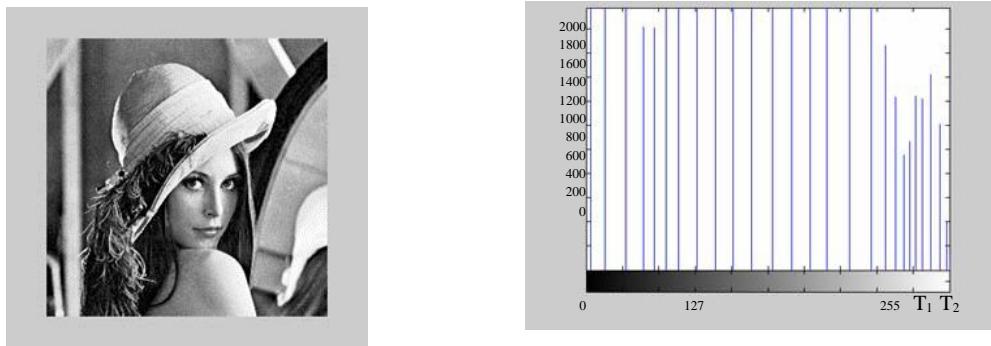


Рис. 6.18. Пример выравнивания гистограммы изображения, представленного на рис. 6.17: преобразованное изображение и его гистограмма

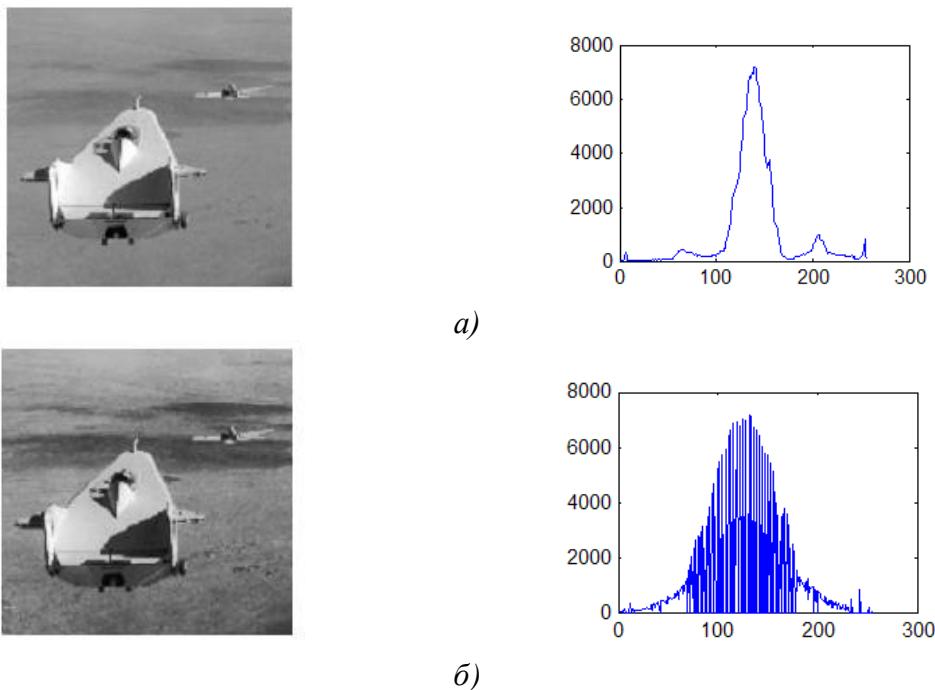


Рис. 6.19. Приведение гистограммы к распределению яркостей по нормальному закону (по функции Гаусса): *а*) исходное изображение и его гистограмма; *б*) результирующее изображение и его гистограмма

На рис. 6.18 показан результат выравнивания гистограммы изображения. Если изображение имеет много темных участков и малое число участков с высокой яркостью или наоборот, выравнивание гистограммы корректирует яркость изображения, равняя интегральные площади участков с различными яркостями.

Преобразования гистограммы могут быть и нелинейными, например, приближенные к кривой любой формы (рис. 6.19), полученные по описанному ниже алгоритму [5].

#### Алгоритм нелинейного преобразования гистограммы

**Шаг 1.** Для изображения  $X$  вычислить гистограмму  $h_x$  исходного изображения и его кумулятивную гистограмму  $H_x$ , тогда функция выравнивания этой гистограммы имеет вид

$$H_x[j] = \sum_{i=0}^j h_x[i].$$

**Шаг 2.** Определить желаемую гистограмму  $h_z$  и ее кумулятивную гистограмму  $H_z$ , тогда функция выравнивания этой гистограммы имеет вид

$$H_z[j] = \sum_{i=0}^j h_z[i].$$

**Шаг 3.** Построить таблицу LUT-преобразования яркостей. В цикле по  $i$  для всех  $0 \leq i \leq 255$  выполнить

$$j = i;$$

если  $H_x[i] \leq H_z[j]$ , тогда  $LUT[i] = j$ ;

иначе, пока  $H_x[i] > H_z[j]$ , увеличить  $j = j + 1$ ;

если  $H_x[i] - H_z[j] > H_x[i] - H_z[j-1]$ , уменьшить  $j = j - 1$ ;

иначе,  $LUT[i] = j$ .

Определить желаемую гистограмму можно разными способами, например вычислив гистограмму изображения-образца либо задав  $h_z$  в виде функции. Если функция имеет вид  $h_z = 1/256$  для всех  $i$ , результатом преобразования будет линейное выравнивание гистограммы.

## 6.6. Локально-адаптивная обработка изображений

Среднюю яркость изображения можно скорректировать глобально: увеличив ее, затемненные участки изображения можно сделать более светлыми, а уменьшив – светлые участки изображения затемнить. Однако если на изображении присутствуют и темные, и светлые области, глобальная коррекция ухудшит визуальное восприятие изображения. Статистические характеристики, необходимые для работы алгоритмов

обработки изображений, оцениваются по самим изображениям. Если характеристики не меняются по всему изображению, можно применять алгоритмы обработки с фиксированными параметрами, т. е. глобальную обработку. Но во многих случаях яркость, контраст, освещение, фон не являются однородными. Поэтому глобальные методы обработки не дают желаемый результат. В таких случаях используются адаптивные, т. е. подстраиваемые под локальные характеристики преобразования, изображения.

Простейший подход к построению адаптивных алгоритмов заключается в разбиении изображения на небольшие непересекающиеся фрагменты, в них вычисляются характеристики и каждый фрагмент обрабатывается как отдельное изображение. Достоинство такого подхода – простота, недостаток – плохаястыковка частей объектов на границах обработанных фрагментов, скачки яркости [6].

Для снижения нежелательных эффектов на краях оценки локальных характеристик вычисляют в пересекающихся фрагментах. В предельном случае эти оценки вычисляются и используются для коррекции единственного пикселя в центре фрагмента. Это обработка методом «скользящего окна», центр которого проходит все пиксели исходного изображения. Она относится к преобразованиям *локального типа*.

Рассмотрим особенности адаптивных алгоритмов применительно к преобразованиям точечного типа.

**Адаптивное повышение контраста** аналогично линейному контрастированию, но коэффициенты меняются в зависимости от значений пикселов, попавших в скользящее окно:

$$g(i, j) = a(i, j) \cdot f(i, j) + \mu - \mu_f(i, j) \cdot a(i, j),$$

где  $a(i, j) = \frac{\sigma}{\sigma_f(i, j)}$ ,  $\mu_f(i, j)$ ,  $\sigma_f(i, j)$  – локальное среднее и дисперсия;  $\mu$  и  $\sigma$  – глобальное среднее и дисперсия.

Если в окне изменение яркости мало,  $\sigma_f(i, j)$  тоже мало, поэтому диапазон значений яркости больше растягивается внутри окна. По этой же причине растягивается диапазон значений в темных участках изображения (рис. 6.20).

Еще один вариант адаптивного контрастирования исходного изображения  $L_{in}$  при вычислении интенсивности нового изображения  $L_{out}$  осуществляется следующей операцией для каждого пикселя  $(i, j)$ :

$$L_{out}(i, j) = k \frac{\bar{L}}{\sigma(i, j)} [L_{in}(i, j) - \bar{L}_{loc}(i, j)] + \bar{L}_{loc}, \quad (6.5)$$

где  $\bar{L}$  – среднее значение яркости всего изображения  $L_{in}$ ;  $\sigma(i, j)$  – среднеквадратическое отклонение яркости в локальной окрестности изображе-

ния в пикселе с координатами  $(i, j)$ ;  $\bar{L}_{loc}(i, j)$  – среднее значение яркости в окрестности с центром в пикселе  $(i, j)$ ;  $k$  – некоторая константа,  $0 < k < 1$ .

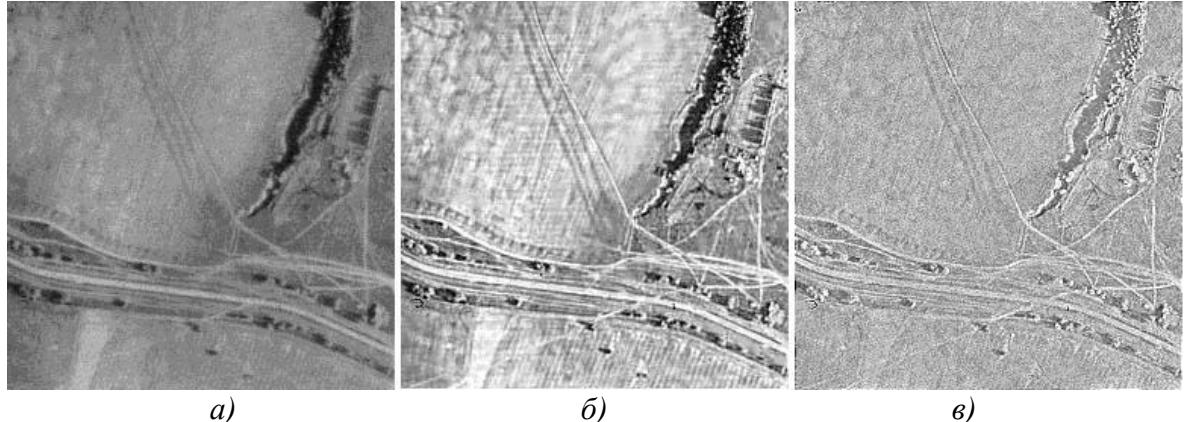


Рис. 6.20. Исходное изображение (а), результаты локального (б) и глобального (в) контрастирования исходного изображения

Отметим, что значения параметров  $\sigma$  и  $\bar{L}_{loc}$  зависят от размеров и формы окрестности пикселя, что делает этот метод адаптивным. Локальные изменения увеличиваются за счет умножения разности между значением яркости пикселя исходного изображения  $L_{in}(i, j)$  и локальным средним  $\bar{L}_{loc}(i, j)$  на  $k \frac{\bar{L}}{\sigma(i, j)}$ .

Среднеквадратическое отклонение будет меньше в малоконтрастных окрестностях и больше в окрестностях с более высоким контрастом, поэтому участки с низким контрастом будут иметь большее усиление, чем участки с большим контрастом. Целесообразно ограничивать диапазон значений множителя  $k \frac{\bar{L}}{\sigma(i, j)}$  во избежание больших отклонений интенсивностей на отдельных участках.

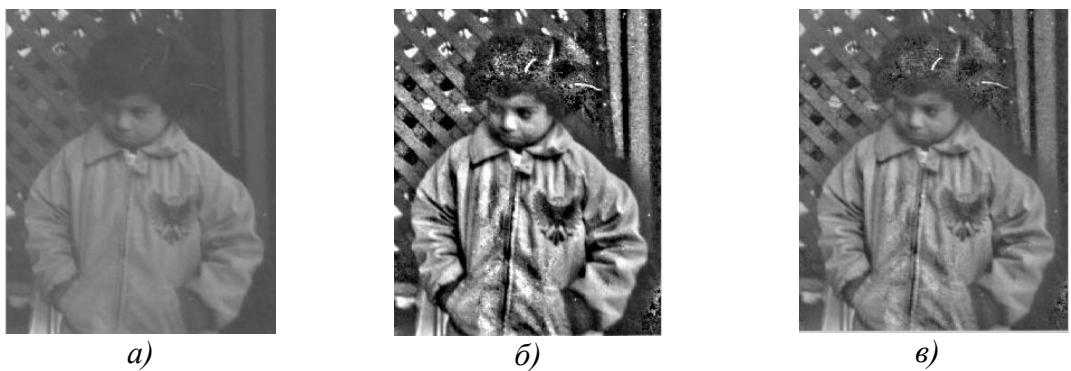


Рис. 6.21. Результаты обработки изображения, полученные с использованием формулы (6.5): а) исходное изображение; б) результирующее изображение при  $k = 0,7, m = 35$ ; в) при  $k = 0,3, m = 35$

На рис. 6.21 представлены результаты адаптивного повышения контраста по формуле (6.5). Уменьшение размера локального окна  $m$  приводит к увеличению детальности обработки. Уменьшение коэффициента  $k$  приводит к устранению резких перепадов на изображении и понижению его контрастности. Таким образом, используя различные значения параметров  $m$  и  $k$ , можно управлять уровнем контрастности и детальности обработки изображений.

*Адаптивное преобразование гистограммы* можно выполнить аналогично адаптивному повышению контраста. Оно выполняется так же, как и стандартное (глобальное) преобразование гистограммы, но применяется локально, т. е. к фрагментам изображения. Например, выравнивание гистограммы можно выполнить в скользящем окне (рис. 6.22).

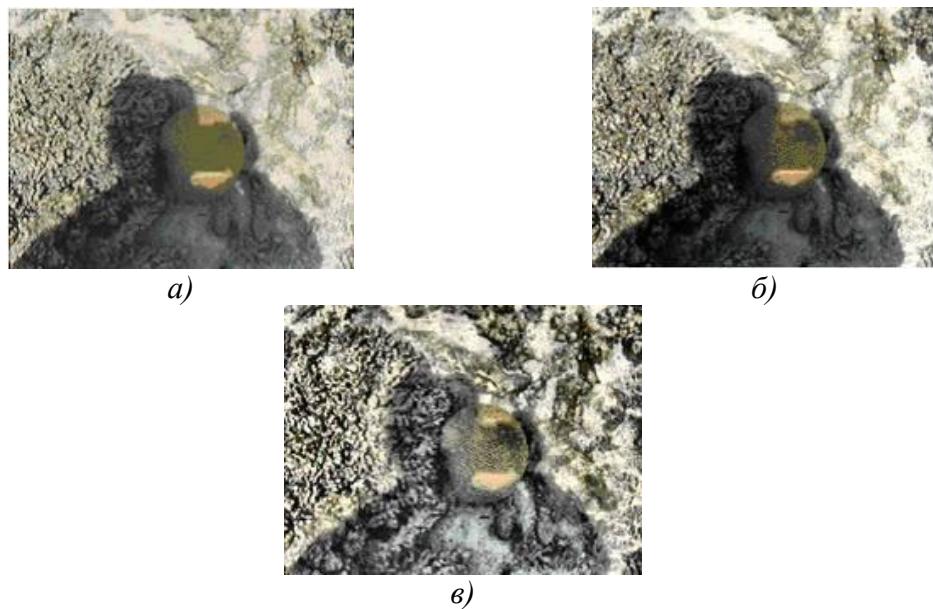


Рис. 6.22. Исходное изображение (а), после глобального выравнивания гистограммы (б), после адаптивного выравнивания гистограммы компоненты  $L$  в пространстве Lab (в), размер окна 8x8

Основной недостаток адаптивных методов обработки – увеличение объема вычислений. Для уменьшения вычислений используют рекурсивные процедуры коррекции статистических характеристик текущего фрагмента при вычислении характеристик соседнего фрагмента. Например, если фрагмент имеет форму прямоугольника и в нем вычислена гистограмма яркостей, при сдвиге вправо на один пиксель локальная гистограмма корректируется путем вычитания вклада пикселов, образующих левый столбец текущего фрагмента, и добавления пикселов, образующих правый столбец соседнего фрагмента.

## Глава 7. Преобразования локального типа

Цифровые изображения, сформированные различными оптико-электронными системами и зарегистрированные разными приемниками, искажаются под действием помех различного характера. Так, искажения изображения вносятся всеми компонентами изображающего прибора, начиная с осветительной системы (например, неравномерность освещенности предмета). Оптическая система вносит искажения, которые называются аберрациями, а электронные приемники излучения, например ПЗС-матрицы, – электронный шум. Все помехи, вносимые в цифровое изображение, принято называть *шумом*. Шум затрудняет как визуальный анализ изображений человеком, так и их автоматическую обработку.

Многие методы улучшения изображений уменьшают влияние шума, внесенного в процесс формирования цифрового изображения.

### 7.1. Типы шумов

Шум, появляющийся на изображениях, может быть классифицирован различными способами. Он может быть коррелированным или некоррелированным, зависящим или не зависящим от самого регистрируемого сигнала и т. д. Знания относительно системы получения изображений, визуальный анализ зашумленного изображения помогают в построении модели шума и оценке его статистических характеристик. Математически определенная модель шума помогает построить оптимальную для данного приложения систему улучшения изображений.

Рассмотрим наиболее часто используемые модели шумов на изображениях.

Под *импульсным шумом* понимается искажение сигнала импульсами, т. е. выбросами с очень большими положительными или отрицательными значениями и малой длительностью. При обработке изображений импульсный шум возникает, например, вследствие ошибок декодирования, которые приводят к появлению черных и белых точек на изображении. Поэтому его часто называют точечным шумом. Выбросы шума особенно заметны на очень темных или очень светлых участках изображений.

Импульсный шум моделируется следующим образом. Появление шумового выброса в каждом пикселе  $(i, j)$  изображения имеет вероятность  $p$  и не зависит ни от наличия шума в других точках изображения, ни от качества исходного изображения. Значение яркости пикселя заменяется на новое значение  $d$  (от 0 до 255). Пусть  $\{x_{i,j}\}$  – искаженное изображение. Тогда

$$x_{i,j} = \begin{cases} d & \text{с вероятностью } p, \\ s_{i,j} & \text{с вероятностью } (1-p), \end{cases}$$

где  $s_{i,j}$  – значения яркости исходного изображения. Если новое значение  $d = 0$ , то вносятся черные значения яркости (шум типа «перец»), если  $d = 255$  – белые значения яркости (шум типа «соль»).

Параметр  $d$  может также принимать случайные значения. Предполагается, что они являются независимыми случайными величинами с гауссовым распределением на интервале  $[0, 255]$ . Удаление такого шума представляет более сложную задачу.

*Аддитивный шум* описывается соотношением

$$g(x, y) = f(x, y) + \eta(x, y),$$

где  $f(x, y)$  – исходное изображение;  $g(x, y)$  – зашумленное изображение;  $\eta(x, y)$  – аддитивный и не зависящий от сигнала шум с гауссовым или другим распределением функции плотности вероятности.

Гауссов шум (называемый также нормальным шумом) возникает на изображении в результате проявления таких факторов, как шум в электрических цепях, шум сенсоров (из-за недостатка освещения и/или высокой температуры). Модель этого шума широко применяется при фильтрации изображений и сигналов, поскольку предполагает независимость и нулевое математическое ожидание помех.

Функция плотности распределения вероятностей гауссовой случайной величины  $z$  задается следующим выражением (рис. 7.1):

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\mu)^2/2\sigma^2},$$

где  $z$  определяет уровень яркости из всех возможных значений для изображения (например, для восьмибитового изображения  $0 \leq z \leq 255$ );  $\mu$  – среднее значение (математическое ожидание) случайной величины  $z$ ;  $\sigma$  – среднеквадратичное отклонение. Величина, равная квадрату среднеквадратичного отклонения, называется дисперсией  $\sigma^2$  и определяет мощность вносимого шума.

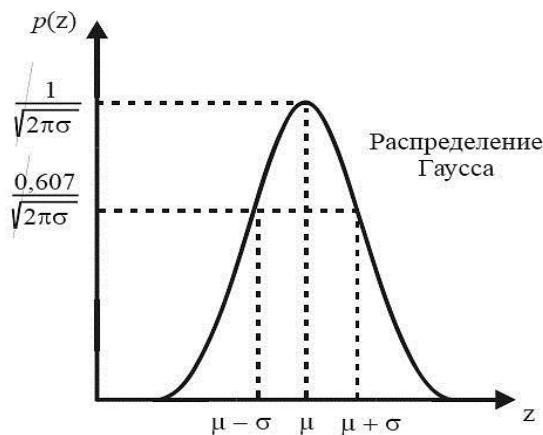


Рис. 7.1. Функция плотности распределения вероятностей гауссовой случайной величины  $z$  в форме колокола

Таким образом,  $p(z)$  определяет вероятность того, что значение текущего отсчета шума будет иметь яркость, равную  $z$ . Если случайная величина  $z$  описывается функцией Гаусса, то приблизительно 67 % ее значений сосредоточено в диапазоне  $[(\mu - \sigma), (\mu + \sigma)]$  и около 96 % в диапазоне  $[(\mu - 2\sigma), (\mu + 2\sigma)]$ .

*Мультипликативный шум* описывается соотношением

$$g(x, y) = f(x, y) \cdot \eta(x, y),$$

т. е. шумовая составляющая умножает принятый сигнал. Примерами такого шума являются: зернистость фотопленки, изображения, полученные с помощью ультразвука; радиолокационные изображения.

*Шум квантования* зависит от сигнала и выбранного шага квантования значений сигнала. Например, в результате бинаризации яркость изображения описывается двумя значениями, соответствующими черному и белому цвету. Такой шум может производить нежелательные артефакты, например ложные контуры вокруг объектов, или убирать низкоконтрастные детали на изображении. Он не устраняется.

*Спекл-шум*, как и во всех сканирующих системах с когерентным формированием изображения, вызван энергетическими помехами из-за беспорядочно распределенных отражателей сигнала, слишком мелких для того, чтобы их могла отобразить система. Это класс шумов, зависящих от самого сигнала изображения. Для объектов с низким разрешением обычно является мультипликативным. Спекл-шум появляется при получении цифровых изображений с помощью ультразвуковых медицинских сканеров и радаров. В результате изображение выглядит «зернистым», а на радарных изображениях появляются искажения в виде светлых крестов (см. рис. 3.27).

## 7.2. Фильтрация изображений

Ослабление помех достигается фильтрацией. Для этого изображение описывается двухмерной функцией, значения которой изменяются медленнее, чем значения двухмерной функции, описывающей шум. Многообразие методов фильтрации изображений связано с разнообразием математических моделей сигналов, помех и критериев оптимальности фильтрации. Фильтрация производится в пространственной или частотной областях. Так, в частотной области изображение должно быть преобразовано в частотное представление, например, с помощью преобразования Фурье, которое относится к преобразованиям глобального типа.

Рассмотрим основные методы улучшения изображений в пространственной области на примере полутоновых изображений.

Преобразования, которые для вычисления значения яркости одного пикселя учитывают значения пикселов в некоторой окрестности, называют локальными, а окрестность – *окном*. Форма и размеры окна могут быть произвольными, положение центра окна также может быть любым, но для удобства вычисления чаще используют прямоугольные окна.

Часто в операциях используют значения некоторой матрицы, имеющей те же размеры, что и окрестность. Такую матрицу называют *маской*, *фильтром*, *ядром фильтра*, а значения элементов матрицы – коэффициентами.

При линейной фильтрации изображений форма и размер используемой окрестности определяют маску фильтра. Мaska фильтра – это матрица весовых коэффициентов. Центр маски совмещается с анализируемым пикселиом, ее коэффициенты умножаются на значения яркости пикселов, накрытых маской. На практике в основном используют маски симметричной формы, центры которых совпадают с центром окрестности, т. е. маски. Распространенным видом окрестности является квадрат размером  $3 \times 3$ ,  $5 \times 5$  и т. п.

Фильтрация изображения  $f(x, y)$ , имеющего размеры  $M \times N$ , с помощью маски размерами  $m \times n$  описывается формулой

$$g(x, y) = \sum_s \sum_t w(s, t) f(x + s, y + t),$$

где  $s$  и  $t$  – координаты элементов маски относительно ее центра (в центре  $s = t = 0$ ). Такого рода преобразования называются линейными и их можно выполнить с помощью операции свертки (схематично эта операция представлена на рис. 7.2).

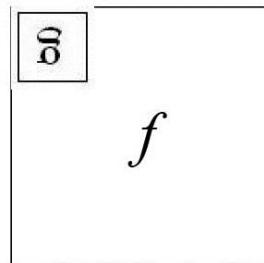


Рис. 7.2. Операция свертки изображения  $f$  с маской  $g$

Пусть  $f$  – функция яркости изображения, а  $g$  – маска фильтра. Операция свертки изображения  $f$  с помощью  $g$  обозначается как  $f * g$  и описывается следующим образом:

$$(f * g)(m, n) = \sum_{k,l} f(m - k, n - l) g(k, l). \quad (7.1)$$

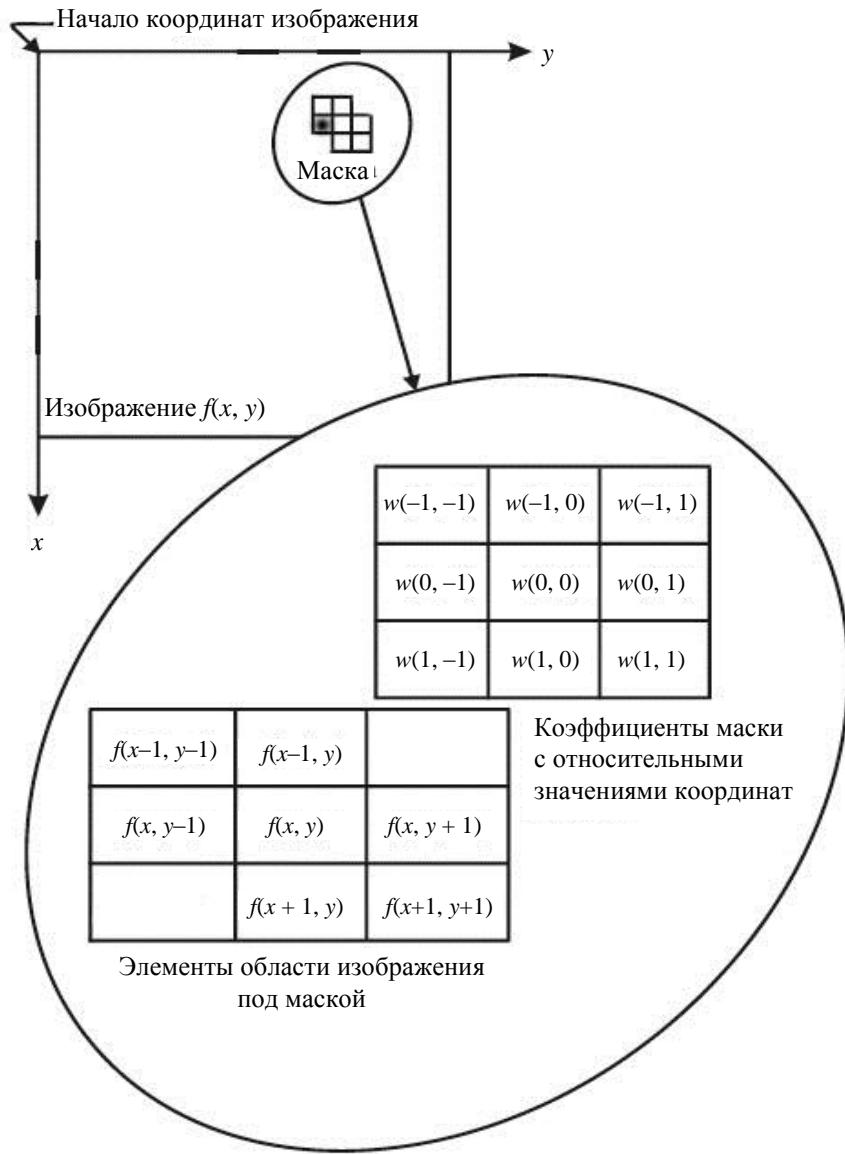


Рис. 7.3. Изображение, окно фильтра, наложенное на пикセル (показанный темным цветом), коэффициенты ядра фильтра (причем  $w(1, -1) = w(-1, 1) = 0$ ), значения яркости пикселов, попавших в окно

Отметим, если маска не симметрична, то ее надо преобразовать симметрично относительно ее же центра, а затем выполнить умножения по формуле (7.1).

Идея фильтрации изображения основывается на интегральном использовании яркости пикселов, попавших в заданное окно, с учетом коэффициентов маски фильтра (рис. 7.3). В этом состоит главное отличие фильтрации от рассмотренных выше операций обработки изображений точечного типа: фильтрация является процедурой обработки изображений локального типа. После вычисления нового значения яркости пикселя окно, в котором описана маска фильтра, сдвигается и вычисляется

яркость следующего пикселя. Поэтому подобное преобразование называют *фильтрацией в скользящем окне*.

При реализации операций пространственной фильтрации в окне возникает неопределенная ситуация, когда центр окна близок к границе изображения и несколько элементов маски находятся вне изображения. Существует три способа, учитывающих это обстоятельство:

- 1) ограничение перемещения центра окна по изображению (т. е. не выходить окном за края изображения);
- 2) расширение изображения путем добавления строк и столбцов с нулевыми значениями;
- 3) расширение изображения, повторяя строки и столбцы изображения симметрично границе.

После обработки добавленные строки и столбцы удаляются. Это позволяет получить обработанное изображение размером с исходное. Следует помнить, что значения пикселов, использовавшихся для расширения изображения, оказывают влияние на результат фильтрации пикселя внутри граничной полосы, ширина которой зависит от размеров окрестности, используемой при фильтрации. Если окрестность имеет размеры 5x5 пикселов, то с каждой стороны добавляются по две строки и столбца.

### 7.3. Низкочастотные фильтры

Низкочастотные пространственные фильтры оставляют низкочастотные компоненты изображения (области с плавными изменениями яркости) нетронутыми и ослабляют высокочастотные компоненты. Такие фильтры используются для понижения шума, содержащегося в изображении, а также для удаления высокочастотных компонент из изображения с тем, чтобы можно было тщательнее исследовать содержание низкочастотных компонент. Результатом низкочастотной фильтрации является размытие изображения. Отличительные признаки низкочастотных фильтров: неотрицательные коэффициенты маски; сумма всех коэффициентов равна единице.

Примеры ядер низкочастотных фильтров:

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

Рассмотрим основные варианты низкочастотных фильтров. Они реализуются линейными операциями локального типа.

Большую группу низкочастотных фильтров составляют усредняющие (или сглаживающие) фильтры. В усредняющих фильтрах различен

способ нахождения среднего значения яркости в окрестности. Различают арифметические, геометрические, гармонические и контргармонические усредняющие фильтры.

*Арифметический усредняющий фильтр*, или *box-фильтр*, усредняет значение яркости пикселя по окрестности путем использования маски с одинаковыми коэффициентами, например, для маски размером 3x3 коэффициенты равны 1/9, при 5x5 – 1/25. Такая нормировка необходима для того, чтобы привести значение результата фильтрации к диапазону яркости исходного изображения. Графическое представление двухмерной функции, описывающей маску этого фильтра, похоже на коробку или параллелепипед, отсюда и англоязычное название – *box-фильтр*.

На рис. 7.4 приведен пример графического представления функции

$$f(x, y) = \frac{1}{M \cdot N} \sum_{i=0}^M \sum_{j=0}^N g(i, j), \quad (7.2)$$

где  $f(x, y)$  – значение пикселя выходного изображения;  $g(i, j)$  – текущее значение пикселя исходного изображения, соответствующее центру маски;  $M$  и  $N$  – ширина и высота маски соответственно. Эта операция аналогична операции линейной фильтрации с коэффициентами маски, равными  $1/(M \cdot N)$ . Из формулы (7.2) следует, что арифметический метод усреднения является частным случаем алгоритма линейной свертки и может применяться для устранения высокочастотных помех и сглаживания изображений. Алгоритм арифметического усреднения эффективен лишь для слабо зашумленных изображений.

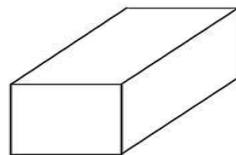


Рис. 7.4. Графическое представление функции усредняющего фильтра (box-фильтра)

*Геометрический усредняющий фильтр*. Геометрическое усреднение может быть записано в виде следующей формулы:

$$f(x, y) = \left[ \prod_{i=0}^M \prod_{j=0}^N g(i, j) \right]^{\frac{1}{M \cdot N}}.$$

При таком усреднении происходит сглаживание изображения, аналогичное арифметическому усреднению, наблюдается ухудшение резко-

сти, свойственное всем фильтрам из этого класса, но отдельные объекты исходного изображения искажаются меньше. Этот фильтр так же, как и усредняющий арифметический, может использоваться для подавления высокочастотного аддитивного шума, имея при этом лучшие статистические характеристики, хотя новые значения яркости дольше вычисляются.

*Гармонический усредняющий фильтр.* Фильтр на базе гармонического усреднения основывается на выражении

$$f(x, y) = \frac{M \cdot N}{\sum_{i=0}^M \sum_{j=0}^N \frac{1}{g(i, j)}}.$$

Данный фильтр хорошо работает с импульсными шумами типа «соль» (белыми точками) и не работает с шумами типа «перец».

*Контргармонический усредняющий фильтр* представляет собой обобщение арифметического и гармонического усреднений. Этот фильтр меняет яркость пикселя согласно функции

$$f(x, y) = \frac{\sum_{i=0}^M \sum_{j=0}^N g(i, j)^{Q+1}}{M \cdot N \cdot \sum_{i=0}^M \sum_{j=0}^N g(i, j)^Q},$$

где  $Q$  – порядок фильтра. Контргармонический фильтр идеально подходит для уменьшения или полного устранения импульсных шумов типа «соль-и-перец», причем при положительных значениях  $Q$  подавляются шумы «перец», а при отрицательных – «соль». Однако одновременное удаление биполярного импульсного шума (белых и черных точек) невозможно.

При  $Q = 0$  контргармонический фильтр превращается в арифметический, а при  $Q = -1$  – в гармонический, поэтому данный фильтр можно считать обобщенным фильтром из класса усредняющих.

*Фильтр Гаусса.* При задании фильтров можно использовать маски с разными весовыми коэффициентами (см. рис. 7.3). Логично предположить, что пиксели, расположенные ближе к анализируемому пиксели, оказывают большее влияние на яркость, вычисляемую в процессе фильтрации. К такому типу фильтров относятся фильтры, значения весов которых описываются колоколообразной функцией Гаусса

$$f(x) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$

где  $\mu$  – координата центральной точки;  $\sigma$  – вещественная константа, определяющая ширину колокола. При фильтрации изображений используется двухмерный фильтр Гаусса

$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}.$$

Другие варианты представления этой функции приведены на рис. 7.5.

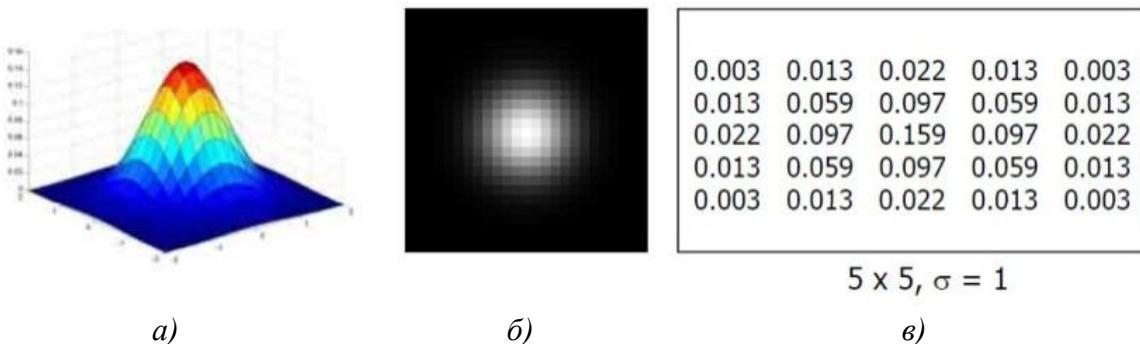


Рис. 7.5. Разное представление функции Гаусса: а) в виде поверхности; б) полутонового изображения; в) маски с коэффициентами

Чем больше  $\sigma$ , тем больше размывается изображение при применении фильтра. Значение данного параметра следует выбирать в зависимости от того, насколько сильно необходимо размыть изображение. Обычно радиус фильтра  $r$  выбирается равным  $3\sigma$ . При этом размер маски равен  $2r + 1$ , т. е. она описывается матрицей размером  $(6\sigma + 1) \times (6\sigma + 1)$ . За пределами этой окрестности значения функции Гаусса будут равны нулю или пренебрежимо малы.

Фильтр Гаусса обладает важными свойствами:

1) является сепарабельным (разделимым). Это значит, что двухмерная функция Гаусса может быть представлена в виде произведения двух одномерных функций:

$$G_\sigma(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} = \left( \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \right) \cdot \left( \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{y^2}{2\sigma^2}} \right),$$

что позволяет снизить количество вычислений с  $(2r + 1)^2$  до  $2(2r + 1)$  на каждый пиксел, т. е. примерно в  $r$  раз, где  $(2r + 1)$  – размер маски;

2) свертка, выполненная два раза с маской фильтром радиуса  $r$ , дает тот же результат, что и с маской радиуса  $r\sqrt{2}$ .

## 7.4. Высокочастотные фильтры

Высокочастотные фильтры используются для выделения перепадов яркости. На их базе строятся алгоритмы подчеркивания границ и выделения небольших объектов. Они усиливают резкие перепады на изображении, в то время как плавные подавляются. После обработки высокочастотным фильтром изображение становится более резким.

Основной класс высокочастотных фильтров выделяет края. Краевым называется пиксел, в котором резко изменяется локальная яркость изображения. Методы выделения таких пикселов называют *детекторами края*. Их связная совокупность образует границы объектов, представленных на изображении.

На изображении можно выделить два основных типа краев: в виде ступеньки (или ската) и в виде крыши (рис. 7.6).



Рис. 7.6. Основные типы краев

Как правило, резкое изменение яркости можно определить, анализируя первую производную функции яркости (рис. 7.7). Производная характеризует величину градиента. Граница объекта на изображении обычно перпендикулярна вектору градиента.

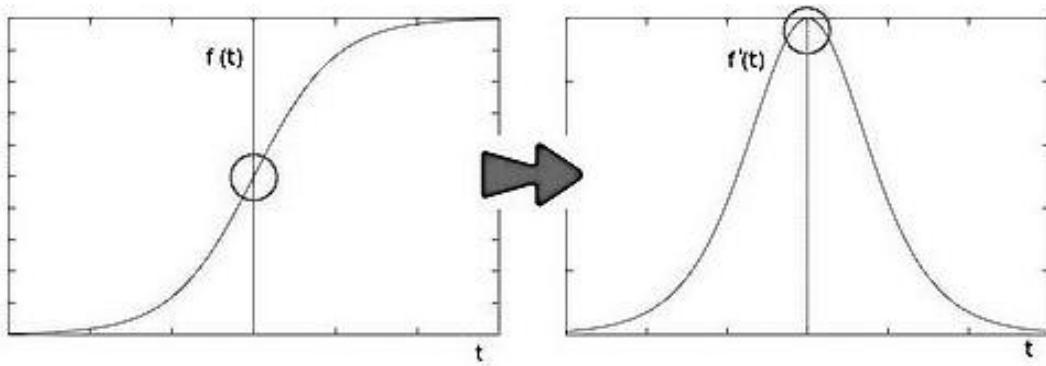


Рис. 7.7. Функция яркости и ее первая производная, максимум производной соответствует краю

Для вычисления производной применяют высокочастотные фильтры, которые аппроксимируют вычисление производных по направлению. При этом  $\Delta x$  равна единице или двум. В масках высокочастотных фильтров часть коэффициентов имеет отрицательные значения, но *сумма всех коэффициентов маски равна нулю*, что означает вычисление разности значений яркости соседних пикселов. Для ускорения вычислений все коэффициенты, как правило, целочисленные.

*Фильтр Робертса.* Метод Робертса является самым простым, быстрым и достаточно эффективным. Он работает с наименьшими возможными масками размером 2x2 (рис. 7.8).

<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>+1</td><td>-1</td></tr> <tr><td>0</td><td>0</td></tr> </table>	+1	-1	0	0	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>+1</td><td>0</td></tr> <tr><td>-1</td><td>0</td></tr> </table>	+1	0	-1	0	либо	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>+1</td><td>0</td></tr> <tr><td>0</td><td>-1</td></tr> </table>	+1	0	0	-1	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0</td><td>+1</td></tr> <tr><td>-1</td><td>0</td></tr> </table>	0	+1	-1	0
+1	-1																			
0	0																			
+1	0																			
-1	0																			
+1	0																			
0	-1																			
0	+1																			
-1	0																			
$G_x$	$G_y$		$G_x$	$G_y$																

Рис. 7.8. Маски фильтра Робертса для вычисления производных по  $X$  и по  $Y$

В результате свертки изображения с масками фильтра Робертса получаем оценку градиента (перепада яркости) по направлениям  $G_x, G_y$ .

Модуль градиента (максимум перепада) всех детекторов края вычисляется по формуле  $G = \sqrt{G_x^2 + G_y^2}$  или упрощенно  $G = |G_x| + |G_y|$ .

Направление градиента, т. е. максимального перепада яркости вычисляется по формуле  $\arctan(G_y / G_x)$ .

*Фильтр Превитта.* Метод Превитта предлагает две ортогональные матрицы (маски Превитта) разрешением 3x3 для более точного вычисления производных по  $X$  и по  $Y$ :

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix}, \quad G_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{bmatrix}.$$

*Фильтр Собела* использует разные веса в масках. Классическим вариантом этого фильтра являются маски

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}, \quad G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}.$$

*Фильтр Лапласа.* Предыдущие фильтры использовали аппроксимацию первой производной по оси  $X$  и по оси  $Y$ . В фильтре Лапласа применяется аппроксимация вторых производных по этим осям (рис. 7.9). В этом случае градиент вычисляется независимо от направления и границы выделяются точнее, чем в предыдущих методах. Лапласиан от функции  $f(x, y)$  – это сумма вторых производных:

$$L(f(x, y)) = \frac{d^2 f}{dx^2} + \frac{d^2 f}{dy^2}.$$

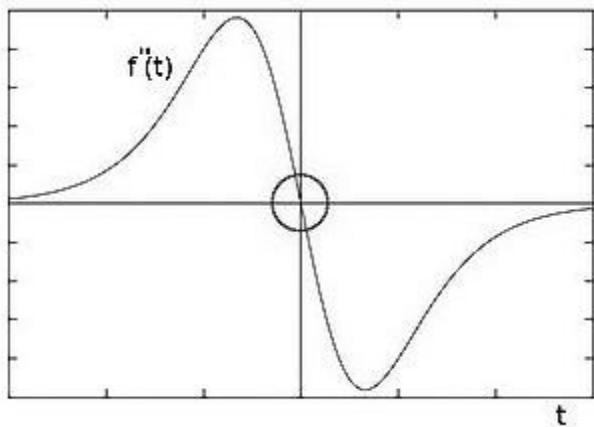


Рис. 7.9. Вторая производная функции яркости меняет знак (проходит через ноль в точке, соответствующей краю)

Для дискретных функций вторые производные могут быть аппроксимированы следующим образом:

$$L(f(x, y)) = [(f(x, y) - f(x - 1, y)) - (f(x + 1, y) - f(x, y))] + [(f(x, y) - f(x, y - 1)) - (f(x, y + 1) - f(x, y))] = -f(x, y - 1) - f(x - 1, y) - f(x, y + 1) - f(x - 1, y) + 4f(x, y).$$

Данное выражение позволяет реализовать фильтр Лапласа посредством свертки маски  $w(s, t)$  с изображением  $f(x, y)$ :

$$w(s, t) = \begin{vmatrix} 0 & -1 & 0 \\ -1 & +4 & -1 \\ 0 & -1 & 0 \end{vmatrix}.$$

После свертки близкие к нулю значения соответствуют краевым пикселам. Маска фильтра Лапласа может быть описана и другими способами с целочисленными весами, например:

$$\begin{vmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{vmatrix}, \quad \begin{vmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{vmatrix}.$$

**Выделение краев методом Кэнни.** Цель алгоритма Кэнни – более точное выделение не краевых пикселов, а связных граничных линий на изображении (рис. 7.10). Он основан на использовании двух порогов при анализе матрицы градиентов. Пороги выделяют сильные и слабые границы, причем слабые отмечаются только в тех случаях, если они соединены с сильными границами.

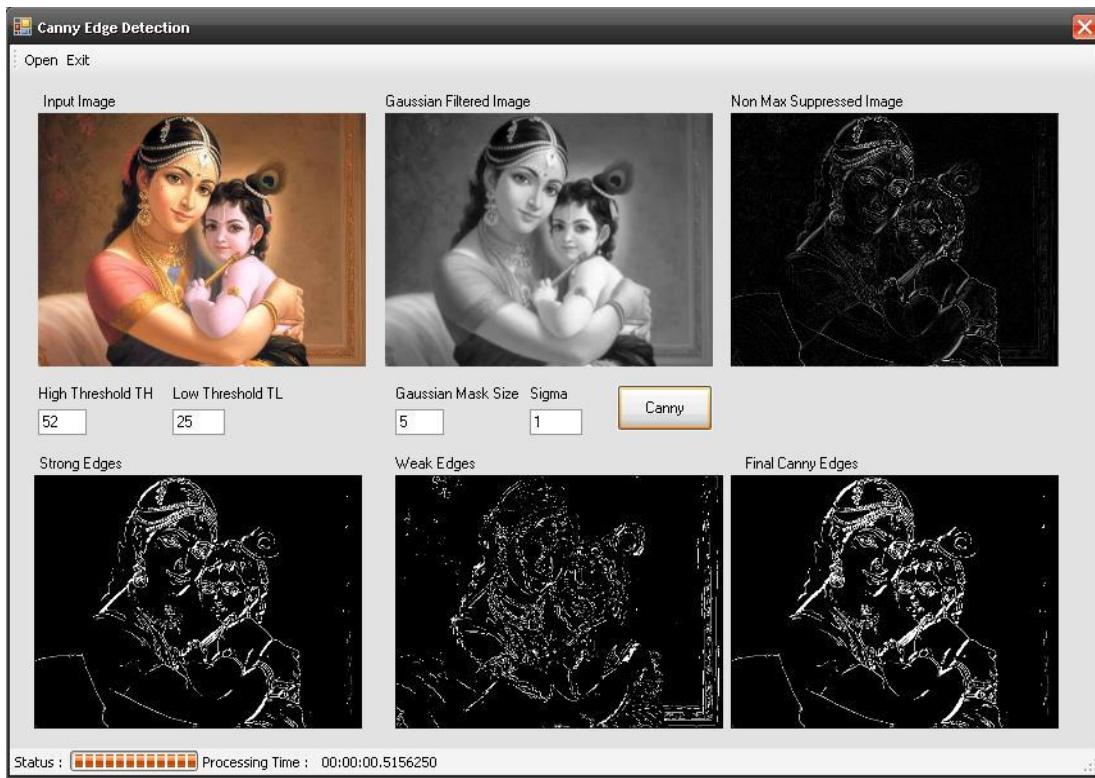


Рис. 7.10. Пример последовательной реализации алгоритма выделения границ методом Кэнни

### Алгоритм Кэнни

**Шаг 1.** Сгладить изображение. Задать параметры радиуса и сигмы. Выполнить сглаживание изображения фильтром Гаусса.

**Шаг 2.** Вычислить градиенты всех пикселов. Применить фильтр Собела для вычисления значений модуля градиента во всех пикселях изображения. Направление градиента округляется до  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  и  $135^\circ$ .

**Шаг 3.** Подавить немаксимумы модуля градиента. Краевым пикселям соответствуют максимальные значения модуля градиента. Сравнивая значения модулей градиентов, отобрать пиксели ортогонально вычисленным направлениям градиента. Если направление градиента в пикселе равно  $0^\circ$ , проанализировать значения модулей градиента ортогонально расположенных соседних пикселов, т. е. вертикальных соседей. Пиксель считается краевым, если его градиент больше, чем у соседей, иначе он – немаксимум. Аналогично анализируются значения модуля градиента, если он имеет три других направления ( $45^\circ$ ,  $90^\circ$  и  $135^\circ$ ).

После подавления немаксимумов края становятся более точными, тонкими и связными. (Простой выбор локальных максимумов выделит мало краевых пикселов.)

**Шаг 4.** Выполнить двойную пороговую фильтрацию.

Задать два пороговых значения  $T_1$  и  $T_2$ .

Для пикселов, отобранных на предыдущем шаге, уточнить, являются ли они краевыми, сравнив значения их модулей градиентов с порогом  $T_1$ . Чем ниже значение порога  $T_1$ , тем больше пикселов будет отобрано и более зашумленным будет результат. Высокое значение порога  $T_1$ , наоборот, может отбросить слабые краевые пиксели и получить линию края с множеством разрывов.

Для выделения краев используются два значения порога: если значение модуля градиента в пикселе выше верхнего порога  $T_2$ , то он принимает максимальное значение (наличие края в нем считается достоверным); если меньше нижнего порога  $T_1$  – пикセル не считается краевым. Пиксели со значениями модуля градиента, попадающими в диапазон между порогами  $T_1$  и  $T_2$ , считаются неоднозначными, им присваивают фиксированное среднее значение, а их статус уточняется на следующем шаге.

*Шаг 5. Уточнить края путем трассировки области неоднозначности.* Итоговые края находятся путем подавления всех краевых пикселов, не связанных с ранее определенными достоверными краевыми пикселями. Выделяются группы пикселов, получивших на предыдущем этапе среднее значение. Если они соединены с одним из достоверных краев, то они относятся к краю, а в противном случае отбрасываются. Пиксел добавляется к группе, когда он соседствует с ней по одному из восьми направлений (8-связное соседство).

Краевым пикселям присваивается белый цвет, остальным – черный, или наоборот (рис. 7.10).

## 7.5. Нелинейная фильтрация

Низкочастотные методы фильтрации приводят к сглаживанию резких перепадов яркости на изображении. Они линейны и оптимальны при гауссовом распределении помех в наблюдаемых данных. На реальных изображениях в районе границ разных объектов распределение яркости имеет другой вид. В таких случаях для фильтрации используют нелинейные методы. Рассмотрим их подробнее.

*Медианная фильтрация.* Помехи в виде белых или черных точек на изображении – это шумы импульсного типа. Линейные фильтры не устраниют их полностью, а лишь локально усредняют их значения. Шумы такого типа удаляются с помощью нелинейных фильтров, например медианных.

При классической медианной фильтрации используется понятие окрестности и ее центра, но *не задаются весовые коэффициенты*.

Окрестность может иметь произвольную форму и размер, а центр располагаться произвольно относительно окрестности. При совмещении центра окрестности с анализируемым пикселом окрестность становится окном, в которое попадает ряд соседних с центром пикселов. Значения яркости пикселов, попавших в окно, сортируются по возрастанию, значение *среднего в ряду (медианного) элемента после сортировки* и будет результатом медианной фильтрации в данном окне. Затем окно смещается, и процедура повторяется для всех пикселов исходного изображения.

На практике окно часто имеет прямоугольную форму с нечетным количеством элементов, а его центр располагается в геометрическом центре. Например, пусть в окно размером 3x3 попали девять пикселов, после сортировки их значений в результате медианной фильтрации центральный пикセル принимает значение 6 (рис. 7.11).

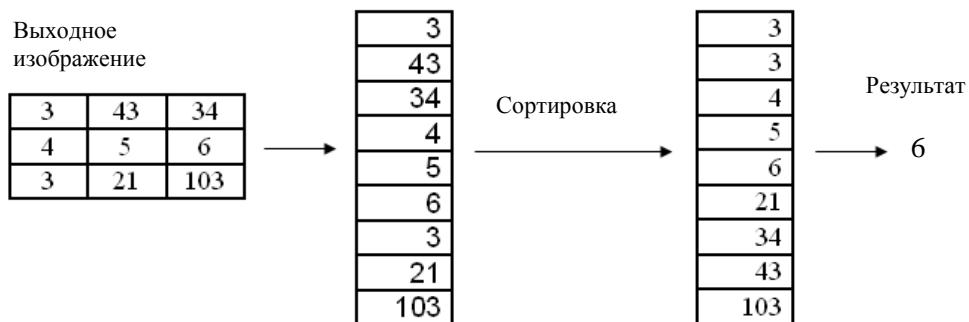


Рис. 7.11. Идея медианной фильтрации в окне 3x3

Поскольку окно может иметь произвольную форму, для его описания используют прямоугольную маску, элементы которой содержат значения 0 и 1. Множество единиц описывает окно нужной формы. В процессе фильтрации в сортировке участвуют только пиксели, которые соответствуют ненулевым элементам такой маски.

Существуют алгоритмы *взвешенной медианной фильтрации*. В маске, соответствующей окну такого фильтра, кроме 0 и 1 используются целые числа 2, 3 и др. Они означают, сколько раз повторить яркость соответствующего пикселя перед сортировкой. Как правило, большие веса располагаются ближе к центру окна для усиления влияния яркости центральных пикселов на результат. Число значений яркости  $N$ , участвующих в сортировке, равно сумме весов маски. Номер медианного элемента после сортировки будет равен  $(N + 1)/2$ .

Медианный фильтр обладает следующими свойствами:

- несепарельный;
- нелинейный;

- на полутонах изображениях не вносит новых значений яркости, которые отсутствуют в исходном изображении;
- качественно удаляет шумы импульсного типа.

Медианные фильтры успешно улучшают отсканированные старые бумажные фотографии с белыми полосами на месте сгибов.

На рис. 7.12 приводятся примеры медианной фильтрации полутонаового, зашумленного гауссовым аддитивным шумом с разными размерами окна. Медианный фильтр хорошо справляется со слабой и средней степенью зашумления, но для подавления более интенсивных шумов необходимо использовать медианный фильтр с большими размерами окна фильтрации. Отметим, что с увеличением размера окна растет способность медианного фильтра подавлять шумовую компоненту. Однако увеличение размеров окна фильтра может привести к искажению очертаний объектов. Кроме того, малые по размеру объекты могут быть целиком удалены с изображения. Поэтому в каждом конкретном случае параметры фильтра необходимо настраивать в зависимости от степени искажений и характерных размеров наблюдаемых объектов. Иногда лучше применить два раза фильтр с окном 3x3, чем один раз с окном 5x5 (рис. 7.13).

Результат медианной фильтрации не чувствителен к плотности распределения шума. Это связано с тем, что медианный фильтр с площадью окна  $2M + 1$  эффективно подавляет шумовые пиксели в окне суммарной площадью менее  $M$ . В то же время при фильтрации контрастных крупноразмерных объектов медианный фильтр не размывает и не смещает их края (точки перепада яркости).

Вариант *адаптивного медианного фильтра* использует прямоугольное окно размером  $S \times S$ . Адаптация состоит в увеличении размера окна  $S$  в процессе фильтрации в зависимости от его локальных статистических характеристик.

Введем обозначения:  $Z_{\min}$ ,  $Z_{\max}$ ,  $Z_{\text{med}}$  – минимальное, максимальное, медианное значения в окне;  $Z_{i,j}$  – значение пикселя с координатами  $(i, j)$ ;  $S_{\max}$  – максимально допустимый размер окна.

#### Алгоритм адаптивной медианной фильтрации

**Шаг 1.** Загрузить изображение. Задать параметр  $S_{\max}$  – максимально допустимый размер окна. Организовать цикл по строкам и столбцам.

**Шаг 2.** Задать исходные размеры окна фильтрации  $S$ .

**Шаг 3.** Для пикселя  $(i, j)$  с яркостью  $Z_{i,j}$  вычислить  $Z_{\min}$ ,  $Z_{\max}$ ,  $Z_{\text{med}}$ ,  $A_1 = Z_{\text{med}} - Z_{\min}$ ,  $A_2 = Z_{\text{med}} - Z_{\max}$  в заданном окне.

Если  $A_1 > 0$  и  $A_2 < 0$ , перейти на шаг 4.

В противном случае следует увеличить размер окна.

Если текущий размер окна  $S \leq S_{\max}$ , то повторить шаг 3.

В противном случае результат фильтрации равен величине  $Z_{i,j}$ .

**Шаг 4.** Вычислить  $B_1 = Z_{i,j} - Z_{\min}$ ,  $B_2 = Z_{i,j} - Z_{\max}$ . Если  $B_1 > 0$  и  $B_2 < 0$ , результат фильтрации равен  $Z_{i,j}$ . В противном случае результат фильтрации равен  $Z_{\text{med}}$ .

**Шаг 5.** Изменить координаты  $(i, j)$ . Если не вышли за пределы изображения, перейти на шаг 2. В противном случае фильтрация окончена.

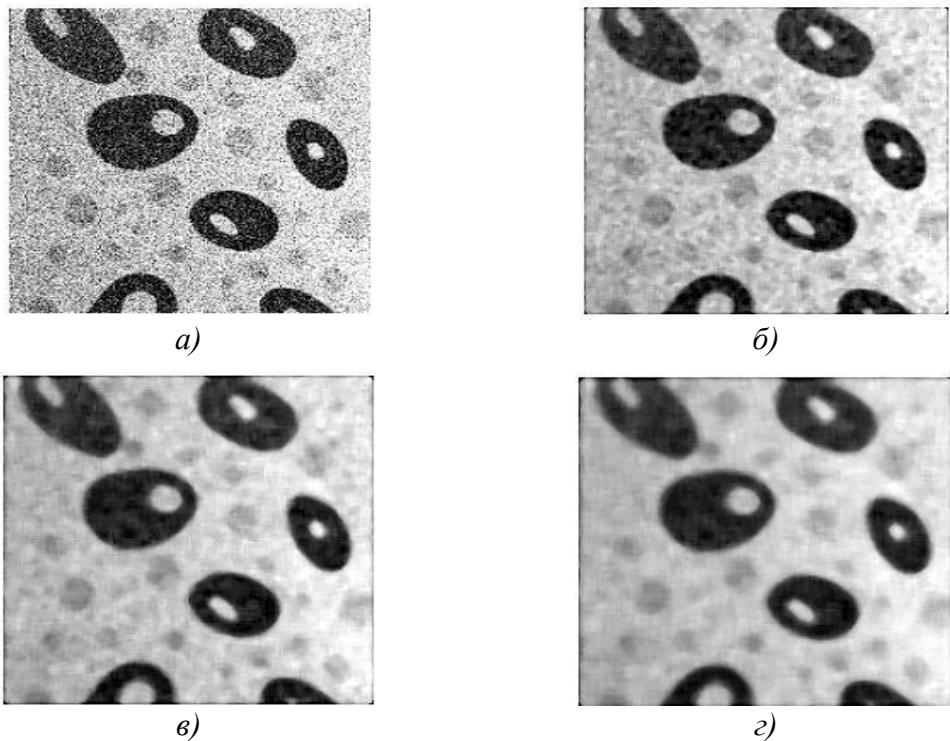


Рис. 7.12. Примеры медианной фильтрации: а) исходное изображение с импульсным шумом (сильная степень зашумления); б) результат в окне  $5 \times 5$ ; в)  $7 \times 7$ ; г)  $9 \times 9$

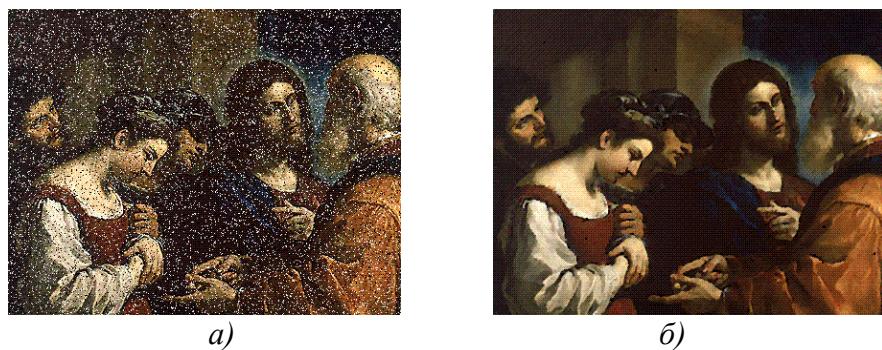


Рис. 7.13. Пример трехкратного применения медианного фильтра:  
а) шум занимает 15 % площади изображения; б) отфильтрованное изображение

Цель шага 3 – определить, является ли результат медианной фильтрации в окрестности пикселя  $Z_{\text{med}}$  импульсным шумом (черным или бе-

лым) или нет. Если условие  $Z_{\min} < Z_{\text{med}} < Z_{\max}$  соблюдается, то  $Z_{\text{med}}$  не может быть импульсом. В этом случае осуществляется переход на шаг 4 и проверяется, является ли пиксель  $Z_{i,j}$  в центре окна импульсом.

Если условия  $B_1 > 0$  и  $B_2 < 0$  верны, то  $Z_{\min} < Z_{\text{med}} < Z_{\max}$  и  $Z_{i,j}$  не может быть импульсом по той же причине, что и  $Z_{\text{med}}$ . Тогда алгоритм возвращает исходное значение яркости  $Z_{i,j}$ . Так как подобные пиксели с промежуточными значениями не изменяются (в отличие от медианного фильтра, обрабатывающего все пиксели изображения), суммарный уровень искажений в восстановленном изображении меньше (рис. 7.14).

Если условие  $B_1 > 0$  и  $B_2 < 0$  неверно, тогда  $Z_{i,j} = Z_{\min}$  или  $Z_{i,j} = Z_{\max}$ . В том и другом случае значение пикселя является критическим и алгоритм выводит медиану  $Z_{\text{med}}$ , которая, как следует из шага 3, не является импульсным шумом.

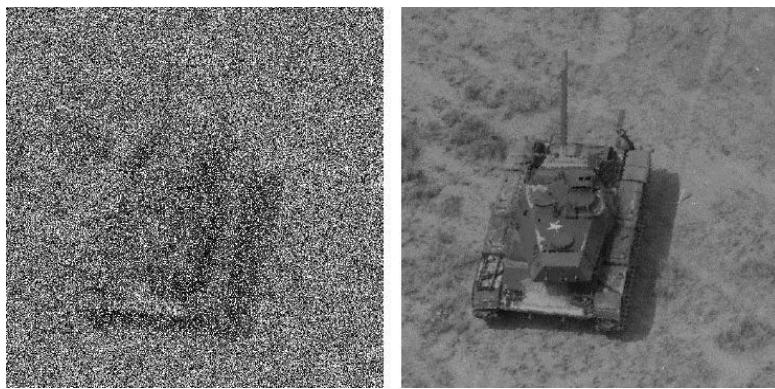


Рис. 7.14. Пример фильтрации адаптивным медианным фильтром: изображение с импульсным шумом (слева), результат фильтрации описанным алгоритмом (справа)

Предположим, что на шаге 3 обнаружен импульсный шум. Тогда увеличивается размер окна и повторяется шаг 3. Повтор шага происходит до тех пор, пока алгоритм не находит медианное значение, которое не является импульсным шумом (и выполняет переход к шагу 4), или пока не достигнут максимальный размер окна. Если это так, то алгоритм возвращает величину  $Z_{i,j}$ . Заметим, нет гарантии, что это значение не является шумовым импульсом. Чем меньше вероятность шума  $p$ , тем выше вероятность определения подходящего значения яркости. Для фильтрации шумов с высокой плотностью требуется больший размер окна. Последний этап – обработка аналогично алгоритму стандартного медианного фильтра.

Таким образом, рассмотренная модификация медианного фильтра преследует следующие цели:

- оптимальное удаление импульсных шумов;
- сглаживание других типов шумов;
- уменьшение искажений в виде потери мелких деталей.

Основной недостаток аддитивных методов обработки – увеличение объема вычислений. Для уменьшения вычислений используют рекурсивные процедуры коррекции статистических характеристик текущего фрагмента при вычислении характеристик соседнего фрагмента. Например, если фрагмент имеет форму прямоугольника и в нем вычислена гистограмма яркостей, при сдвиге вправо на один пиксель локальная гистограмма корректируется путем вычитания вклада пикселов, образующих левый столбец текущего фрагмента, и добавления пикселов, образующих правый столбец соседнего фрагмента.

*Ранговая фильтрация.* Медианный фильтр является частным случаем класса фильтров, называемых ранговыми, или порядковыми. Ранговый фильтр порядка  $r$  ( $1 \leq r \leq N$ , где  $N$  – число элементов в окрестности) выбирает из полученного ряда элемент с номером  $r$  и присваивает его значение как результат фильтрации пикселя исходного изображения.

Если число  $N$  нечетное и  $r = (N + 1)/2$ , фильтр становится медианным. Если  $r = 1$ , фильтр выбирает минимальное значение яркости в окне и называется min-фильтром. Если  $r = N$ , фильтр выбирает максимальное значение яркости в окне и называется max-фильтром.

Иногда ранг задается в процентах. Тогда выбор минимального значения соответствует 0 %, медианного – 50 %, а максимального – 100 %.

## 7.6. Внесение эффектов в изображение

Одним из способов обработки изображений является их изменение путем внесения различных эффектов. Опишем некоторые из них.

*Повышение резкости изображения.* Если к центральному элементу маски Лапласа добавить единицу и выполнить свертку изображения с такой маской, фактически получим добавление к исходному изображению градиентной информации. В результате повысится резкость исходного изображения. Для увеличения контраста между центральным пикселиом и соседями используются отрицательные весовые коэффициенты.

Пример маски для увеличения резкости изображения:

$$G = \begin{vmatrix} -0,1 & -0,1 & -0,1 \\ -0,1 & 1,8 & -0,1 \\ -0,1 & -0,1 & -0,1 \end{vmatrix}$$

Сумма коэффициентов маски равна +1, но может и немного отличаться за счет изменения значения центрального элемента маски.

*Создание эффекта тиснения.* Эффект тиснения можно придать изображению с помощью фильтров

$$T_4 = \begin{vmatrix} 0 & 1 & 2 \\ 0 & 0 & 0 \\ 0 & -1 & -2 \end{vmatrix}, \quad T_5 = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{vmatrix}, \quad T_6 = \begin{vmatrix} -2 & -1 & 0 \\ -1 & 0 & +1 \\ 0 & +1 & 2 \end{vmatrix}.$$

Коэффициенты в маске антисимметричны относительно центра или оси, проходящей через центр, а их сумма равна нулю.

При изменении знаков в маске фильтра на противоположные изменяется направление подсветки.

*Создание эффекта акварелизации.* В результате применения данного фильтра изображение будет выглядеть так, будто оно нарисовано акварелью.

На первом этапе сглаживается яркость редактируемого изображения. Можно использовать медианный фильтр или низкочастотный, например, с маской

$$M = \frac{1}{16} \begin{vmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{vmatrix}.$$



Рис. 7.15. К исходному изображению (слева) применили медианный фильтр в окне 5x5 и свертку с маской R; результирующее изображение (справа)

На следующем этапе повышается резкость переходов полученного изображения для завершения создания эффекта акварели, например, высокочастотным фильтром с маской (рис. 7.15)

$$R = \begin{vmatrix} -0,5 & -0,5 & -0,5 \\ -0,5 & 5 & -0,5 \\ -0,5 & -0,5 & -0,5 \end{vmatrix}.$$

## 7.7. «Продвинутые» алгоритмы

Вышеописанные алгоритмы не учитывают локальных особенностей изображений, и если изображение содержит очень темные и очень светлые участки, то они не могут обеспечить качественного улучшения изображения в этих участках. Для таких изображений применяют так называемые продвинутые алгоритмы, например Retinex и его модификации. Слово Retinex – композиция от слов retina (сетчатка) и cortex (кора головного мозга). Эдвин Н. Ленд в 1971 г. сформулировал теорию ретинекса, которая объясняет особенность цветопостоянства человеческого зрения, заключающуюся в том, что человек воспринимает цвет объекта примерно одинаково при изменении цвета освещения (апельсин всегда выглядит оранжевым). Он же предложил одноименный алгоритм Retinex и запатентовал его (патент принадлежит NASA).

Алгоритм Single-Scale Retinex (SSR) выравнивает освещенность изображения, сохраняя локальный контраст в плохо и ярко освещенных областях (рис. 7.16). Коррекция изображения  $I$  выполняется по формуле

$$R(x, y, \sigma) = \log[I(x, y)] - \log[I(x, y)*G(x, y, \sigma)], \quad (7.3)$$

где  $G$  – гауссиан;  $\sigma$  – параметр размытия;  $*$  – оператор свертки.

После применения формулы (7.3) большинство значений  $R$  находится в диапазоне  $[-1; 1]$ . Для визуализации  $R$  его значения необходимо преобразовать в диапазон  $[0-255]$  и округлить до целых значений по формуле  $I = \text{Int}(127,5*(R + 1))$ .



Рис. 7.16. Пример применения алгоритма Single-Scale Retinex при  $\sigma = 18$ : исходное изображение (слева), результат (справа)

Алгоритм SSR можно применить к исходному изображению несколько раз с разными параметрами фильтра Гаусса и вычислить взвешенную сумму яркостей каждого пикселя, тогда получим алгоритм Multi-Scale Retinex (MSR) (рис. 7.17):

$$MSR = w_1 \cdot SSR_1 + w_2 \cdot SSR_2 + \dots + w_n \cdot SSR_n, \quad w_1 + w_2 + \dots + w_n = 1.$$

Обычно  $n$  невелико, три или пять.



Рис. 7.17. Примеры применения алгоритма Multi-Scale Retinex для значений  $\sigma = 7, 480, 4800$  (слева) плюс линейное растяжение контраста (справа)

### *Сравнение фильтров*

Выбор конкретного фильтра и его параметров зависит от задачи и условий ее реализации, например скорости вычислений.

Арифметический усредняющий фильтр помимо подавления шума искажает резкие границы и размывает мелкие детали изображения, но он самый быстрый.

Фильтр Гаусса меньше размывает мелкие детали и лучше убирает шум, достаточно быстр, если коэффициенты вычисляются один раз.

Медианный фильтр резких границ не искажает, убирает мелкие детали, изображение становится более однородным. Если маска велика, сортировка замедляет вычисления.

Адаптивный фильтр меньше искажает детали, но зависит от большего числа параметров, требующих дополнительных вычислений.

«Продвинутые» фильтры лучше сохраняют детали и меньше размывают контуры на изображении, но они часто сложны в реализации и работают медленнее.

## **Глава 8. Операции математической морфологии**

Одним из сравнительно новых направлений в обработке изображений является математическая морфология. Она базируется на теории множеств, представлении изображений в виде геометрических форм и их анализе. Напомним основные понятия из теории множеств.

Если  $X = \{x\}$  и  $Y = \{y\}$  – множества, то объединением множеств  $X$  и  $Y$  называется множество  $Z = \{z : z \in X \text{ или } z \in Y\} = X \cup Y$  (множество, состоящее из таких элементов  $z$ , которые принадлежат  $X$  или  $Y$ ), а пересечением множеств  $X$  и  $Y$  называется множество  $Z = X \cap Y = \{z : z \in X, z \in Y\}$ . Множество  $Z = X^c = \{z : z \notin X\}$  называется дополнением множества  $X$ . Разностью множеств  $X$  и  $Y$  называется множество

$Z = \{z : z \in X, z \notin Y\} = X \setminus Y$ . Множество  $Z$  называется пустым, если оно не содержит ни одного элемента, обозначается  $Z = \emptyset$ . Справедливы следующие соотношения:

$$(X \cup Y)^C = X^C \cap Y^C;$$

$$(X \cap Y)^C = X^C \cup Y^C;$$

$$X \setminus Y = X \cap Y^C;$$

Бинарные изображения легко интерпретировать в виде множеств. Если яркость пикселя равна единице (белый цвет), он принадлежит некоторому множеству, если нулю (черный) – не принадлежит.

Бинарное изображение можно рассматривать как индикаторную функцию набора множеств, т. е. пятен (например, объединения множеств на рис. 8.1). К такому представлению бинарных изображений применим математический аппарат теории множеств.

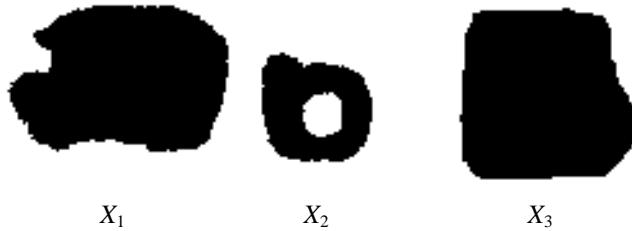


Рис. 8.1. Примеры бинарных изображений

Для любого множества  $X$  можно задать некоторый элемент (не обязательно принадлежащий этому множеству), называемый центром (или началом координат) этого множества. Обозначим через  $X_a$  множество  $X$ , центр которого помещен в точку  $a$ .

Для бинарных изображений легко определить меру множества  $X$  – это его площадь или количество входящих в множество пикселов.

Одним из основных понятий математической морфологии является понятие структурного элемента. *Структурный элемент*  $B$  – это множество точек, для которого определен его центр (не обязательно являющийся центром этого множества). Важно, что структурный элемент определяет некоторую геометрическую форму, с помощью которой выполняются морфологические операции. Он может иметь произвольный размер и форму, но на практике чаще используются симметричные элементы (прямоугольник, круг), а центром являются их геометрические центры.

Рассмотрим основные операции математической морфологии, используемые для обработки бинарных изображений [20].

## 8.1. Математическая морфология бинарных изображений

### 8.1.1. Эрозия и дилатация

Пусть задан структурный элемент  $B$ . Эрозией множества  $X$  называется множество  $Y$ , состоящее из тех элементов исходного множества  $X$ , для которых выполняется условие (рис. 8.2):

$$Y = X \ominus B = \{x : B_x \leq X\}.$$

Другими словами, если  $x \in X$ , а  $b \in B$ , то в множество  $Y$  включаются такие элементы  $x$ , для которых при всех  $b$  выполняется условие

$$x + b \in X. \quad (8.1)$$

Эрозию можно трактовать следующим образом: помещаем центр структурного элемента в точку  $x \in X$ , если элемент полностью принадлежит  $X$ , тогда точка  $x \in Y$ , где  $Y$  – множество подобных точек. Очевидно, что после этой операции множество  $Y$  меньше  $X$ .

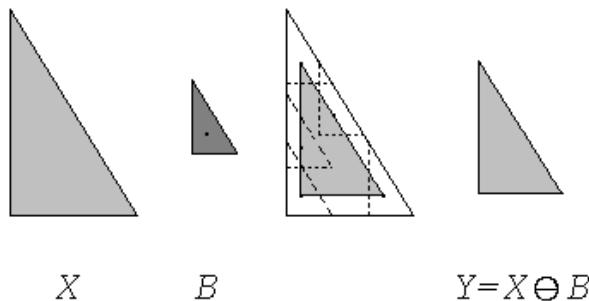


Рис. 8.2. Пример выполнения операции эрозии множества  $X$  с элементом  $B$

С другой стороны, если точка  $b$  пробегает все возможные положения в множестве  $B$ , условие (8.1) выполняется тогда и только тогда, когда  $x$  принадлежит смещенному множеству  $X_{-b}$  (рис. 8.3). Поэтому операцию эрозии можно записать как

$$Y = X \ominus B = \bigcap_{b \in B} X_{-b} = \bigcap_{-b \in B} X_b = \bigcap_{b \in B} X_b,$$

где  $\overline{B} = \{b : -b \in B\}$  – множество, симметричное элементу  $B$  относительно его центра.

Операцией, двойственной к эрозии, является *дилатация* (или расширение) (рис. 8.4), которая определяется следующим образом:

$$Y = X \oplus B = \left\{ x : B_x \cap X \neq \emptyset \right\}.$$

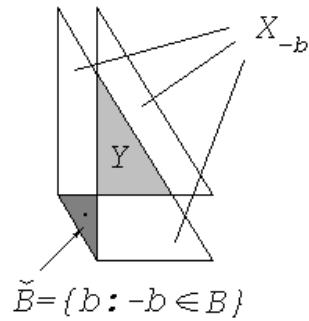


Рис. 8.3. Эрозия как пересечение смещенных множеств

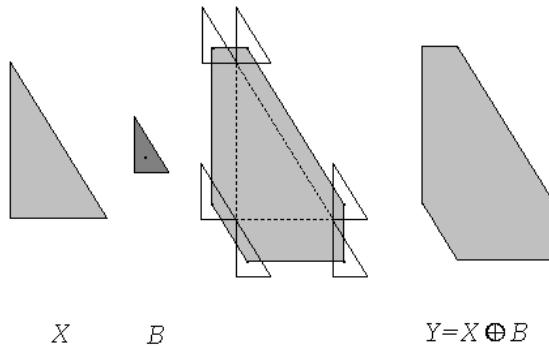


Рис. 8.4. Пример выполнения операции дилатации множества  $X$  с элементом  $B$

Эту операцию можно записать по-другому (рис. 8.5):

$$Y = X \oplus B = \bigcup_{-b \in B} X_b.$$

Дилатацию можно трактовать следующим образом: помещаем центр структурного элемента во все точки  $x \in X$  и добавляем к  $X$  все точки смещенного структурного элемента. Объединение полученных точек определяет множество  $Y$ . Очевидно, что после этой операции множество  $X$  увеличивается.

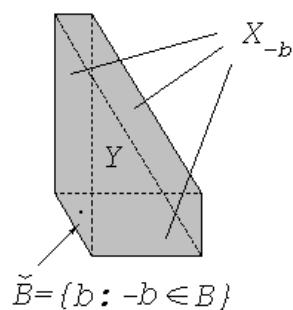


Рис. 8.5. Дилатация как объединение смещенных множеств

Если рассматривать множество  $X$  как объект, а его дополнение  $X^C$  как фон на изображении, то дилатацию объекта можно интерпретировать как эрозию фона:

$$X \oplus B = (X^C \Theta B)^C. \quad (8.2)$$

Доказательство тривиально:

$$(X^C \Theta B)^C = (\bigcap_{b \in B} X_b^c)^c = \bigcup_{b \in B} X_b = X \oplus B.$$

Операции эрозии и дилатации являются базовыми в математической морфологии. Приведем их основные свойства.

1. *Дилатация и эрозия дистрибутивны относительно объединения множеств*

$$\left(\bigcup_{i=1}^N X_i\right) \oplus B = \bigcup_{i=1}^N (X_i \oplus B), \quad \left(\bigcup_{i=1}^N X_i \Theta B\right) = \bigcup_{i=1}^N (X_i \Theta B)$$

и относительно их пересечения

$$\left(\bigcap_{i=1}^N X_i\right) \oplus B = \bigcap_{i=1}^N (X_i \oplus B), \quad \left(\bigcap_{i=1}^N X_i \Theta B\right) = \bigcap_{i=1}^N (X_i \Theta B).$$

Свойство дистрибутивности с учетом соотношения (8.2) позволяет выполнять операции над  $X$  по фрагментам, комбинируя затем результаты посредством объединения или пересечения.

2. *Дилатация ассоциативна*

$$(X \oplus A) \oplus B = X \oplus (A \oplus B),$$

а эрозия – нет, но верно следующее:

$$(X \Theta A) \Theta B = X \Theta (A \oplus B).$$

Данные свойства позволяют упростить выполнение операций посредством разложения сложных структурных элементов в композицию более простых (рис. 8.6). При этом операции со сложными структурными элементами выполняются быстрее как последовательность операций с более простыми элементами, содержащими меньшее количество точек. Так, эрозию посредством структурного элемента  $H$ , приведенного справа на рис. 8.6, можно заменить четырьмя последовательными операциями эрозии со структурными элементами  $A, B, C, D$ , что значительно уменьшает объем вычислений.

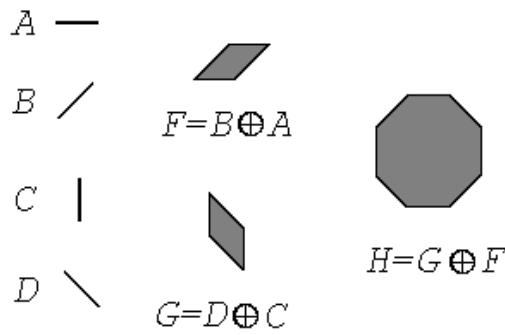


Рис. 8.6. Декомпозиция структурных элементов  $F$ ,  $G$  и  $H$

3. *Инвариантность к изменению масштаба:*

$$\begin{aligned}\lambda X \oplus \lambda B &= \lambda(X \oplus B); \\ \lambda X \ominus \lambda B &= \lambda(X \ominus B),\end{aligned}$$

где  $\lambda B$ ,  $\lambda A$  – множества, промасштабированные относительно своих центров (рис. 8.7):

$$\begin{aligned}B' = \lambda B &= \{b' = \lambda b : b \in B\}, \\ X' = \lambda X &= \{x' = \lambda x : x \in X\}.\end{aligned}$$

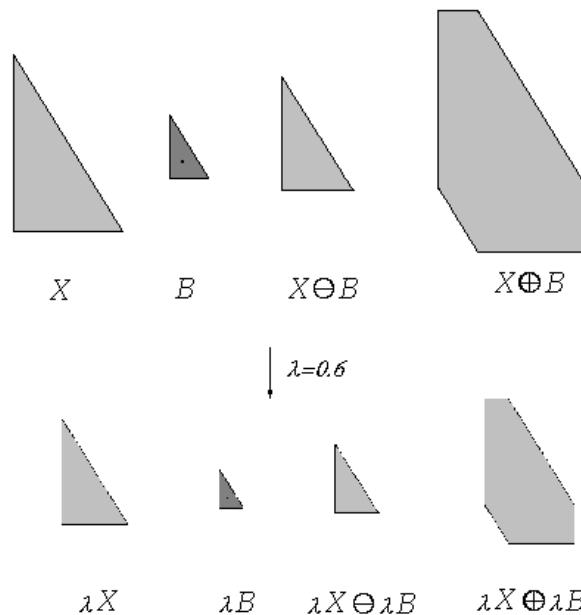


Рис. 8.7. Операции эрозии и дилатации инвариантны к масштабированию с коэффициентом  $\lambda$

Если центр структурного элемента  $B$  находится внутри него, тогда для любых множеств  $A$  и  $C$  справедливы следующие отношения:

если  $A \subseteq C$ , то  $A \ominus B \subseteq C \ominus B$ ,  $A \oplus B \subseteq C \oplus B$ .

Эрозия и дилатация – операции, предназначенные в первую очередь для выделения на бинарном изображении объектов (связных множеств точек) с определенной формой путем использования структурных элементов разной формы и размеров. Например, эрозия посредством круга с радиусом  $r$  позволяет найти в изображении объекты, минимальный диаметр которых превышает  $2r$ . Если же в качестве структурного элемента взять две точки, образующие вектор  $B$ , эрозия позволяет выделить объекты, имеющие соседей в направлении и на расстоянии, заданным этим вектором (рис. 8.8). Под объектами подразумеваются односвязные множества пикселов.

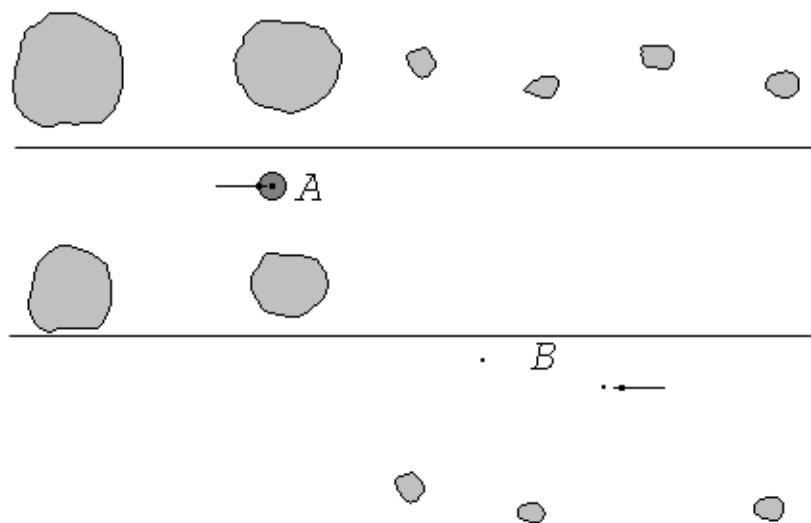


Рис. 8.8. Верхний ряд: исходное множество  $X$ ; средний ряд: структурный элемент  $A$  и результат операции  $X \Theta A$ ; нижний ряд: структурный элемент  $B$  и результат операции  $X \Theta B$  (стрелками отмечены центры структурных элементов  $A$  и  $B$ )

Одно из простейших применений дилатации – это устранение разрывов линий путем их соединения. На рис. 8.9, *a* показано изображение с разорванными символами. Пусть известно, что максимальная длина разрывов составляет два пикселя. На рис. 8.9, *б* показан наименьший структурный элемент, позволяющий устранить разрывы. Результат дилатации исходного изображения с таким примитивом показан на рис. 8.9, *в*. В местах разрывов образовались «мостики». Одно из преимуществ морфологического подхода по сравнению с методом устранения разрывов путем низкочастотной фильтрации состоит в том, что морфологический метод сразу дает бинарное изображение. При низкочастотной фильтрации из исходного бинарного изображения вначале получается полутоновое, которое затем требуется преобразовать обратно в бинарное.

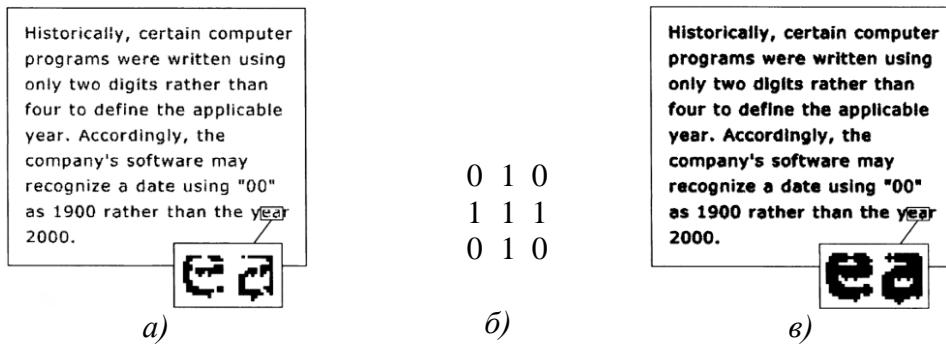


Рис. 8.9. Пример дилатации текста с недостаточным разрешением

Рассмотрим способ морфологического выделения границ (рис. 8.10). Определим морфологическую границу множества  $X$  (т. е. всех темных пятен на изображении) как разность дилатации и эрозии этого множества с круглым структурным элементом  $B$ :  $\delta X = X - (X \ominus B)$ .

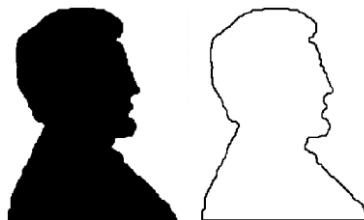


Рис. 8.10. Пример морфологического выделения границы бинарного объекта

### 8.1.2. Отмыкание и замыкание

В общем случае невозможно полностью восстановить исходное множество  $X$  после выполнения эрозии  $X \ominus B$  путем применения операции дилатации с тем же структурным элементом  $B$ , но последовательное выполнение эрозии и дилатации порождает новые операции математической морфологии.

Операция *отмыкания* множества  $X$  посредством структурного элемента  $B$  определяется формулой

$$A \circ B = (A \ominus B) \oplus B. \quad (8.3)$$

Операция *замыкания* множества  $X$  посредством структурного элемента  $B$  определяется применением эрозии и дилатации в обратном порядке

$$A \bullet B = (A \oplus B) \ominus B. \quad (8.4)$$

Рассмотрим основные свойства операций (8.3) и (8.4). Они двойственны друг другу:

$$A \bullet B = (A^c \circ B^s)^c. \quad (8.5)$$

Применительно к бинарным изображениям соотношение (8.5) означает, что операция замыкания эквивалентна дополнению результата отмыкания дополнения  $A$  с элементом  $B^S$ , симметричным  $B$  относительно центра. Очевидно, если  $B$  симметричен, то  $B = B^S$ .

Важно следующее свойство операций (8.3)–(8.5), их идемпотентность:

$$(A \circ B) \circ B = A \circ B \text{ и } (A \bullet B) \bullet B = A \bullet B.$$

Данное свойство означает неизменность результата при повторении операций отмыкания и замыкания.

Если центр структурного элемента  $B$  находится внутри него, тогда для любого множества  $A$  справедливо

$$A \Theta B \subseteq A \circ B \subseteq A \subseteq A \bullet B \subseteq A \oplus B.$$

Отмыкание и замыкание используются для фильтрации бинарных изображений: сглаживания формы объектов путем удаления мелких и узких частей, удаления мелких полостей и узких «каналов» соответственно. Степень сглаживания и размеры удаляемых артефактов зависят от размеров и формы структурного элемента.

Рассмотрим способ морфологического выделения углов и острых пиков. Детектор острых углов можно определить как разность между отмыканием множества  $X$  и замыканием  $X$  с круглым структурным элементом  $B$ :

$$gX_m = X \circ B - X \bullet B.$$

Метод нормализации фона изображения использует два структурирующих элемента  $B_1$  и  $B_2$ . Элемент  $B_1$  выбирается очень малым, чтобы сохранить даже самые малые детали объекта на изображении, но отсечь импульсный шум. Элемент  $B_2$  имеет большие размеры, чтобы подавить любой, даже самый большой по геометрическим размерам, объект (пятно) на изображении. Операция нормализации фона – это разность операций отмыкания и замыкания:

$$hX = (X \circ B_1) \bullet B_1 - ((X \circ B_1) \bullet B_1) \circ B_2 \bullet B_2.$$

Комбинируя более сложные последовательности базовых операций математической морфологии, можно сконструировать другие операции, ориентированные на решение конкретных задач.

### 8.1.3. Операция уточнения

Цель уточнения – получить связное множество пикселов толщиной не более одного. Эта операция выполняется при обработке изображений, содержащих тонкие линии: чертежи, текст, отпечатки пальцев и т. п.

Операция уточнения определяется следующим образом:

$$Y = XOT = X \setminus (X \otimes T),$$

где  $T = (T_1, T_2)$  – структурный элемент, состоящий из двух непересекающихся подмножеств  $T_1$  и  $T_2$ .

Отметим, что если начало структурного элемента принадлежит  $T_1$ , то  $(X \otimes T) \subseteq X$ , если же начало принадлежит  $T_2$ , то  $X \otimes T \subseteq X^C$ . Поэтому  $XOT = X$  при любом  $T_1$ . Чтобы избежать этого при выполнении операции уточнения, начало структурного элемента не должно принадлежать  $T_1$ .

#### 8.1.4. Модифицированный алгоритм Зонга – Суня

Т. Зонг и У. Сунь в 1984 г. предложили быстрый алгоритм уточнения объектов на бинарных изображениях. В 1988 г. Ю. Зонг и П. Вонг [27] модифицировали этот алгоритм, упростив и ускорив его в 1,5 раза.

Исходное изображение – бинарное со значением яркости 0 и 1. Будем использовать следующую нумерацию соседей анализируемого пикселя  $P_1$ :

$$\begin{bmatrix} P_{10} & P_{11} & P_{12} & P_{13} \\ P_9 & P_2 & P_3 & P_{14} \\ P_8 & P_1 & P_4 & P_{15} \\ P_7 & P_6 & P_5 & P_{16} \end{bmatrix}.$$

Введем обозначения:  $B(P_1) = \sum_{i=2}^9 P_i$ ,  $A(P_1)$  – число переходов яркости

с 0 на 1 в последовательности соседних пикселов  $P_9, P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9$ ,

#### Модифицированный алгоритм уточнения

**Шаг 1.** В цикле по  $i$  и  $j$  для каждого пикселя  $P(i, j)$  выполнить:  $P_1 = P(i, j)$ .

Пиксель  $P_1$  присваивается 0, если

$$P_1 = 1,$$

$$2 \leq B(P_1) \leq 6,$$

$$A(P_1) = 1,$$

$$(P_2 \times P_4 \times P_8 = 0 \text{ или } P_{11} = 1),$$

$$(P_2 \times P_4 \times P_6 = 0 \text{ или } P_{15} = 1).$$

**Шаг 2.** Если на предыдущем шаге изображение изменилось, повторить шаг 1.

Иначе – закончить работу.

Пример выполнения операции уточнения представлен на рис. 8.11.



Рис. 8.11. Исходное бинарное изображение (слева),  
результат уточнения (справа)

Скелетон множества точек, заданного на дискретной решетке, только приближенно напоминает скелетон непрерывного множества. Более того, для одного и того же множества результат построения скелетона посредством последовательных утончений может быть различным в зависимости от порядка применения структурных элементов в последовательности (такие топологические свойства скелетона, как количество связных компонент, точек разветвления, ветвей, концевых точек и другие при этом сохраняются).

## 8.2. Математическая морфология полутоновых изображений

Рассмотрим наиболее простой вариант обобщения операций математической морфологии для обработки полутоновых изображений. Полутоновое изображение описывается функцией яркости  $f(x)$  и обрабатывается плоским структурным элементом  $B$ . При этом можно считать, что элемент также описывается функцией нулевой яркости, т. е.  $g(z) = 0$ , если  $z \in B$ . В этом случае

$$(f \oplus B)(x) = \max\{f(x - z), z \in B\};$$

$$(f \ominus B)(x) = \min\{f(x + z), z \in B\}.$$

Результаты выполнения операций эрозии и дилатации полутонового изображения представлены на рис. 8.12. Первая расширяет множество темных пикселов, вторая – светлых.

Эрозию полутонового изображения плоским структурным элементом можно объяснить геометрически. Представим, что функция яркости задает некую поверхность. Двигаем плоский структурный элемент внутри поверхности по ее краю так, чтобы элемент полностью поместился внутри. Поверхность, очерченная центром элемента *внутри* исходной поверхности, будет требуемым результатом.

Дилатацию полутонового изображения плоским структурным элементом можно трактовать подобным образом. Функция яркости задает некую поверхность. При дилатации, если структурный элемент не симметричен относительно своего центра, преобразуем его, сформировав новый, симметричный относительно центра элемент. Двигаем центр преобразованного структурного элемента по поверхности, расширяя ее в стороны в соответствии с формой элемента. Поверхность, очерченная краями элемента *снаружи* исходной поверхности, будет требуемым результатом.

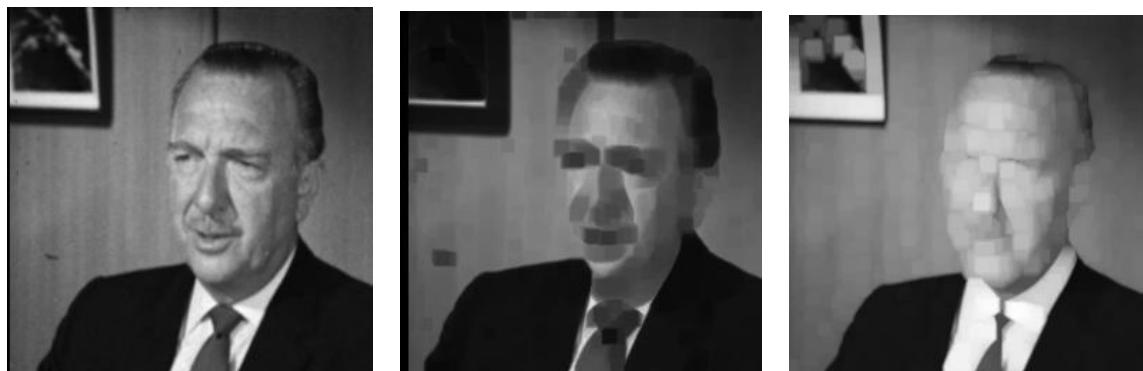


Рис. 8.12. Исходное изображение с примерами эрозии (в центре) и дилатации (справа) плоским структурным элементом квадратной формы

Рассмотрим численный пример выполнения этих операций. Пусть плоский структурный элемент несимметричен и его форма задана единицами  $[0 \ 1 \ 1]$ , центр элемента – это его геометрический центр (выделен жирным шрифтом), а строка яркостей обрабатываемого изображения имеет значения

$$0 \ 0 \ 8 \ 8 \ 9 \ 0 \ 0.$$

Применяя дилатацию, находим максимум в окрестности каждого пикселя, с которым совмещаем центр элемента, симметричного заданному, его окрестность отмечена единицами  $[1 \ 1 \ 0]$ . В результате яркости  $0 \ 0 \ 8 \ 8 \ 9 \ 0 \ 0$  примут значения

$$0 \ 0 \ 8 \ 8 \ 9 \ 9 \ 0.$$

Применяя эрозию, находим минимум в окрестности каждого пикселя, с которым совмещаем центр элемента, окрестность отмечена единицами  $[0 \ 1 \ 1]$ . В результате яркости  $0 \ 0 \ 8 \ 8 \ 9 \ 0 \ 0$  примут значения

$$0 \ 0 \ 0 \ 8 \ 8 \ 0 \ 0.$$

# Глава 9. Обработка цветных изображений

Существуют два основных типа обработки цветных изображений: цветовые компоненты обрабатываются по отдельности или цвет рассматривается как единый вектор в трехмерном пространстве.

В методах первого типа фактически выполняется обработка R, G, B цветов как отдельных полутооновых изображений. Однако в результате такой независимой обработки (например, растяжения контраста) трех цветов могут возникнуть неестественные для исходного изображения оттенки.

В методах второго типа изменения в цветовые составляющие вносятся пропорционально их «вкладу» в каждый пиксель. Рассмотрим некоторые методы улучшения цветных изображений.

## 9.1. Коррекция цвета на изображениях

*Изменение цветового баланса изображения.* Цель этого метода – компенсация неверной настройки цветовосприятия камеры или цветного освещения сцены.

Предполагается, что пользователь может интерактивно указать пиксели (или области) с нужным опорным цветом, например белый цвет – цвет облаков на фотографии, либо источником «правильных» цветов может быть фотография этой же (или похожей) сцены с хорошо переданными цветами (рис. 9.1).



Рис. 9.1. Коррекция изображения с опорным цветом: а) изображение с плохим цветовым балансом; б) откорректированное изображение

Преобразовать цвета по каждому из каналов можно по формулам

$$R = R \frac{R_{real}}{R_{orig}}, \quad G = G \frac{G_{real}}{G_{orig}}, \quad B = B \frac{B_{real}}{B_{orig}},$$

где индекс *orig* означает корректируемый цвет пикселя на исходном изображении; *real* – желаемый цвет.

*Формирование идеальных бликов.* Предполагается, что наиболее яркие области изображения являются бликами на поверхностях. В случае идеальных отражателей (зеркало) цвет блика должен быть равен цвету освещения, т. е. белому. Если значения цвета в бликах меньше допустимого максимума (обычно 255), то все цвета на изображении можно скорректировать по следующему алгоритму.

#### Алгоритм коррекции цвета при идеальных бликах

Шаг 1. Найти максимумы значений в бликах по каждому из каналов:  $R_{\max}$ ,  $G_{\max}$ ,  $B_{\max}$ .

Шаг 2. Изменить значения цвета для каждого пикселя по формулам

$$R = R \frac{255}{R_{\max}}, \quad G = G \frac{255}{G_{\max}}, \quad B = B \frac{255}{B_{\max}}.$$

*Преобразование «Серый мир».* В основе лежит предположение, что сумма всех цветов на изображении естественной сцены имеет серый цвет. Пример коррекции изображения представлен на рис. 9.2.



Рис. 9.2. Изображение с плохим цветовым балансом (a); откорректированное преобразование «Серый мир» (б)

#### Алгоритм коррекции цвета преобразованием «Серый мир»

Шаг 1. Вычислить средние значения всех цветов по формулам

$$\bar{R} = \frac{1}{N} \sum R(x, y), \quad \bar{G} = \frac{1}{N} \sum G(x, y), \quad \bar{B} = \frac{1}{N} \sum B(x, y), \quad Avg = \frac{\bar{R} + \bar{G} + \bar{B}}{3}.$$

Шаг 2. Скорректировать значения цветов согласно коэффициентам:

$$R' = R \frac{Avg}{\bar{R}}, \quad G' = G \frac{Avg}{\bar{G}}, \quad B' = B \frac{Avg}{\bar{B}},$$

где  $R'$ ,  $G'$ ,  $B'$  – новые значения цвета по каждому из каналов.

*Линейное растяжение контраста цветного изображения.* Суть этого метода – линейно растянуть интенсивности по каждому цветовому каналу на весь динамический диапазон. Этот метод используется в пакете Photoshop в функции autolevels.

#### Алгоритм линейного растяжения контраста

**Шаг 1.** Найти минимум и максимум изображения по каждому из каналов ( $R_{\min}, R_{\max}, G_{\min}, G_{\max}, B_{\min}, B_{\max}$ ).

**Шаг 2.** Преобразовать оттенки цвета каждого пикселя:

$$R' = (R - R_{\min}) \cdot \frac{(255 - 0)}{(R_{\max} - R_{\min})}, \quad G' = (G - G_{\min}) \cdot \frac{(255 - 0)}{(G_{\max} - G_{\min})},$$
$$B' = (B - B_{\min}) \cdot \frac{(255 - 0)}{(B_{\max} - B_{\min})}.$$

## 9.2. Автоматическое вычисление порога бинаризации цветного изображения

В некоторых задачах требуется преобразовать цветное изображение в бинарное (например, подобно факсимильному аппарату).

Эвристический выбор порога бинаризации  $T$ , например, на основании визуального изучения гистограммы, требует интерактивного вмешательства пользователя, что невозможно при автоматической обработке множества изображений. На сегодняшний день задача автоматического вычисления порога бинаризации  $T$ , при котором можно получить высокое качество и скорость бинаризации изображений разных типов при любых условиях съемки, остается нерешенной.

Существует несколько способов автоматической подстройки значения порога  $T$ . Для схематических и картографических изображений используется *метод Бернсена*, где константа должна подбираться интерактивно.

*Метод Эйквеля* часто применяется для обработки четких и контрастных изображений. Согласно данному методу изображение обрабатывается с помощью двух концентрических окон. При обработке тонких пересекающихся линий могут возникать разрывы, поэтому данный метод рекомендуется применять исключительно для толстых линий и крупных объектов.

*Методом Яновица и Брукштейна* лучше всего обрабатывать изображения, которым свойственна яркостная зональная неравномерность, когда одни и те же объекты изображения в разных частях имеют значительные различия яркости. К недостаткам данного метода следует отнести крайне медленную скорость работы.

Эффективным методом автоматического выбора порога бинаризации является *метод Отсю*, который использует гистограмму распределения значений яркости растрового изображения. Этот метод дает хороший результат разделения объектов и однородного фона на полутоно-вых изображениях. К недостаткам можно отнести размытие и потерю тонких линий, «слипание» объектов в местах их пересечений, а также итерационный процесс вычисления значения порога  $T$ .

При полной автоматизации и адаптации к разным условиям получения изображений при вычислении порога  $T$  рекомендуется метод, который называется *алгоритмом бинаризации k-средних*. Опишем его суть.

Пусть задана некоторая начальная величина порога  $T$ , она может быть любой, тогда

$$\forall x \in 1 \dots M-2, \forall y \in 1 \dots N-2, \exists B(x, y) = \begin{cases} 1, & \nabla F \geq T, \\ 0, & \nabla F < T, \end{cases} \quad (9.1)$$

где  $M, N$  – соответствующие значения ширины и высоты изображения;  $B(x, y)$  – бинарное изображение (функция яркости, принимающая значения 0 или 1);  $\nabla F$  – модуль градиента изображения;

$$\begin{aligned} \forall B(x, y) = 1, \quad H_i &= R(x, y) + G(x, y) + B(x, y), \quad i \in 1 \dots \alpha, \\ \forall B(x, y) \neq 1, \quad L_j &= R(x, y) + G(x, y) + B(x, y), \quad j \in 1 \dots \beta, \end{aligned} \quad (9.2)$$

где  $R(x, y), G(x, y), B(x, y)$  – соответствующие значения интенсивности красной, зеленой и голубой компонент на исходном цветном изображении;  $\alpha, \beta$  – соответствующие значения количества точек перепада и всех оставшихся.

Следовательно, значение порога  $T$  равно

$$T = \frac{1}{2} \cdot \left( \frac{\sum_{i=1}^{\alpha} H_i + \sum_{j=1}^{\beta} L_j}{(M-2) \cdot (N-2)} \right). \quad (9.3)$$

Из выражений (9.1)–(9.3) следует, что вначале проводится бинаризация изображения с использованием любой оценки порога  $T$ . В результате образуются две группы пикселов:  $H$  – из пикселов со значением  $\nabla F$ , большим либо равным  $T$ ;  $L$  – из пикселов со значением  $\nabla F$ , меньшим  $T$ . Далее находится половина среднего значения яркости пикселов полноцветного изображения по областям  $H$  и  $L$  соответственно. Полученный результат и будет являться требуемым глобальным порогом  $T$  для всего изображения, при использовании которого происходит бинаризация изображений в масштабе реального времени.

В оригинальном методе бинаризации  $k$ -средних наличие двух неизвестных факторов (начальной оценки значения порога, для вычисления которой предлагается ряд методов, и значения наперед заданного параметра  $N_0$ , которое служит ориентиром для количества итераций данного алгоритма) затрудняет его применение на практике, учитывая возникающую неопределенность с временем выполнения алгоритма. Смысл и основная часть оригинального метода не менялись, в модифицированном методе изменению (точнее, ликвидации) подлежали два вышеприведенных фактора. Также важной особенностью модифицированного метода, описанного выше, является переход от обработки исходного монохромного изображения к мультиспектральному. Экспериментально выяснилось, что яркости красной, зеленой и голубой компонент исходного изображения не следует умножать на коэффициенты, учитывающие восприятие последних в видимом диапазоне длин волн. В противном случае приведенный метод не будет обеспечивать нужное качество бинаризации изображения.

На рис. 9.3 представлен пример выполнения бинаризации изображения с применением фильтра Собела для вычисления градиента и автоматическим вычислением порога бинаризации  $T$ .



Рис. 9.3. Пример выполнения бинаризации: *а*) исходное цветное изображение; *б*) бинарное изображение, полученное из исходного вышеописанным методом

### 9.3. Повышение четкости цветного портрета

Повысить четкость цветного портрета можно разными способами. Один из них – преобразовать RGB-изображение в пространство Lab, что позволяет избежать появления цветных ореолов, которые возникают при значительном увеличении четкости цветного изображения. Пространство Lab точнее других цветовых пространств моделирует восприятие цвета человеком. Основное отличие цветовой модели Lab заключается в полном разделении цвета и контраста. В моделях RGB и CMYK все ка-

налы отвечают за цвет и контраст. В Lab весь контраст содержится в канале L, информация о цвете (точнее, о цветности) – в каналах a и b. Канал L можно представить как черно-белую версию цветного изображения. Каналы a и b являются чисто цветовыми. Если изображение не цветное, каналы a и b должны быть серыми (точнее, на 50 % серыми). Чем дальше от этого уровня серого и ближе к белому или черному, тем красочнее становится изображение. Канал a рассматриваем как изображение с распределением тонов от 0 до 100 %. Величина 50 % – это ни пурпурный и ни зеленый цвета. Более светлые области означают приближение к пурпурному, а более темные – к зеленому. В канале b схема цветов другая: светло-серые области представляют желтый, а темно-серые – синий. Уровень 50 % означает нейтральный серый.

При обработке цветных изображений Lab часто является промежуточным представлением цвета при коррекции. Изображения преобразуются в это пространство для обработки, а по ее завершении конвертируются в пространство отображения (не обязательно RGB). Этим избегают появления цветных ореолов, поскольку каналы a и b, т. е. каналы, где находится информация о цвете, не задействованы.

#### Алгоритм повышения четкости портрета

Шаг 1. Открыть изображение и преобразовать его из RGB в Lab.

Шаг 2. К полутоновому изображению канала L применить фильтр типа нерезкого маскирования, получим  $L_{new}$ .

Шаг 3. Преобразовать изображение, представленное в  $L_{new}ab$ , в  $R_{new}G_{new}B_{new}$  (рис. 9.4).

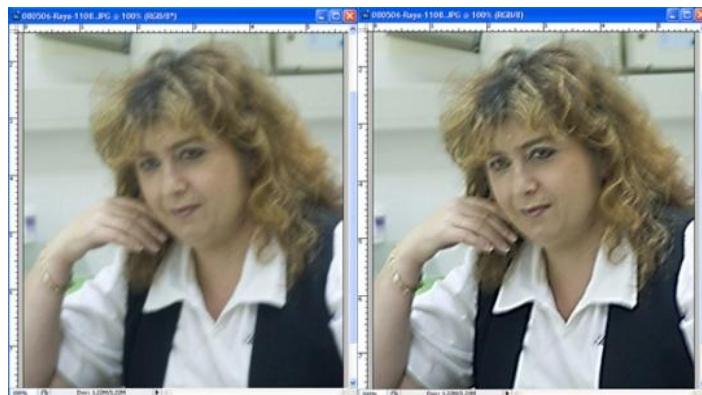


Рис. 9.4. Пример повышения четкости в пространстве Lab

Условия применения параметров максимального увеличения четкости:

– портрет не в фокусе, поэтому необходимо максимально увеличить его четкость, чтобы попытаться вернуть объекту снимка первоначальную резкость;

– на снимке много объектов с ярко выраженными границами (например, архитектурные сооружения, монеты, автомобили, какие-либо механизмы и др.).

Для увеличения резкости в пакете Photoshop используется фильтр UnsharpMask с параметрами: Amount (величина) – 65 %, Radius (радиус) – 4 и Threshold (порог) – 3. Данные параметры дают максимальное увеличение четкости для сильно расфокусированных изображений. Причем фильтр UnsharpMask с одними и теми же параметрами применяется дважды.



Рис. 9.5. Пример повышения четкости путем изменения только канала R

Описанный алгоритм часто применяется при обработке сделанных цифровой камерой в неблагоприятных условиях и отсканированных снимков. При применении этого алгоритма для увеличения четкости портрета, сделанного крупным планом, часто появляются на лице отчетливо видимые поры, морщины и др. Если увеличить четкость всех компонент изображения RGB или же четкость только канала L, используя UnsharpMask, то текстура кожи будет выглядеть грубее, чем на исходном изображении. В этом случае можно выполнить обработку в пространстве RGB, применив к каналу Red (красный) фильтр UnsharpMask, т. е. изменить четкость только одного канала Red. Остальные использовать без изменений. Красный канал, как правило, содержит наименьшее количество мелких деталей изображения, поэтому применение фильтра UnsharpMask к одному этому каналу позволяет повысить четкость только нужных областей лица (глаз, губ и т. д.), а четкость текстуры кожи остается практически неизменной и гладкой (рис. 9.5). Если вместо Red использовать канал Blue, то увеличится интенсивность текстуры кожи.

Незначительное увеличение четкости используется для обработки мягких снимков, т. е. сделанных на природе (например, цветов, котят, радуги, людей и т. п.). В этом случае следует использовать параметры фильтра UnsharpMask (Amount – 150 %, Radius – 1 и Threshold – 10).

Для увеличения четкости веб-изображений используют значения Amount – 400 %, Radius – 0,3 и Threshold – 0. Изображение получается мягким и немного размытым. Перед размещением его в сети Интернет разрешение высококачественного снимка уменьшается с 300 до 72 dpi.

Если интенсивность эффекта слишком велика, можно уменьшить значение параметра Amount до 200 %. Значение параметра Amount 400 % используется для сильно расфокусированных снимков.

## Глава 10. Сегментация изображений

Часто изображение представляется для анализа в виде совокупности однородных участков. Количество различных типов (или классов) участков обычно невелико. Все изображение может быть разделено на непересекающиеся области, каждая из которых представлена цветом или текстурой одного типа (например, море, песок и растительность на фотографии). При анализе таких изображений важным является выделение однородных областей и определение их геометрических характеристик. Процесс разделения изображения на однородные области называется *сегментацией*. Отметим, что похожие по свойствам области относятся к одному классу, но могут располагаться в разных участках изображения.

Цель сегментации – разбиение всего изображения на однородные в определенном смысле области. Обозначим через  $R$  область, занимаемую всем изображением. Сегментацию можно рассматривать как процесс разбиения  $R$  на  $n$  подобластей:  $R_1, R_2, \dots, R_n$ , при этом должны выполняться условия:

- a)  $\bigcup_{i=1}^n R_i = R;$
- б)  $R_i$  является связной областью,  $i = 1, 2, \dots, n$ ;
- в)  $R_i \cap R_j = \emptyset$  для всех  $i, j = 1, 2, \dots, n; i \neq j$ ;
- г)  $P(R_i) = \text{ИСТИНА}$  для  $i = 1, 2, \dots, n$ ;
- д)  $P(R_i \cup R_j) = \text{ЛОЖЬ}$  для  $i \neq j$ .

Согласно данным условиям: а) сегментирована должна быть вся область изображения; б) пиксели одной области связные; в) области не должны пересекаться; г) пиксели одного класса однородности; д) разные классы различимы.

Количество признаков, которыми могут отличаться друг от друга области различных типов, может быть различно. Большое разнообразие в задачу сегментации вносит обработка цветных изображений. Одним из важнейших признаков для сегментации в этом случае служит цвет, который дополняет совокупность характеристик, применяемых при обработке полутоновых изображений.

Единого, или универсального, подхода к решению задачи сегментации нет. Существует множество алгоритмов сегментации. Каждый из них предназначен для работы с определенным классом изображений и решения конкретных задач. Главная идея большей части алгоритмов заключается в нахождении соответствий или различий в характеристиках

пикселов. Характеристиками могут быть их взаимное расположение, яркость, цвет и текстура.

В большинстве случаев используются методы сегментации на основе пороговой обработки, наращивания областей, разделения и слияния областей, морфологического водораздела, анализа текстур.

## 10.1. Сегментация изображений по яркости на основе пороговой обработки

*Вычисление порогов по гистограмме.* Пороговая обработка – самый простой метод сегментации, ориентированный на обработку изображений, отдельные однородные участки которых различаются средней яркостью. Простейшим вариантом сегментации является бинаризация изображения, т. е. сегментация на два типа однородных участков. При этом преобразование каждого пикселя исходного изображения  $x(i_1, i_2)$  в выходное  $y(i_1, i_2)$  выполняется по правилу

$$y(i_1, i_2) = \begin{cases} y_0, & \text{при } x(i_1, i_2) \leq T; \\ y_1, & \text{при } x(i_1, i_2) > T, \end{cases}$$

где  $T$  – единственный параметр обработки, называемый порогом. Уровни выходной яркости  $y_0$  и  $y_1$  могут быть произвольными, они лишь выполняют функции меток, при помощи которых осуществляется разметка получаемой маски – отнесение ее точек к классам  $H_0$  или  $H_1$  соответственно. Часто их значения соответствуют уровням черного и белого. Если существует более двух классов, то при пороговой обработке должен быть задан набор порогов, отделяющих яркости различных классов друг от друга.

Центральным вопросом пороговой сегментации является определение значений порогов, которое должно выполняться автоматически. Предложено много методов вычисления порогов, в основном они базируются на анализе гистограммы исходного изображения.

Пусть  $w(x)$ ,  $0 \leq x \leq 255$ , – гистограмма полутонового изображения, т. е. яркости изменяются в диапазоне от 0 до 255. Идея гистограммного метода определения порога основывается на предположениях о том, что распределения вероятностей для каждого класса унимодальны (т. е. содержат по одному доминирующему пику), а граничных пикселов, разделяющих участки разных классов на изображении, гораздо меньше, чем пикселов принадлежащих обоим классам. Этим предположениям соответствует гистограмма, которая имеет многомодальный характер. Отдельные моды (холмы на гистограмме) соответствуют различным классам, а разделяющие их впадины – малочисленным по количеству точек границ. Пороги сегментации находятся при этом по положению впадин.

Рис. 10.1 иллюстрирует сказанное выше применительно к сегментации на два класса.

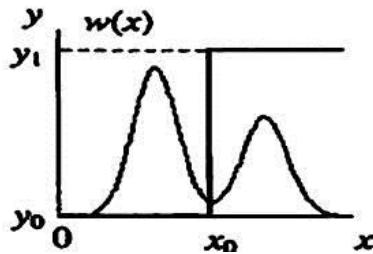


Рис. 10.1. Определение порога по гистограмме при сегментации по яркости

На практике воспользоваться такими упрощенными соображениями для выбора порога удается крайне редко, так как реальные гистограммы обычно имеют множество мелких пиков и сильно изрезаны. В связи с этим применяется сглаживание гистограммы как одномерного массива данных любым низкочастотным фильтром, описанным ранее. Вторым препятствием является то, что границы между однородными участками на изображении бывают размытыми, вследствие чего уровень гистограммы в тех ее частях, которые отображают точки границы, возрастает. Это приводит к уменьшению провалов в гистограмме или даже их исчезновению.

Если на темном фоне присутствуют светлые и белые предметы, гистограмма будет иметь три пика. Сегментацию можно выполнить с помощью двух порогов. Аналогично выполняется сегментация по цвету. Вначале строятся гистограммы отдельных цветов или исходное изображение преобразуется в другое цветовое представление, например HSI, Lab. Затем выполняется сегментация по  $I$ ,  $L$  или  $ab$ .

Алгоритмы, адаптирующие свое поведение, основываясь на статистиках обрабатываемого изображения, обладают большей устойчивостью к изменению характеристик изображения.

#### Алгоритм сегментации методом $k$ -средних

**Шаг 1.** Загрузить изображение, задать число классов  $k$ , на которое нужно разбить изображение. Определить признаки, по которым выполняется сегментация, например для полутоновых изображений можно использовать яркость. Все пиксели рассматриваются как набор векторов  $x_i$ ,  $i = 1, \dots, p$ .

**Шаг 2.** Определить  $k$  векторов  $m_j$ ,  $j = 1, \dots, k$ , которые объявляются начальными центрами кластеров. Для этого выбрать значения  $m_j$ ,  $j = 1, \dots, k$  (можно случайным образом или вычислить).

**Шаг 3.** Обновить значения средних векторов  $m_j$ ,  $j = 1, \dots, k$  (центров кластеров). Для этого вычислить расстояние от каждого  $x_i$ ,  $i = 1, \dots, p$  до каждого  $m_j$ ,  $j = 1, \dots, k$ . Отнести (приписать) каждый  $x_i$  к кластеру  $j^*$ ,

расстояние до центра которого  $m_j*$  минимально. Пересчитать средние значения  $m_j$  по всем кластерам.

**Шаг 4.** Повторять шаги 2, 3 пока центры кластеров не перестанут изменяться.

**Сегментация по принципу Вебера.** Опишем алгоритм сегментации полутоновых изображений, основанный на принципе Вебера. Этот принцип подразумевает, что различия уровней серого  $W(I)$ , которые могут быть замечены человеческим глазом, можно описать нелинейной функцией градаций серого  $I$ . Простейшей функцией Вебера является

$$W(I) = \begin{cases} 20 - \frac{12I}{88}, & 0 \leq I \leq 88; \\ 0,002(I - 88)^2 + 8, & 88 \leq I < 138; \\ \frac{7(I - 138)}{255 - 138} + 13, & 138 \leq I \leq 255. \end{cases}$$

Согласно принципу Вебера человеческий глаз едва ли различает отличия уровней серого между  $[I(n), I(n)+W(I(n))]$ , поэтому можно объединить уровни серого из диапазона  $[I(n), I(n)+W(I(n))]$ , заменив их одним значением яркости.

**Алгоритм нелинейного преобразования уровней серого для полутоновых изображений**

**Шаг 1.** Установить номер первого класса  $n=1$  и начальный уровень серого  $I(n) = 0$ .

**Шаг 2.** Вычислить значение  $W(I(n))$ , соответствующее яркости  $I(n)$  по формуле Вебера.

**Шаг 3.** В исходном изображении  $I$  установить значения яркости  $I(n)$  для всех пикселов, яркость которых находится в диапазоне  $[I(n), I(n)+W(I(n))]$ .

**Шаг 4.** Найти пиксели, значение яркости которых выше  $G = I(n) + W(I(n)) + 1$ . Если такие пиксели есть, увеличить номер класса  $n = n + 1$ ,  $I(n) = G$ , перейти к шагу 2. Если таких нет, закончить работу. Изображение сегментировано на  $n$  классов, каждый класс показан яркостью  $W(I(n))$ .

Данный способ сегментации можно реализовать, построив таблицу LUT, вычислив новые яркости с помощью функции Вебера и заменив исходные значения яркости найденными номерами классов одномоментно.

При сегментации цветных изображений можно использовать статистические характеристики в окрестности пикселя: отклонение каждой из цветовых компонент от среднего значения по окрестности; дисперсию каждой цветовой компоненты в окрестности пикселя и т. д.

Для простого случая разбиения по яркости можно использовать рекурсивный алгоритм наращивания областей (Region Growing).

**Алгоритм сегментации Вежневца.** Вежневец разработал следующий итеративный алгоритм сегментации.

Шаг 1. Начать обход изображения из левого верхнего угла. Объявить левый верхний пиксель изображения новым классом  $C_1$ .

Для пикселов первой строки вычислить отклонение от класса левого пикселя и сравнить с заданным порогом. Если оно меньше порога, добавить пиксель к классу соседа, иначе – завести новый класс  $C_{i+1}$ . Приналежность к классу можно отметить номером этого класса в матрице такого же размера, как, например, матрица, соответствующая компоненте R (красный цвет).

Шаг 2. Первый пиксель каждой следующей строки сравнить с классом верхнего соседнего пикселя. Далее сравнивать пиксель с классами двух соседей: левого и верхнего.

Если отклонение от обоих сравниваемых классов больше порога, то завести новый класс, если отклонение больше только для одного класса, то добавить пиксель к тому классу, отклонение от которого меньше порога.

В случае если отклонение допустимо для обоих классов, возможны два варианта:

$L(g(C_i) - g(C_j)) < \delta$  – объединить эти два класса (если это не один и тот же класс) и добавить к объединенному классу текущий пиксель;

$L(g(C_i) - g(C_j)) > \delta$  – добавить пиксель к тому из двух классов, от которого отклонение минимально.

В качестве меры  $L$  можно использовать любую функцию расстояния, например разность в RGB-пространстве.



Рис. 10.2. Исходное изображение (а) и результат его сегментации при  $\delta = 50$  (б)

Отметим, что алгоритм использует четырехсвязность пикселов, но возможна реализация и для восьмисвязных областей. Результат сегментации представлен на рис. 10.2.

**Сегментация по цвету кожи.** Использование цвета кожи как свойства лица при нахождении лиц на изображениях хорошо зарекомендовало себя на предварительном этапе в процессе распознавания людей, так как цвет кожи не зависит от ориентации лица, анализ цвета пикселов вычислительно эффективен. В литературе описано много подходов к сегментации цветных изображений с целью выделения на фотографиях областей, соответствующих цвету кожи людей. Далее в найденных областях выполняется детекция лица человека и его распознавание.

В качестве примера выполним сегментацию изображения по цвету кожи. Решение данной задачи состоит в нахождении критерия оценки близости цвета каждого пикселя к оттенку кожи. Определить понятие «оттенок кожи» математически невозможно, так как оно основано на человеческом восприятии цвета, меняется при изменении освещения, отличается у разных рас. Тем не менее существует ряд эмпирических подходов к приближенному решению этой задачи. Их суть заключается в описании некоторой модели распределения цвета, с помощью которой и оценивается сходство с цветом кожи. Разработка модели цвета кожи разбивается на три этапа:

1) накопление тренировочных данных, используя изображения, на которых указывают области «кожа» и «не кожа». По этим данным накапливается статистика оттенков кожи (иногда гистограмма оттенков, к коже не относящихся, т. е. «не кожа»);

2) обработку полученной статистики и выбор параметров модели цвета кожи для последующего использования; выбор критериев оценки принадлежности пикселов к области «кожа»;

3) обработку изображений с использованием полученных критериев.

Простейшим (но не всегда дающими корректные результаты) вариантом такой сегментации является разделение цветовых моделей RGB или других по порогу. Рассмотрим пороговые критерии, которые в качестве модели цвета кожи используют многогранник, заданный в определенном цветовом пространстве, т. е. задается набор правил для контекстно-независимой сегментации. Будем считать, что цвет пикселя ( $R, G, B$ ) причисляется к области «кожа», если выполнены следующие условия:

$$R > 95, \text{ и } G > 40, \text{ и } B < 20, \text{ и } \max\{R, G, B\} - \min\{R, G, B\} > 15, \\ \text{и } |R - G| > 15, \text{ и } R > G \& R > B$$

или

$$R > 220, \text{ и } G > 210, \text{ и } B > 170, \text{ и } |R - G| \leq 15, \text{ и } R > B, \text{ и } G > B.$$

Другой вариант:

$$\left( \frac{r}{g} > 1,185 \right), \text{ и } \left( \frac{r \cdot b}{(r + g + b)^2} > 0,107 \right), \text{ и } \left( \frac{r \cdot g}{(r + g + b)^2} > 0,112 \right),$$

где  $r = R/(R + G + B)$ ,  $g = G/(R + G + B)$ ,  $b = B/(R + G + B)$ .

После сегментации по цвету кожи (рис. 10.3) необходимо выполнить морфологическую или медианную фильтрацию.



Рис. 10.3. Пример сегментации по цвету кожи

Простота реализации – это несомненное преимущество пороговых подходов к сегментации, однако есть и недостатки. Необходимо подбирать цветовое пространство и правила эмпирически, при этом возможна неточная сегментация кожи, трудно подобрать единые критерии к людям разных рас.

## 10.2. Статистический подход к текстурной сегментации

В цифровой обработке изображений для описания текстуры применяются два основных подхода: статистический и структурный [15]. Статистические методы позволяют охарактеризовать текстуру области как гладкую, грубую и зернистую. Структурные методы определяют и описывают взаимное расположение простейших повторяющихся элементов изображения, например отрезков параллельных линий, проходящих с постоянным шагом, клеток на шахматной доске.

Рассмотрим статистический подход к сегментации на примере изображения, содержащего два типа областей, представленных разными текстурами (рис. 10.4). Результатом сегментации является разбиение изображения на водную поверхность и сушу. Это невозможно сделать методами бинаризации, только путем анализа параметров текстуры в окрестности каждого пикселя.



Рис. 10.4. Изображение с областями текстуры разных типов, соответствующих суше и воде

### Алгоритм сегментации посредством текстурного анализа

**Шаг 1.** Считать изображение.

**Шаг 2.** Определить параметры текстуры на изображении. Предполагая, что яркость в пикселях изображения – это случайная величина  $z$ , ей соответствует вероятность распределения  $p(z_i)$ ,  $i = 0, 1, 2, \dots, L - 1$ , взятая из гистограммы ( $L$  – число уровней яркости).

Центральный момент порядка  $n$  случайной величины  $z$  равен

$$\mu_n(z) = \sum_{i=0}^{L-1} (z_i - m)^n p(z_i), \quad (10.1)$$

где  $m$  – среднее значение  $z$  (средняя яркость изображения),

$$m = \sum_{i=0}^{L-1} z_i p(z_i). \quad (10.2)$$

Из формулы (10.1) следует, что  $\mu_0 = 1$  и  $\mu_1 = 0$ . Для описания текстуры важен второй момент, т. е. дисперсия  $\sigma^2(z) = \mu_2(z)$ . Она является мерой яркостного контраста, что можно использовать для вычисления признаков относительной гладкости. Например,

$$R = 1 - \frac{1}{1 + \sigma^2(z)} \quad (10.3)$$

равно нулю для областей постоянной яркости (где дисперсия нулевая) и приближается к единице для больших значений  $\sigma^2(z)$ . Поскольку для полутоновых изображений с яркостью от 0 до 255 значения дисперсии велики, целесообразно нормировать дисперсию до интервала [0,1]. Для этого необходимо поделить  $\sigma^2(z)$  на  $(L - 1)^2$ . Значение стандартного отклонения

$$s = \sigma(z) \quad (10.4)$$

также часто используется в качестве характеристики текстуры. Третий момент является характеристикой симметрии гистограммы:

$$\mu_3(z) = \sum_{i=0}^{L-1} (z_i - m)^3 p(z_i). \quad (10.5)$$

Для оценки текстурных особенностей используется функция энтропии, определяющая разброс по яркости соседних пикселов:

$$e = -\sum_{i=0}^{L-1} p(z_i) \log_2 p(z_i), \quad (10.6)$$

где  $p(z_i)$  – вероятность текущей яркости в окрестности точки;  $L$  – количество уровней яркости;  $e$  – значение энтропии в текущей точке.

Для описания текстуры также используется мера однородности, оценивающая равномерность гистограммы:

$$U = \sum_{i=0}^{L-1} p^2(z_i). \quad (10.7)$$

В табл. 10.1 приведены значения описанных выше характеристик, подобранные для гладкой, грубой и периодической текстур.

Таблица 10.1  
Параметры трех различных типов текстур

Текстура	Среднее	Стандартное отклонение	$R$ (норм)	Третий момент	Однородность	Энтропия
Гладкая	82,64	11,79	0,002	-0,105	0,026	5,434
Грубая	143,56	74,63	0,0079	-0,151	0,005	7,783
Периодическая	99,72	33,73	0,017	0,750	0,013	6,674

**Шаг 3.** Создать маску для выделения более крупной текстуры. Пусть на изображении имеются текстуры двух типов: крупная и мелкая (более зернистая). Мелкая соответствует водной зоне. Для отделения одной области от другой создать маску, которая удаляет маленькие объекты. Для этого использовать функцию определения связного множества пикселов на бинарном изображении и вычислить площади полученных объектов. Использовать тип связности восемь. Если цвет соседей совпадает, то они принадлежат одному объекту, в противном случае – к различным. Все объекты с площадью меньше заданной величины  $S$  удалить.

Результат текстурной фильтрации на базе вычисления энтропии по формуле (10.6) в окне 9x9 представлен на рис. 10.5. Бинарная маска водной поверхности после удаления объектов малой площади и результат сегментации суши представлены на рис. 10.6.

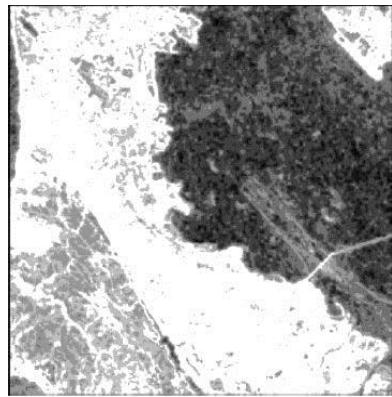


Рис. 10.5. Результат текстурной фильтрации на базе вычисления энтропии по формуле (10.6) в окне 9x9. Области двух текстур показаны темными (вода) и светлыми (суша) оттенками

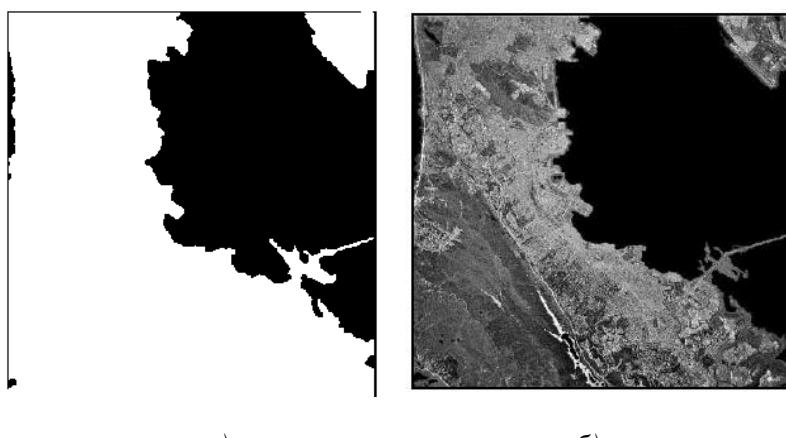


Рис. 10.6. Мaska водной поверхности после удаления объектов малой площади (а) и результат сегментации суши (б)

Для сегментации двух указанных типов текстур можно использовать разные текстурные признаки, например наборы признаков, вычисляемых по формулам (10.2)–(10.4), (10.3)–(10.5), (10.2)–(10.6), (10.4)–(10.7), (10.6)–(10.7), (10.5)–(10.6), (10.2)–(10.3), (10.3)–(10.7), (10.4)–(10.6) и др.

### 10.3. Сегментация методом морфологического водораздела

В методе морфологического водораздела полутоновое изображение рассматривается как цифровая модель местности, где значения яркости трактуются как высоты относительно некоторого уровня, т. е. изображение – это матрица высот. Если на такую местность льет дождь, образуется множество бассейнов. Вода заполняет маленькие бассейны, затем из переполненных бассейнов выливается и бассейны объединяются в более крупные согласно высотам уровня воды. Места объединения бассейнов отмечаются как линии водораздела. В итоге вся местность может быть затоплена. Результат сегментации зависит от момента прекращения поступления

воды. Если процесс остановить рано, изображение будет сегментировано на мелкие области, если поздно – на очень крупные.

В такой интерпретации все пиксели делятся на три типа: локальные минимумы; находящиеся на склоне, т. е. те, с которых вода скатывается в один и тот же локальный минимум; локальные максимумы, т. е. те, с которых вода скатывается более чем в один минимум.

При сегментации с помощью данного метода нужно определить водосборные бассейны и линии водораздела на изображении путем обработки локальных областей в зависимости от их яркостных характеристик.

При реализации метода водораздела вычисляется функция сегментации (это касается изображений, где объекты размещены в темных областях и являются трудно различимыми); вычисляются маркеры переднего плана изображений на основании анализа связности пикселов каждого объекта; вычисляются фоновые маркеры, представляющие собой пиксели, которые не являются частями объектов; модифицируется функция сегментации на основании значений расположения маркеров фона и маркеров переднего плана.

Одним из важнейших шагов сегментации по водоразделам является выделение на фоне изображения однородных по яркости объектов (в виде пятен). Области, характеризующиеся малыми вариациями яркости, имеют малые значения градиента. Поэтому на практике метод сегментации по водоразделам обычно применяется не к самому изображению, а к его градиентному представлению.

## **Выводы по части II**

Общей теории обработки изображений не существует. Какой метод использовать при решении конкретной практической задачи, зависит от знаний и опыта исполнителя. Если решение задачи предполагает регулярно повторяющуюся последовательность действий, то меняются только исходные данные, нарабатываются определенные алгоритмы обработки и их параметры. Примером может служить ретушь фотографий для документов. Это пользовательский вариант обработки изображений. Для него достаточно освоить один-два пакета обработки и следить за их обновлениями, осваивая новые возможности. Такая обработка сводится к применению нескольких методик, т. е. последовательности определенных действий.

Если исходные изображения относятся к разным классам, а классы регулярно меняются, – это исследовательский тип обработки изображений. В этом случае разносторонние знания методов обработки и умение оценивать полученные результаты подскажут правильную методику обработки. В обоих случаях знание основных методов, их сути и параметров ускоряет достижение желаемого результата.

# ЧАСТЬ III. ПРИМЕРЫ ПРИМЕНЕНИЯ ЦИФРОВОЙ ОБРАБОТКИ ИЗОБРАЖЕНИЙ

## Глава 11. Решение прикладных задач

При решении реальных задач обработки изображений, как правило, недостаточно одного метода. Требуется последовательное, а нередко итерационное, применение нескольких методов. Тогда возникает проблема выбора этих методов и их конкретных параметров. Рассмотрим примеры решения реальных задач.

### 11.1. Повышение контраста рентгеновских изображений

Рентгеновские изображения часто имеют слабый контраст, плохую проработку деталей, и не всегда можно сделать повторный снимок. Возникает необходимость цифровой обработки для улучшения таких снимков. Например, исходное изображение низкоконтрастное, что затрудняет анализ мелких деталей (рис. 11.1).

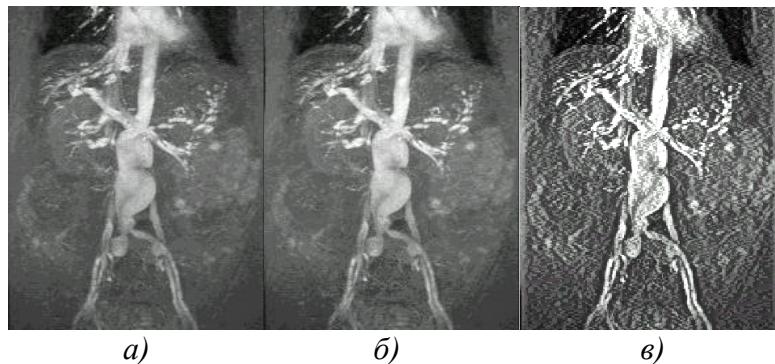


Рис. 11.1. Повышение контраста рентгеновского изображения: *а*) полуточное изображение с низким контрастом; *б*) после линейного растяжения контраста; *в*) после локального контрастирования

Для улучшения снимка можно сначала применить линейное растяжение гистограммы исходного изображения на максимальный диапазон  $[0, \dots, 255]$  по формуле (6.1) (рис. 11.1, *б*). Затем выполнить локальное контрастирование изображения. Опишем алгоритм [5]:

Шаг 1. Выполнить усредняющую фильтрацию изображения  $L_c$  ядром фильтра  $3 \times 3$  по формуле (7.2). Обозначить результат  $L_{yc}$ .

Шаг 2. Вычислить значения локального контраста каждого пикселя  $(i, j)$ :

$$C = / (L - L_{yc}) / (L + L_{yc})^{0,55}.$$

**Шаг 3.** Для каждого пикселя  $(i, j)$ :

если  $L(i, j) > L_{yc}(i, j)$ ,  $L_{pe3}(i, j) = L_{yc}(i, j) \cdot (1 + C(i, j)) / (1 - C(i, j))$ ;  
иначе  $L_{pe3}(i, j) = L_{yc}(i, j) \cdot (1 - C(i, j)) / (1 + C(i, j))$ .

Результирующее изображение  $L_{pe3}$  визуально контрастнее исходного рентгеновского изображения (рис. 11.1, в).

## 11.2. Морфологическая фильтрация бинарных изображений

Морфологические операции можно использовать для построения фильтров, похожих по принципу работы на пространственные фильтры. На рис. 11.2 показано бинарное изображение фрагмента отпечатка пальца, искаженное шумом, в виде светлых точек на темном фоне и темных на светлых полосах, представляющих отпечаток. Задача состоит в устранении шума при минимальном искажении формы и особенностей отпечатка. Для решения этой задачи можно применить морфологический фильтр, выполняющий вначале операцию размыкания, затем замыкания. На рис. 11.2, б представлен примитив, остальные части рисунка иллюстрируют последовательность шагов операции фильтрации. Шум в фоновой области изображения полностью устраняется на этапе эрозии (рис. 11.2, в), входящем в состав операции размыкания, поскольку в данном примере физические размеры всех составляющих шума меньше размеров примитива. Но шумовые компоненты в виде темных пятен на отпечатке увеличились в размерах. Такое расширение нейтрализуется путем дилатации, что приводит к результату, показанному на рис. 11.2, г. Заметно, что шумовые составляющие на полосах отпечатка пальца уменьшились в размерах или полностью исчезли. Две описанные выше операции (эрозия и дилатация) вместе составляют операцию размыкания исходного изображения по примитиву, представленному на рис. 11.2, б.

Из рис. 11.2, г видно, что результирующий эффект от размыкания состоит в устраниении практически всех шумовых составляющих как фоновой области, так и на самом отпечатке. Однако в результате этой операции появились ранее отсутствовавшие пропуски на полосах отпечатка. Чтобы преодолеть этот нежелательный эффект, можно применить к результату размыкания операцию дилатации, как показано на рис. 11.2, д. В большинстве указанных промежутков целостность полос восстановилась, но полосы стали шире. Такое увеличение можно компенсировать эрозией.



$$\begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}$$

б)





Рис. 11.2. Повышение качества изображения путем морфологической фильтрации: *а*) отпечаток пальца; *б*) структурный элемент; *в*) изображение после операции эрозии; *г*) результат размыкания исходного изображения; *д*) дилатация результата размыкания; *е*) замыкание результата размыкания

Результат на рис. 11.2, *е* получен путем применения операции замыкания к ранее построенному размыканию исходного изображения, представленному на рис. 11.2, *г*. В конечном итоге отсутствуют шумовые точки, хотя некоторые из полос отпечатка пальца восстановлены не полностью, а содержат разрывы. Этого следовало ожидать, поскольку в описываемую процедуру не было заложено никаких условий для сохранения связности.

### **11.3. Коррекция неравномерного фона изображения и выделение объектов**

Дано изображение сцены, полученное при неравномерном освещении (рис. 11.3). Автоматический анализ объектов, представленных на нем, затруднен. Требуется выполнить коррекцию фона, сделав его однородным.

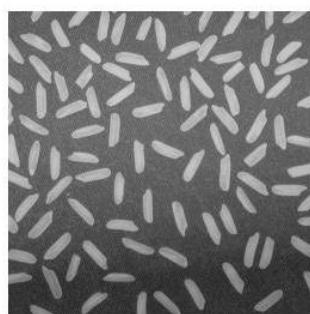


Рис. 11.3. Изображение с неравномерным фоном при неравномерном освещении сцены при регистрации

**Алгоритм коррекции фона**

Шаг 1. Загрузить изображение.

**Шаг 2.** Применить морфологическую операцию отмыкания для оценки фона (рис. 11.4).

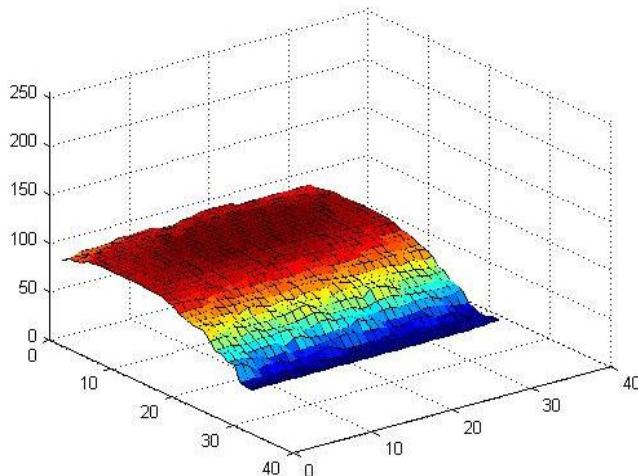


Рис. 11.4. Яркость фона исходного изображения, полученная в результате операции отмыкания с плоским структурным элементом радиуса 15

**Шаг 3.** Попиксельно вычесть яркость фона из исходного изображения. Выполнить линейное растяжение контраста полученного изображения (рис. 11.5).

**Шаг 4.** Выполнить бинаризацию изображения методом Отсу (рис. 11.5).

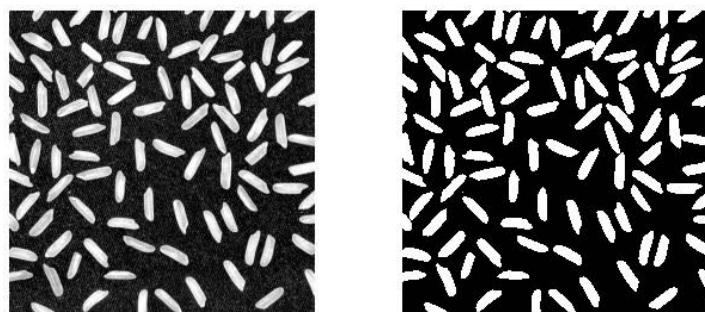


Рис. 11.5. Изображение после выполнения шага 3 (слева) и шага 4 (справа)

**Шаг 5.** Выполнить выделение объектов как связных компонент на бинарном изображении и отобразить их разным цветом (рис. 11.6).

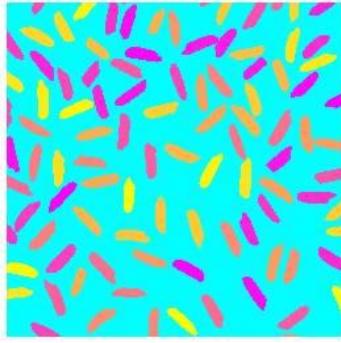


Рис. 11.6. Выделенные объекты

Шаг 6. Вычислить характеристики выделенных объектов.

#### 11.4. Повышение разрешения мультиспектрального изображения

Мультиспектральные спутниковые изображения представляют собой пакет изображений местности, зарегистрированных в диапазонах B, G, R, NIR (ближний инфракрасный) и Pan (панхроматический), причем Pan имеет разрешение выше предыдущих в два или четыре раза. Требуется повысить разрешение мультиспектральных изображений до панхроматического уровня и подготовить их к отображению в виде цветного изображения. Такая процедура называется слиянием с панхроматическим каналом (Pansharpening), она объединяет данные панхроматического изображения с многоканальным набором растровых изображений этой же местности более низкого разрешения. В результате получается композитное цветное изображение местности с таким же разрешением, как у панхроматического изображения. Например, для спутника IKONOS оно повышается в четыре раза (рис. 11.7). Данная процедура используется при отображении местности в Google.maps. Отметим, что масштабирование исходных изображений R, G, B даст размытое изображение.

Кратко опишем идеи, использованные в несложных методах решения данной задачи [23]. Самым простым является

##### Алгоритм Бровея

Шаг 1. Повысить разрешение изображений R, G, B до разрешения изображения Pan (увеличить до одинакового размера).

Шаг 2. Для каждого пикселя  $(i, j)$  спектральных изображений скорректировать значения яркости:

$$\begin{aligned}R_{out}(i, j) &= R(i, j) \cdot \text{Pan}(i, j) / (\text{B}(i, j) + \text{G}(i, j) + \text{R}(i, j)); \\G_{out}(i, j) &= G(i, j) \cdot \text{Pan}(i, j) / (\text{B}(i, j) + \text{G}(i, j) + \text{R}(i, j)); \\B_{out}(i, j) &= B(i, j) \cdot \text{Pan}(i, j) / (\text{B}(i, j) + \text{G}(i, j) + \text{R}(i, j)).\end{aligned}$$



Рис. 11.7. Процедура PanSharpening снимков спутника IKONOS

**Шаг 3.** Полученные спектральные изображения  $R_{out}$ ,  $G_{out}$ ,  $B_{out}$  объединить в виде цветного изображения RGB.

Описанный алгоритм быстр, но в некоторых пикселях искажает цветовое представление местности.

Если панхроматический канал покрывает диапазон ближнего инфракрасного канала, то вычитание взвешенного вклада инфракрасного диапазона из панхроматического канала уменьшает цветовые искажения. Формулы шага 2 с использованием весов DNF для каждого пикселя каждого канала приобретают вид

$$\begin{aligned}
 DNF(i, j) &= (\text{Pan}(i, j) - \text{NIR}_w \cdot \text{NIR}(i, j)) / (\text{R}_w \cdot \text{R}(i, j) + \text{G}_w \cdot \text{G}(i, j) + \text{B}_w \cdot \text{B}(i, j)); \\
 R_{out}(i, j) &= \text{R}(i, j) \cdot DNF(i, j); \\
 G_{out}(i, j) &= \text{G}(i, j) \cdot DNF(i, j); \\
 B_{out}(i, j) &= \text{B}(i, j) \cdot DNF(i, j); \\
 \text{NIR}_{out}(i, j) &= \text{NIR}(i, j) \cdot DNF(i, j),
 \end{aligned}$$

где Pan – панхроматический, R – красный, G – зеленый, B – синий, NIR – ближний инфракрасный каналы;  $\text{NIR}_w$ ,  $\text{R}_w$ ,  $\text{G}_w$ ,  $\text{B}_w$  – константы для разных каналов.

Еще один вариант алгоритма операции PanSharpening – преобразование из RGB в другое цветовое пространство, например в IHS. В нем можно корректировать яркостную компоненту I .

### Алгоритм Pansharpening в пространстве IHS

Шаг 1. Повысить разрешение изображений R, G, B до разрешения изображения Pan (увеличить до одинакового размера).

Шаг 2. Объединить их в цветное изображение RGB и преобразовать в пространство IHS. Выделить компоненту I.

Шаг 3. Построить гистограммы изображений Pan и компоненты I. Преобразовать форму гистограммы I в форму гистограммы Pan (см. разд. 6.5) и сформировать новое представление яркостной компоненты  $I_{new}$ .

Шаг 4. В IHS заменить яркостную компоненту на  $I_{new}$  и выполнить преобразование из представления  $I_{new}$  HS в  $R_{out}G_{out}B_{out}$  (см. приложение).

Последние шаги всех алгоритмов – выполнить переквантование значений яркости полученного цветного изображения в восьмибитовое представление (см. 2.1.2).

### 11.5. Детекция предупреждающих дорожных знаков

Необходимо автоматически найти на видеокадре предупреждающие дорожные знаки (рис. 11.8) со скоростью обработки не менее 10 кадров в секунду. Для этого нужно преобразовать цветовое пространство, удалить шумы, выделить области интереса и распознать тип знака. Опишем алгоритм решения данной задачи [4]:

Шаг 1. Преобразовать цветовое пространство.

Все предупреждающие дорожные знаки имеют общий признак – контур красного цвета (рис. 11.8). Следовательно, первоначальный поиск объектов интереса заключается в определении областей красного цвета.



Рис. 11.8. Предупреждающие дорожные знаки

Входное изображение представлено в цветовом пространстве RGB. Для детекции знаков по цвету целесообразнее перейти в другое цветовое пространство, например  $YC_bC_r$ . Для конвертации из RGB в  $YC_bC_r$  используется уравнение

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 65,481 & 128,553 & 24,966 \\ -37,797 & -74,203 & 112 \\ 112 & -93,786 & -18,214 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}.$$

Для сегментации изображения по красному цвету можно применять составляющую  $C_r$ , но она содержит шумы.

**Шаг 2.** Удалить шумы на изображении. Для этого следует выполнить фильтрацию изображения. Фильтрацию можно выполнять как на полутонастом представлении  $C_r$ , так и на бинарном.

Для получения бинарного изображения выполняется операция бинаризации изображения  $C_r$  (рис. 11.9).



Рис. 11.9. Бинарное представление составляющей  $C_r$  реального изображения с дорожным знаком (точки слева – шум)

Для удаления мелких объектов (рис. 11.10, а) в обоих случаях требуется использовать морфологические фильтры размыкания и замыкания.

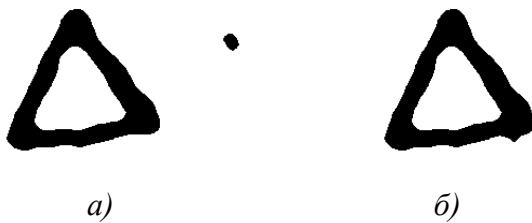


Рис. 11.10. Изображение дорожного знака после медианной фильтрации:  
а) с морфологической фильтрацией; б) без нее остаются мелкие нежелательные объекты (в центре)

Очевидно, что одного цветового признака для определения принадлежности выделенного объекта к классу дорожных знаков недостаточно, так как на изображении могут находиться другие объекты красного цвета (например, автомобили, рекламные щиты). Второй признак, который можно использовать для детекции дорожных знаков рассматриваемого класса, – форма правильного треугольника.

Для определения формы необходимо провести анализ найденных объектов независимо друг от друга. Сначала выделяется граница каждого объекта. Граница определяет местоположение объекта на изображении, но не определяет его форму. Для идентификации типа объекта необходимо определить углы между тремя прямыми, которые аппроксимируют его контур. С этой целью с каждой стороны объекта выбирается несколько граничных точек, которые аппроксимируются тремя прямыми (рис. 11.11).

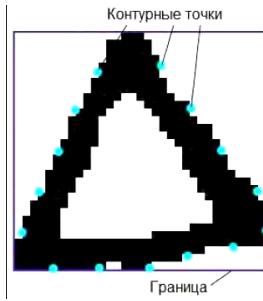


Рис. 11.11. Бинарное изображение с граничными точками

При условии, что прямая проходит через две заданные точки с координатами  $(x_1, y_1)$  и  $(x_2, y_2)$ , угол между двумя такими прямыми определяется по формулам

$$A = y_2 - y_1; \quad B = x_1 - x_2;$$

$$\cos\phi = \frac{|A_1A_2 + B_1B_2|}{\sqrt{A_1^2 + B_1^2} \sqrt{A_2^2 + B_2^2}}.$$

Объект считается распознанным как знак, если

$$60^\circ - \Delta\phi \leq \phi \leq 60^\circ + \Delta\phi,$$

где  $\phi$  – угол между прямыми;  $\Delta\phi$  – допустимое отклонение угла.

Далее выполняется распознавание найденного знака путем анализа его внутреннего содержания.

## 11.6. Автоматическое определение нефтяных пятен

Пятна нефтяных разливов на воде могут быть обнаружены путем изображений дистанционного зондирования из-за резкого контраста между областями пятен нефти и окружающей воды.

Один из подходов заключается в вычислении спектральных индексов на основе соотношения яркости изображений, зарегистрированных в синем и красном диапазонах электромагнитного спектра. При наличии пятен нефти на водной поверхности возникает процесс флуоресценции, при этом увеличивается отражение сигнала в красном диапазоне спектра.

Индекс флуоресценции (FI) вычисляется для всех пикселов мультиспектрального изображения на базе синего В и красного R диапазонов спектра по формуле, аналогичной вычислению контраста изображения [25]:

$$FI_{i,j} = (B_{i,j} - R_{i,j}) / (B_{i,j} + R_{i,j}).$$

Нефтяные пятна на воде меняют оптические свойства в зависимости от угла падения солнечных лучей и геометрии регистрации изображений спутниковыми датчиками. Пятна практически не видны на безбликовых изображениях, но могут быть заметны в виде довольно темных областей (из-за отрицательного контраста) при низких или умеренных солнечных бликах либо в виде довольно ярких областей (из-за положительного контраста) при сильных солнечных бликах.

## 11.7. Контроль отверстий в 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 [21]. Для получения цифрового изображения поперечной балки под транспортировочной линией были помещены осветители, а над линией – видеокамера. На изображении темные пиксели в области тени от балки имеют значения 1 и обозначают сталь, а пиксели в области светлых просверленных отверстий имеют значения 0. Количество отверстий вычисляется как четверть разности числа внешних и внутренних углов.

На рис. 11.12, в показаны три светлых отверстия (значения пикселов 0) на темном фоне (значения пикселов 1). Внешний угол – это окрестность  $2 \times 2$  соседних пикселов, содержащая три единицы, а внутренний угол – окрестность  $2 \times 2$  соседних пикселов с тремя нулевыми значениями. На рис. 11.12 показан пример обработки изображения из 7 строк и 16 столбцов.

### Алгоритм подсчета количества отверстий

Входные данные – бинарное изображение, выходные – количество найденных отверстий.

Обозначения:  $M$  – бинарное изображение; 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 четырехсвязной и все граничные пиксели изображения должны быть равны 1;  $e$  – счетчик внешних углов (три нуля и единица в окрестности  $2 \times 2$ );  $i$  – счетчик внутренних углов (три нуля и единица в окрестности  $2 \times 2$ ).

Шаг 1. Обработать все изображение  $M$  по две строки на каждом шаге.

Шаг 2. Вычислить количество внешних углов  $e$  в строке.

Шаг 3. Вычислить количество внутренних углов  $i$  в строке.

Шаг 4. Вычислить суммы углов и число отверстий  $(e - i)/4$ .

$$\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

a)

$$\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$$

b)



c)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
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

c)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	e	i
0	e			e					e		e		e		e		6	0
1									e	i							1	1
2	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

d)

Рис. 11.12. Пример подсчета отверстий на бинарном изображении: a) шаблоны окрестностей 2x2 пикселов, представляющие «внешние углы»; б) шаблоны окрестностей 2x2 пикселов, представляющие «внутренние углы»; в) три светлых отверстия на темном фоне; г) входное бинарное изображение; д) окрестности, соответствующие внешним *e* и внутренним *i* углам,  $(21 - 9)/4 = 3$  отверстия

## 11.8. Улучшение изображений в системах видеонаблюдения

Изображения, которые регистрируются системой видеонаблюдения, не всегда качественны для визуального восприятия, поскольку системы регистрируют изображения в различных погодных условиях, при разном освещении. Рассмотрим, как можно улучшить низкокачественные изображения в системах видеонаблюдения [5].

*Усиление контрастности.* Низкий контраст является одним из наиболее распространенных дефектов непрофессиональных фотографических и телевизионных изображений. Причины могут быть самые раз-

личные: плохое качество аппаратуры, внешние условия и т. п. Что касается систем видеонаблюдения, то в этом случае недостаточно высокое качество регистрируемых изображений объясняется условиями съемки (погодные условия, темное время суток).

Исходное изображение цветное, поэтому можно цветовые компоненты обрабатывать независимо. Выполним нерезкое маскирование путем свертки каждой компоненты с ядром фильтра:

$$\frac{1}{6} \begin{bmatrix} -1 & -4 & -1 \\ -4 & 26 & -4 \\ -1 & -4 & -1 \end{bmatrix}$$

Изображение становится более контрастным (рис. 11.13).



Рис. 11.13. Исходное (слева) и контрастированное (справа) изображения

*Коррекция динамического диапазона яркости.* При недостаточном освещении регистрируемые изображения имеют узкий динамический диапазон значений, поэтому коррекция диапазона в системах видеонаблюдения очень актуальна.

Выполним нелинейное растяжение гистограммы для трех цветовых компонент независимо по формуле (значение  $a$  подбирается экспериментально)

$$K_{stretch} = ((K - K_{min}) / (K_{max} - K_{min}))^a.$$

Затем выполним медианную фильтрацию. Фильтрация необходима, поскольку после контрастирования темных изображений они имеют зернистый вид. Результаты фильтрации и коррекции динамического диапазона представлены на рис. 11.14 и 11.15.



Рис. 11.14. Исходное изображение после нелинейного растяжения гистограммы контрастирования (слева) и после фильтрации (справа)

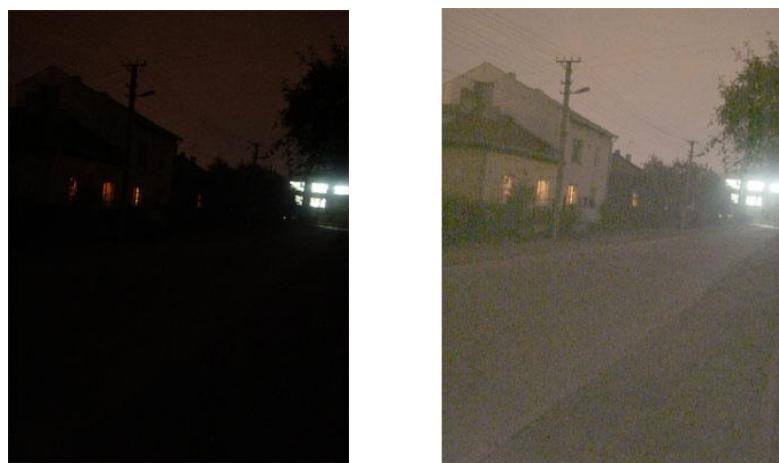


Рис. 11.15. Исходное изображение (слева) и обработанное при  $a = 0,25$  (справа)

## Глава 12. Программные средства обработки цифровых изображений

В данной главе не рассматриваются пакеты обработки изображений типа Adobe Photoshop. Назначение этого пакета – интерактивное редактирование изображений. Его коды закрыты, и вся обработка изображений выполняется нажатием комбинации клавиш и щелчком кнопкой мыши без описания алгоритмов преобразования, без понимания сути обработки изображений.

### 12.1. Использовать готовый код библиотеки или написать собственный?

Многие разработчики программных комплексов рано или поздно сталкивались с выбором: разработать что-то недостающее самому или приобрести готовое? Готовые коды, как правило, продаются в виде библиотек, но есть и открытые библиотеки.

Несомненно, если взять функцию из уже существующей библиотеки, разработка проекта будет закончена намного быстрее. Разработчик будет концентрироваться именно на задаче, а не на вспомогательной инфраструктуре. Однако библиотеки обычно пишутся для универсального применения, решения множества различных задач. Часто их код довольно громоздок для понимания и не всегда является оптимальным для решения конкретной задачи. Если реализуется собственное решение, то его можно написать оптимальным (согласно опыту) способом и именно под свою задачу.

При использовании библиотеки получится решение, оттестированное множеством пользователей. Если же все-таки будет найдена ошибка, то в Интернете можно найти советы людей, столкнувшихся с этой ошибкой. С другой стороны, если с ней никто не сталкивался или о ней никто не пишет, можно потратить много времени, чтобы разобраться, что и почему не работает. Поскольку код библиотеки может быть довольно громоздким и не всегда доступным.

Если весь код написан самостоятельно, значит разработчик хорошо в нем ориентируется и найти ошибку будет проще. Однако ошибок, скорее всего, будет значительно больше, так как тестировал этот код только разработчик. Если код пишется с нуля, то можно упустить из вида какой-нибудь редкий случай, который в библиотеке со временем уже выявили и закодировали.

Если над проектом работает команда и приходит новый сотрудник, есть вероятность того, что он уже работал с методами, реализованными в библиотеке (либо специалиста берут с условием, что он имеет соответствующий опыт). Если же используется свое собственное решение, то новому сотруднику придется разбираться в деталях, как оно работает. С другой стороны, написать самостоятельно код, который уже реализован в библиотеках, весьма непростая задача (особенно для начинающих). Для этого нужен высококвалифицированный специалист, иначе все остальные преимущества собственного решения сведутся на нет, если код будет написан начинаящим.

Резюмируем плюсы и минусы каждого подхода.

*Плюсы использования готовых кодов:*

- быстрая разработка проекта;
- код оттестирован многими пользователями;
- новичкам можно объяснить, как работает программа;
- решение проблем, которые трудно предвидеть, реализуя функцию библиотеки самостоятельно;
- программист концентрируется на основной задаче, а не на вспомогательных;

- разработка собственной библиотеки стоит большего количества человеко-часов, чем готовая.

*Минусы:*

- обычно программные компоненты продаются с лицензиями на каждый компьютер, и стоимость всех лицензий может превысить стоимость собственной разработки;
- в библиотеке очень много классов, сложная общая архитектура, тяжело разбираться в возникающих ошибках;
- одна библиотека может содержать не все специфические функции, которые нужны для проекта;
- задача не решается оптимальным способом (скорость/память), требуется загрузка нескольких библиотек с множеством ненужных для проекта функций;
- могут возникнуть проблемы с обновлением библиотек в будущем;
- к чужой библиотеке тоже надо писать код и разбираться в документации.

В использовании собственной разработки тоже есть свои *плюсы*:

- члены команды знают, как работают программы;
- проще искать причины ошибок, так как написанный код минимален;
- максимально производительное решение;
- возможность развития библиотеки в нужном вам направлении.

*И минусы:*

- требуется высококвалифицированный разработчик для написания хорошей библиотеки;
- длительная разработка и тестирование;
- содержит достаточное количество ошибок, особенно на стадии внедрения;
- новички не знают, как работает программа.

В каких случаях не стоит писать программы самим:

- вы не обладаете необходимой квалификацией (как программист или конкретно в этой области) для качественной реализации библиотеки;
- речь идет об одной из стандартных функций, которые используются в библиотеках как базовые;
- существующая библиотека написана достаточно просто и ясно;
- нет уверенности, что ваше решение будет проще или быстрее работать.

Если задача, решаемая библиотекой действительно сложна и избыточна, то собственная реализация будет лучше и проще, на ее написание и отладку уйдет меньше времени и сил, чем на обнаружение и исправление ошибок.

ние ошибок в чужой библиотеке. Кроме того, нередко удается эту работу переложить на автора библиотеки.

Можно сделать следующий вывод: если только писать код, много сил и времени может уйти на поиск решения задач, которые уже давно решены, причем оптимальным образом.

Создавать свою библиотеку необходимо, если нужную программу быстрее сделать, чем найти готовую, и последующее использование несовместимо с существующими библиотеками.

Каждый из подходов имеет свои факторы и риски. При использовании чужих программ ключевым фактором является минимизация времени и стоимости начальной разработки. Основной риск при первом подходе – проблема внедрения и сопровождения.

Ключевой фактор при самостоятельном программировании – длительная жизнь проекта. С запуском проекта разработка не заканчивается, а только начинается. Другими словами, после начальной разработки и тестирования имеется конкурентоспособный продукт, ориентированный на долгосрочную перспективу. Любое удачное нестандартное решение – это изюминка проекта, его конкурентное преимущество. Проект уже на этапе разработки проектируется готовым к будущему расширению и тиражированию. Производится тщательное тестирование. Разработчики должны быть опытными, а это дорого. Риск второго подхода один – это возможная высокая стоимость проекта. Этот риск компенсируется тем, что такой проект, будучи успешным, способен «кормить» предприятие многие годы.

Для начинающих использование и анализ готовых решений позволяют быстрее повышать квалификацию и создавать несложные проекты. Рассмотрим два варианта библиотек, реализующих многие функции обработки изображений и имеющих открытые коды: Матлаб и OpenCV.

## 12.2. Система Матлаб

Matlab (от англ. MatrixLaboratory – матричная лаборатория) – это пакет прикладных программ для решения задач технических вычислений, множество специализированных библиотек и одноименный язык программирования. Язык Матлаб – интерпретатор, он реализован для большинства современных операционных систем.

Использование Матлаб ускоряет исследовательскую работу в несколько раз, позволяет сосредоточиться именно на проверке алгоритма, а не на его реализации, ошибки реализации отслеживаются легче.

В Матлаб можно создавать специальные библиотеки функций (toolbox), расширяющие его функциональность. Например, в базовой версии имеются библиотеки Digital Signal Processing Toolbox, Image Pro-

cessing Toolbox, Wavelet Toolbox, Filter Design Toolbox, которые используются для цифровой обработки сигналов и изображений.

Пакет Image Processing Toolbox освобождает разработчика от поиска решений, кодирования и отладки алгоритмов, позволяя сосредоточить усилия на решении основной практической задачи. Матлаб и пакет Image Processing максимально приспособлены для обучения, развития новых идей и их проверки [3].

Image Processing Toolbox (<http://matlab.exponenta.ru/imageprocess/book5/index.php>) представляет собой библиотеку основных функций обработки изображений. В их число входят:

- чтение и запись различных форматов изображений, видеопоследовательностей изображений;
- средства построения интерактивного графического интерфейса пользователя (GUI Builder) для ускорения разработки GUI-шаблонов, позволяющие настраивать его под задачи пользователя;
- преобразование в различные типы изображений и цветовые модели, изменение палитры изображений;
- геометрические преобразования;
- пространственные преобразования изображений;
- обработка выделенной области изображения;
- морфологические операции;
- скользящая и блочная обработка;
- линейная и нелинейная фильтрация;
- улучшение качества изображений;
- восстановление изображений;
- анализ изображений.

Большинство функций библиотеки представлены в системе в виде *m*-файлов. В них оптимально реализованы наиболее часто используемые алгоритмы обработки изображений. Коды этих функций открыты.

Отличительной чертой Матлаб является ориентация на матричные операции. Даже константа трактуется как матрица размером 1x1. Такая ориентация очень удобна для обработки изображений.

Преимущества системы Матлаб:

- большое количество встроенных операций по работе с матрицами. Это упрощает как программирование различных задач, связанных с решением систем линейных алгебраических уравнений, так и работу с изображениями;
- богатый набор постоянно развивающихся библиотек различных математических методов (численные методы, дифференциальные уравнения, обработка сигналов, вычислительная геометрия и т. д.);

- в Интернете доступно множество дополнительных библиотек и функций для Матлаб;
- многие функции описаны в книгах;
- большие возможности по быстрой и простой (для разработчика) визуализации различных данных (двухмерных и трехмерных графиков функций, множеств и т. д.);
- имеется интерпретатор с постоянно доступной командной строкой (в нужный момент программу можно приостановить и просмотреть данные, полученные на определенном шаге, отобразить их графически, модифицировать, исправить параметры вызванной функции и т. д.), что облегчает отладку и понимание ошибок в программе;
- существует возможность использовать Матлаб в программах, написанных на C++, и наоборот (в Матлаб можно использовать библиотеки, написанные на C++). Таким образом удается тестировать алгоритмы, написанные на C++, с помощью среды Матлаб и использовать в своих программах средства Матлаб (визуализацию данных и другие рутинные операции). Например, добавлять в свою программу вывод 3D-графиком (для отладки) весьма трудоемко, а подключившись к Матлаб, есть возможность передать туда данные и построить необходимые графики.

Приведем пример кода простейшей программы обработки изображения на языке Матлаб (результат обработки представлен на рис. 12.1):

```
A=imread ('cameraman.tif'); % прочитать изображение в A
B=A/2; % уменьшить яркость всех пикселов в два раза
imshow(B); % отобразить результат обработки.
```

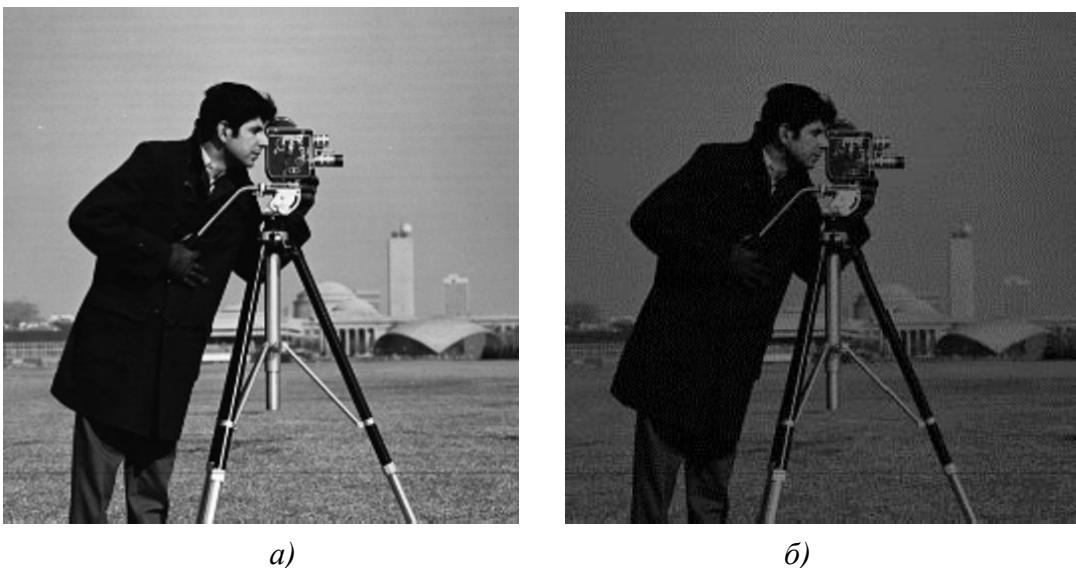


Рис. 12.1. Исходное изображение (*a*) и результат обработки системой Матлаб (*б*)

### **12.3. Открытая библиотека OpenCv**

OpenCV (от англ. OpenSourceComputerVisionLibrary – библиотека компьютерного зрения с открытым исходным кодом) – это библиотека функций, реализующих алгоритмы компьютерного зрения, обработки изображений и численных алгоритмов общего назначения с открытым кодом. Библиотека написана на С и С++, работает под операционными системами Linux, Windows и Mac OS, может работать с многоядерными процессорами. Активно разрабатываются интерфейсы для Python, Ruby, Matlab, Java, Lua и других языков [10].

Библиотека OpenCV содержит более 500 функций, которые используются в различных областях. Например, в таких приложениях, как контроль качества на производстве, системы безопасности, обработка изображений, пользовательские интерфейсы, стереоскопическое зрение и роботостроение, получение видео с камер, распознание образов (включая обнаружение лиц, отслеживание движений в 2D- и 3D-измерениях), 3D-реконструкция стереоскопического зрения. OpenCV также включает библиотеку обучения машин, ориентированную на задачи статистического распознавания образов и кластеризации.

OpenCV используется во многих приложениях и научно-исследовательских работах. Основные компоненты OpenCV:

opencv\_core – библиотека основных универсальных функций, включает базовые структуры, вычисления математических функций, генераторы случайных чисел, операции линейной алгебры, дискретное преобразование Фурье, дискретное косинусное преобразование, функции ввода/вывода для XML и YAWL и т. д.;

opencv\_imgproc – библиотека функций для обработки изображений (фильтрация, геометрические преобразования, преобразование цветовых пространств и т. д.);

opencv\_highgui – простой пользовательский интерфейс, ввод/вывод изображений и видео;

opencv\_ml – библиотека статистических методов машинного обучения (машины опорных векторов, деревья решений, обучение с учителем и т. д.);

opencv\_features2d – библиотека функций для распознавания и описания плоских примитивов (SURF, FAST и др.);

opencv\_video – библиотека функций для анализа движения и отслеживания объектов (оптический поток, шаблоны движения, устранение фона);

`opencv_objdetect` – библиотека функций для обнаружения объектов на изображении (поиск лиц алгоритмом Виолы – Джонса, распознавание людей HOG и т. д.);

`opencv_calib3d` – библиотека функций для калибровки камеры, поиска стереосоответствия и обработки трехмерных данных;

`opencv_flann` – библиотека быстрого поиска ближайших соседей (FLANN 1.5) и обертки OpenCV;

`opencv_gpu` – библиотека для ускорения некоторых функций OpenCV за счет архитектуры параллельных вычислений CUDA, создана при поддержке NVidia.

Документацию по библиотеке можно посмотреть на сайте <http://docs.opencv.org/index.html>.

Недавно появилась возможность совместить функции, реализованные в системе Матлаб и библиотеке OpenCV [24]. К. Ямагучи разработал библиотеку `texhopencv`, которая позволяет вызвать любую функцию из библиотеки OpenCV в программе, написанной на языке Матлаб. Например, загрузить изображение, используя Матлаб и отобразить его на экране с помощью OpenCV, можно следующими командами:

```
I = imread('cameraman.tif');
cv.imshow(I).
```

### Выводы по части III

Решение конкретной прикладной задачи по обработке изображений зависит от опыта разработчика. К сожалению, при небольшом изменении условий получения снимков одного и того же объекта могут существенно измениться их цифровые версии, что может потребовать изменить методы обработки изображений этих объектов. Например, на рис. 12.2 показаны два снимка одного участка местности, полученные в разных диапазонах электромагнитного спектра.



a)



б)

Рис. 12.2. Космические снимки одной местности в разных диапазонах: *а*) ближнем инфракрасном спектре; *б*) красном спектре

Часто возникает следующая ситуация: имеется одно или несколько изображений (очень часто с разными недостатками); требуется, используя их, решить некоторую задачу. Например, распознать представленный на них объект – номер автомобиля.

Обработка существенно меняющихся изображений разных типов – это в большей степени искусство, чем ремесло. Но искусство, опирающееся на разнообразные знания. Знание методов и опыт в поиске нужной последовательности их применения облегчат решение поставленной задачи. Даже постоянно работая с узким классом изображений (например, ретушируя фотопортреты), полезно знать основы применяемых методов.

## **ЗАКЛЮЧЕНИЕ**

В книге последовательно изложены вопросы получения и описания изображений, их представление в цифровом виде, даны основные методы их обработки, приведен ряд примеров решения прикладных задач посредством обработки изображений разных объектов, а также представлены популярные программные комплексы с реализованными базовыми функциями обработки.

Изображение – это двухмерный сигнал, отраженный в определенном участке электромагнитного спектра и зарегистрированный зрительной системой человека или прибором. Исключением является рентгеновское излучение, которое регистрируется как интегральный сигнал, прошедший через некоторое тело.

Несмотря на разнообразие способов формирования изображений, цифровое изображение представлено матрицей абстрактных целых чисел, определяющих яркость или цвет в условной безразмерной шкале. Эти матрицы могут быть описаны в разных форматах и представлены в виде файлов данных.

К матрицам, представляющим цифровые изображения, последовательно применяются базовые методы обработки, в результате изображения преобразуются в соответствии с поставленной задачей: изменить визуальное представление изображения либо подготовить его для автоматического (компьютерного) анализа.

Получение новых типов изображений требует новых методов их обработки. Реализация новых теорий позволяет создавать новые устройства регистрации изображений. По ходу развития техники совершенствуются известные способы получения изображений и развиваются новые, например голограммические (в данной книге не рассматриваются). Процесс получения новых изображений с улучшенными характеристиками и извлечения из них новой информации постоянно совершенствуется.

## СПИСОК ЛИТЕРАТУРЫ

1. Ватолин, Д.С. Алгоритмы сжатия изображений : метод. пособие // Д.С. Ватолин. – М. : Изд. отд. фак. ВМиК МГУ им. М.В. Ломоносова, 1999. – 76 с.
2. Гонсалес, Р. Цифровая обработка изображений : пер. с англ. // Р. Гонсалес, Р. Вудс. – М. : Техносфера, 2005. – 1072 с.
3. Гонсалес, Р. Цифровая обработка изображений в среде MATLAB / Р. Гонсалес, Р. Вудс, С.М. Эддинс. – М. : Техносфера, 2006. – 616 с.
4. Дзюба, В.Г. Алгоритм распознавания предупреждающих дорожных знаков / В.Г. Дзюба, А.В. Дзюба // Электроника и связь. – 2009. – № 6. – С. 13–17.
5. Журавель, И.М. Краткий курс теории обработки изображений / И.М. Журавель [Электронный ресурс]. – М., 1999. – Режим доступа : <http://matlab.exponenta.ru/imageprocess/book2/> – Дата доступа : 25.03.2014.
6. Методы компьютерной обработки изображений / под ред. В.А. Сойфера. – М. : Физматлит, 2003. – 784 с.
7. Миано, Дж. Форматы и алгоритмы сжатия изображений в действии / Дж. Миано. – М. : Триумф, 2003. – 336 с.
8. Милчев, М.Н. Самое главное о ... Выбор цифрового фотоаппарата / М.Н. Милчев. – Спб. : Питер, 2006.
9. Мюррей, Д.Д. Энциклопедия форматов графических файлов : пер. с англ. / Д.Д. Мюррей, У. Ван. Райнер. – Киев : BHV, 1997. – 672 с.
10. OpenCV – Библиотека компьютерного зрения с открытым исходным кодом [Электронный ресурс]. – Режим доступа : <http://locv.ru>. – Дата доступа : 25.03.2014.
11. Приоров, А.Л. Цифровая обработка изображений : учеб. пособие / А.Л. Приоров, И.В. Апальков, В.В. Хрящев. – Ярославль : ЯрГУ, 2007. – 235 с.
12. Прэтт, У. Цифровая обработка изображений : пер. с англ. / У. Прэтт. – М. : Мир, 1982. – Кн. 2. – 480 с.
13. Самарин, Ю. Солнцем цифры не испортишь или О цифровых источниках получения изображений / Ю. Самарин // КомпьютерАрт. – 2012. – № 1. – Режим доступа : <http://www.compuart.ru/article.aspx?id=22776&id=1046>. – Дата доступа : 25. 03.2014.
14. Стариченко, Б.Е. Теоретические основы информатики : учеб. пособие для вузов. – 2-е изд. – М. : Горячая линия – Телеком, 2003.
15. Старовойтов, В.В. Локальные геометрические методы цифровой обработки и анализа изображений / В.В. Старовойтов. – Минск : Ин-т техн. кибернетики НАН Беларуси, 1997. – 282 с.

16. Сэлмон, Д. Сжатие данных, изображений и звука / Д. Сэлмон. – М. : Техносфера, 2004.
17. Тонквист, Г. Аспекты цвета. Что они значат и как могут быть использованы / Г. Тонквист // Проблема цвета в психологии; отв. ред. А.А. Митькин, Н.Н. Корж. – М. : Наука, 1993. – С. 5–53.
18. Тропченко, А.Ю. Методы сжатия изображений, аудиосигналов и видео : учеб. пособие / А.Ю. Тропченко, А.А. Тропченко. – СПб. : СПбГУ ИТМО, 2009.
19. Фисенко, В.Т. Компьютерная обработка и распознавание изображений : учеб. пособие / В.Т. Фисенко, Т.Ю. Фисенко. – СПб. : СПбГУ ИТМО, 2008. – 192 с.
20. Цифровая обработка изображений в информационных системах : учеб. пособие // И.С. Грузман [и др.]. – Новосибирск : Изд-во НГТУ, 2002. – 352 с.
21. Шапиро, Л. Компьютерное зрение : пер. с англ. / Л. Шапиро, Дж. Стокман. – М. : БИНОМ. Лаборатория знаний, 2006. – 752 с.
22. Яне, Б. Цифровая обработка изображений : пер. с англ. / Б. Яне. – М. : Техносфера, 2007. – 584 с.
23. A survey of classical methods and new trends in pansharpening of multispectral images / I. Amro [et al.] // EURASIP Journal on Advances in Signal Processing. – 2011. – Т. 2011, № 1. – Р. 1–22.
24. Briston, H. MATLAB Bindings for OpenCV / H. Briston [Electronic resource]. – Mode of access : <http://imageprocessingblog.com/matlab-bindings-for-opencv>. – Date of access : 25.03.2014.
25. Detection and mapping of the November 2002 PRESTIGE tanker oil spill in Galicia, Spain, with the airborne multispectral CASI sensor / M. Lennon [et al.] // 3rd EARSEL Workshop on Imaging Spectroscopy, Germany. – 2003. – Р. 245–251.
26. Cuturicu, C. CRYX's note about the JPEG decoding algorithm / C. Cuturicu, O. Fromme [Electronic resource]. – 1999. – Mode of access : [http://www.coco3.com/text/doc\\_JPEG.txt](http://www.coco3.com/text/doc_JPEG.txt). – Перевод на рус. яз. : <http://www.matrix-wb.h1.ru/article/formats/jpeg-rus.html>. – Date of access : 25.03.2014.
27. Zhang, Y.Y. A modified parallel thinning algorithm / Y.Y. Zhang, P.S.P. Wang // Intern. Conf. on Pattern Recognition, Rome, Italy, 14–17 nov. 1988. – Rome, 1988. – Р. 1023–1025.

# ГЛОССАРИЙ

## **Алгоритм**

Набор точных инструкций, описывающих порядок действий исполнителя, для решения задачи за конечное время. Алгоритм может базироваться на некотором методе, но он не объясняет, почему надо делать так, а не иначе, он носит чисто прикладной характер. Алгоритм, как правило, не ориентирован на язык программирования и операционную систему, т. е. это набор абстрагированных инструкций универсального назначения. В отличие от метода и технологии носит более частный характер.

## **Альфа-канал**

Четвертый слой RGB-изображения, определяющий «прозрачность» цветов пикселов исходного изображения. Создается искусственно и служит для формирования эффекта частичной прозрачности.

## **Анимация**

Технология, имитирующая движение объектов в фильме, компьютерной игре путем отображения последовательности искусственно созданных изображений с частотой, обеспечивающей зрительное восприятие плавно перемещающихся объектов.

## **Аппроксимация**

Построение непрерывной функции заданного типа, описывающей заданное дискретное множество значений с минимальной ошибкой. В отличие от интерполяции все или часть заданных значений приближенно описываются новой функцией.

## **Блок-схема**

Способ графического описания алгоритма путем записи отдельных инструкций внутри геометрических фигур определенной формы, соединенных переходами. Правила записи блок-схем определены в ГОСТ 19.701–90 «Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения».

## **Видео**

Технология записи аудио- и видеоданных путем фиксации реальной динамически изменяющейся сцены в виде множества фотографий (не менее 18 в секунду).

## **Вокセル**

Аналогичен пикселу, используется для описания наименьшего элемента трехмерного цифрового изображения, определяется тремя пространственными координатами и значением яркости (цвета).

## **Гамма-коррекция**

Нелинейная коррекция яркости изображения для его единообразного отображения на мониторах разных производителей.

### **Гиперспектральный снимок**

Трехмерное изображение, регистрируемое посредством разложения отраженного света с помощью призмы и проецирования на фотоприемную матрицу.

### **Глубина цвета**

Количество бит (bpp), описывающих значение яркости (цвета) одного пикселя, единица измерения.

### **Градации серого**

Целые числа от 0 до  $2^N - 1$ , описывающие значения оттенков серого цвета в полутонахом представлении изображения.

### **Градиент**

Вектор максимального перепада яркости между соседними пикселями. Описывается направлением и значением модуля.

### **Графический интерфейс**

Графический способ организации взаимодействия пользователя с вычислительной системой. Основные элементы графического интерфейса: окна, меню, линейки инструментов, планки инструментов (toolbar) в виде наборов пиктограмм, означающих некое действие, линейки прокрутки (scrollbar). Элементы управления: кнопки командные (commandbuttons), кнопки настройки (optionsbuttons), переключатели (radiobuttons), наборы значений (valuesets), выключатели (checkbox), списки (listbox), текстовые зоны (textbox) и др.

### **Данные**

Сведения, представленные в формализованном виде, пригодном для понимания и обработки. Основными типами данных являются графические (видео, фото, чертеж) и символьные (числа, буквы). Формы представления данных: аналоговая (надпись на бумаге), дискретная (фильм на кинопленке имеет дискретность по временной шкале) и цифровая (песня в формате MP3). Одни и те же цифровые данные можно записать в разных форматах, например изображение в формате JPEG, числа в формате с плавающей запятой, бинарные данные в виде нулей и единиц.

### **Декодер**

Программа или алгоритм декодирования (восстановления) сжатых данных.

**Дешифрование** (от франц. déchiffrer – разбирать, разгадывать). *Дешифрование (тематическое) снимка* – это интерактивное или ручное опознание определенных объектов и явлений (их классификация) либо всего содержимого снимка. *Интерпретация снимка* – это определение типов объектов, их свойств и параметров. Дешифровщик должен быть

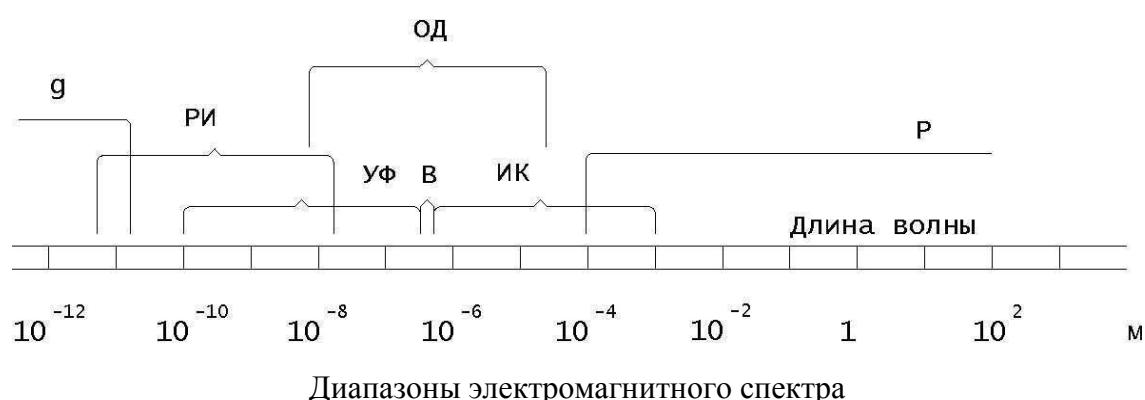
экспертом в определенной тематической области, знать характеристики и свойства изображаемых объектов, их прямые и косвенные признаки.

### Диапазоны электромагнитного спектра:

g (гамма)-излучение, РИ – рентгеновское излучение, ОД – оптический диапазон, УФ – ультрафиолетовое излучение, В – видимое излучение, ИК – инфракрасное излучение, Р – радиоволны (рисунок).

*Видимый* – диапазон волн с длинами от 380 до 780 нм от фиолетового до красного, отраженные волны этого диапазона воспринимает глаз человека.

*Оптический* – участок спектра, отраженные волны которого можно зарегистрировать оптическими системами, включает ультрафиолетовый (180–380 нм), видимый (380–780 нм) и частично инфракрасный (780–2000 нм) диапазоны спектра.



### Дискретизация

Преобразование непрерывной функции в дискретную. Результатом является дискретное множество аргументов с непрерывной областью значений.

### Знания

Результат абстрагирования данных одного или нескольких классов.

### Зрение компьютерное

Область создания искусственных систем, которые получают и обрабатывают информацию из изображений или видеопоследовательностей. Обработка, как правило, выполняется в режиме реального времени.

### Изображения

*Анализ* – вычисление признаков объектов определенных классов по их изображению.

*Высококонтрастное* – изображение с резкими перепадами яркости.

*Обработка* – преобразование, для которого исходные данные представляют собой одно или несколько (цифровых) изображений, результатом также является изображение.

*Разрешение* – характеристика цифрового изображения, определяется устройством регистрации (например, сканером) или графическим редактором при синтезе, измеряется в dpi (количество точек на дюйм).

*Распознавание* – распознавание объектов определенных классов, представленных на изображении.

*Растровое* – цифровое изображение, представленное в виде матрицы пикселов с целочисленными значениями яркости (цвета).

*Сжатие* – преобразование изображений с целью уменьшения занимаемого ими объема при хранении. Сжатие выполняется без потерь, если исходное изображение восстанавливается с точностью до бита; в противном случае – сжатие с потерями.

### **Интерактивная обработка**

Способ обработки данных, результат работы которого зависит от выбора пользователем определенных параметров или ветви алгоритма вычислений в процессе выполнения. Могут изменяться как результат, так и его представление. Реализуется посредством применения графического интерфейса пользователя.

### **Интерполяция**

Построение непрерывной функции, принимающей все заданное дискретное множество значений и вычисляющей промежуточные значения. В отличие от аппроксимации строится функция, принимающая все заданные значения.

### **Информатика**

Область человеческой деятельности, связанная с преобразованием информации средствами вычислительной техники и взаимодействия этих средств со средой применения.

### **Информация**

Данные об объектах и явлениях, их параметрах, свойствах и состояниях, в которых минимизирована избыточность их описания. Информация возникает только в результате обработки данных. Описание одного и того же факта может быть представлено данными разного типа (текстом, звуком, изображением). Основная форма представления информации – цифровая. В отличие от данных ее можно измерить.

*Теория* – математическая теория, включает в себя описание и оценки методов извлечения, передачи, хранения и классификации информации.

### **Кадр**

Одиночное изображение из видеопоследовательности.

### **Квантование яркости изображения**

Преобразование непрерывной функции яркости в конечное множество значений.

## **Кибернетика**

Наука об управлении в различных системах: технических, биологических, социальных и др. Шире, чем информатика.

## **Кодек**

Термин для совместного обозначения кодера и декодера.

## **Кодер**

Программа или алгоритм кодирования (сжатия) данных.

## **Компьютерная программа**

Последовательность инструкций, закодированных на определенном языке программирования. Предназначена для исполнения с помощью вычислительной машины, т. е. это реализация алгоритма. Программа – один из компонентов программного обеспечения.

## **Космическая съемка Земли**

Процесс получения информации о поверхности Земли и расположенных на ней объектах из космоса путем регистрации собственного и отраженного излучения элементов суши, водной поверхности и атмосферы в различных диапазонах электромагнитного спектра.

## **Метод**

Объясняет, почему и как из набора данных можно получить другие данные или решить определенную задачу.

*В информатике* – обобщенный способ решения задач определенного класса (обосновывает, почему и как в пределах этого класса можно применять данный метод с гарантированным получением результата).

*Научный* – совокупность способов получения новых знаний и методов решения задач.

## **Методика**

Перечень действий (не полностью формализуемых), выполняющихся для получения желаемого результата. Например, методика обучения программированию на языке С. Достижение результата (в отличие от метода и технологии) методика не гарантирует. Для достижения одного результата могут применяться разные методики.

## **Монитор жидкокристаллический**

Монитор на базе жидких кристаллов. Различают активно- и пассивно-матричные мониторы.

## **Морфинг**

Преобразование, при котором одно изображение плавно преобразуется в другое путем построения серии промежуточных изображений.

## **Муар**

Вид искажений в виде колец или полос на экране монитора, вызванных в основном биением периодических структур раstra оптическо-

го изображения. Визуально эти искажения воспринимаются как «волокнистость» и волнообразные разводы изображения.

### **Мультимедиа**

Совокупность компьютерных технологий, одновременно использующих несколько информационных сред: изображения, текст, видео, анимацию, звук.

### **Мультиспектральный снимок**

Набор моноспектральных (полутоновых) изображений одной и той же сцены, полученных одновременно в разных спектральных каналах с помощью светофильтров.

### **Науки прикладные**

Целью прикладных наук является применение фундаментальных наук для решения практических проблем.

### **Науки технические**

Комплекс прикладных наук, исследующих особенности развития техники, ее саму и ее приложения для практических нужд. Занимают промежуточное положение между естественными и гуманитарными науками, так как техника является продуктом человека и не является продуктом природы.

### **Науки фундаментальные**

Цель фундаментальных наук – это разработка теорий и познание законов, управляющих взаимодействием базисных структур природы, общества и мышления.

### **Образ**

Унифицированное представление некоторого класса объектов.

### **Освещенность**

Величина светового потока, приходящегося на единицу поверхности. Освещенность обратно пропорциональна квадрату расстояния от источника до освещаемой поверхности и зависит от угла, составляемого направлением светового потока с нормалью к освещаемой поверхности. Единица измерения освещенности – люкс.

### **Отражающая способность**

Характеристика, определяющая способность реальных объектов отражать световое излучение. Характеризуется коэффициентом отражения: снег (90 %), белая краска (75–90 %), бетон (25–30 %), трава (20 %).

### **Оттенок**

Градация тона, оттенок серого (полутон) в полутоновых изображениях.

### **Панхроматический снимок**

Изображение, полученное путем регистрации отраженных волн видимого и ближнего инфракрасного диапазонов спектра.

## **ПЗС**

Прибор с зарядовой связью. ПЗС-матрица – полупроводниковая пластина, содержащая множество маленьких фотоприемников, преобразующих падающий свет в электрический заряд. Светочувствительные элементы ПЗС называют пикселями.

## **Пикセル**

- 1) – это наименьший логический элемент двухмерного цифрового изображения;
- 2) элемент светочувствительной матрицы устройства регистрации изображений (сканера, фотокамеры и т. п.);
- 3) наименьший элемент поверхности визуализации в графических системах вывода (монитора, принтера и т. д.), которому может быть задан цвет.

## **Программа компьютерная**

Последовательность инструкций, закодированных на определенном языке программирования, предназначенная для исполнения с помощью ЭВМ в определенной операционной системе, *реализация* алгоритма.

## **Растр**

- 1) решётка для перевода полутонаовых и цветных изображений в штриховые, для печати (в полиграфии);
- 2) способ представления цифровых изображений в виде матрицы пикселов с целочисленными значениями яркости (цвета) (в информатике).

## **Сжатия**

*Коэффициент* – величина, используемая для определения эффективности метода сжатия; равна отношению размера сжатого файла к размеру исходного.

*Видео*. Основано на двух принципах: 1) использование пространственной избыточности, присутствующей в отдельном кадре; 2) текущий кадр часто похож на соседние кадры (временная избыточность). Алгоритм сжимает первый кадр как статическое изображение, затем каждый следующий кадр описывается как разность от предшествующего и кодируется только эта разность.

## **Светлота**

Субъективная яркость участка изображения, отнесенная к субъективной яркости поверхности, воспринимаемой человеком как белая.

## **Технология**

Совокупность методов, процессов, используемых в определенной отрасли деятельности, а также описание способов технического производства; в узком смысле – комплекс организационных мер и операций, направленных на изготовление, обслуживание, ремонт изделия с опре-

деленным качеством и затратами. В отличие от методики технологии гарантирует получение запланированного результата при выполнении соответствующих условий.

**Информационная** – описание четко упорядоченной последовательности применения средств получения, хранения, обработки и передачи цифровых данных для создания информационных продуктов.

### **Технологический процесс**

Упорядоченная последовательность действий, выполняющихся для получения требуемого результата или продукта. Включает множество алгоритмов.

### **Цвет**

Качественная субъективная характеристика электромагнитного излучения оптического диапазона. Восприятие цвета определяется индивидуальными физиологическими возможностями человека, а также физическими характеристиками внешней среды: спектральным составом, цветовым и яркостным контрастом источников освещения и отражающих объектов.

**Дополнительные** – пары цветов, создающие в смеси белый цвет. Например, голубой-красный, желтый-синий, пурпурный-зеленый и др. Первые цвета указанных пар используются в субтрактивной цветовой модели как основные. При смешении дают черный цвет.

**Индексированные** – табличный набор оттенков цвета (обычно 256), сформирован из исходного изображения либо некоторой цветовой палитры. Используются для описания цветных изображений с целью уменьшения размера файла.

**Насыщенность (Saturation)** – характеристика чистоты цвета, степень визуального отличия серого цвета. Чем больше серого цвета, тем ниже насыщенность.

**Основные** – три цвета, при смешении которых в разных пропорциях можно получить все оттенки цвета. Основной цвет не может быть получен смешением двух других основных цветов. Число вариантов основных цветов бесконечно.

**Параметры** – три количественные характеристики, однозначно определяющие оттенок цвета в определенной цветовой модели.

### **Цветовая модель**

Математическая модель описания оттенков цвета в виде набора из трех параметров, называемых цветовыми координатами в некотором цветовом пространстве.

**Аддитивная** – модель, в которой воспроизведение цвета получают путем сложения основных цветов. Равное смешение трех основных аддитивных цветов (красного, зеленого и синего) дает белый цвет. При по-

парном смешении основных аддитивных цветов получают основные субтрактивные цвета.

### **Цветовая палитра (палитра цвета)**

Фиксированный набор цветов и оттенков, имеющий цифровое описание, обычно 256 или менее.

### **Цветовая таблица**

Матрица цветовых параметров, используемая для вывода цвета на экран и конвертирования цвета из одной модели в другую.

### **Цветовая температура**

Характеристика интенсивности излучения источника света в виде функции от длины волны. Определяется как температура абсолютно черного тела, при которой оно испускает излучение. Измеряется в кельвинах.

### **Цветовой тон**

Характеристика цвета, определяемая положением пика излучения в видимом диапазоне спектра, связанном с названием цвета (красный, синий, зеленый и т. п.), измеряется в градусах от 0 до 360.

### **Цифрование**

Преобразование аналоговых данных в цифровое представление.

### **Формат**

Спецификация структуры данных для их записи в компьютерном файле.

### **Функции задания**

Способы: аналитический, графический, табличный (Look-UpTable).

### **Яркость**

Световая характеристика тел. Отношение силы света, излучаемого поверхностью, к площади ее проекции на плоскости, перпендикулярной оси наблюдения. В цифровых изображениях значение яркости пикселя задается безразмерным целым числом.

*Цвета* – это коэффициент отражения, равный отношению отраженного светового потока к падающему. Синоним – светлота.

### **JPEG**

1) алгоритм, разработанный объединенной группой экспертов по фотографии (Joint Photographic Experts Group), для сжатия полноцветных изображений и монохромных цифровых фотографий реальных сцен. В основе алгоритма лежит разбиение изображения на блоки 8x8 пикселов и использование дискретного косинусного преобразования;

2) стандарт сжатия изображений на базе алгоритма JPEG, принятый Международной организацией стандартизации ISO;

# ПРИЛОЖЕНИЕ

## Преобразование цветовых моделей друг в друга

Таблица 1

Конвертация основных цветовых моделей друг в друга

	XYZ	RGB	L*ab	L*uv	HSL	HSV	CMY	CMYK
XYZ	-	+						
RGB	+	-	XYZ	XYZ	+	+	+	+
L*ab		XYZ	-	XYZ				
L*uv		XYZ	XYZ	-				
HSL		+			-			
HSV		+				-		
CMY		+					-	
CMYK		+						-

*Примечание:* знак + означает прямой способ конвертации;  
XYZ – конвертация через модель XYZ;  
пустая клетка – вариант практически не используется, но можно выполнить конвертацией через RGB.

### 1. RGB —> XYZ

```
var_R = (R / 255)                                // R от 0 до 255
var_G = (G / 255)                                // G от 0 до 255
var_B = (B / 255)                                // B от 0 до 255
if (var_R > 0.04045) var_R = ((var_R + 0.055) / 1.055) ^ 2.4
else var_R = var_R / 12.92
if (var_G > 0.04045) var_G = ((var_G + 0.055) / 1.055) ^ 2.4
else var_G = var_G / 12.92
if (var_B > 0.04045) var_B = ((var_B + 0.055) / 1.055) ^ 2.4
else var_B = var_B / 12.92
var_R = var_R * 100
var_G = var_G * 100
var_B = var_B * 100
X = var_R * 0.4124 + var_G * 0.3576 + var_B * 0.1805
Y = var_R * 0.2126 + var_G * 0.7152 + var_B * 0.0722
Z = var_R * 0.0193 + var_G * 0.1192 + var_B * 0.9505
```

### 2. XYZ —> RGB

```
var_X = X / 100                                  // X от 0 до 95.047
var_Y = Y / 100                                  // Y от 0 до 100.000
var_Z = Z / 100                                  // Z от 0 до 108.883
var_R = var_X * 3.2406 + var_Y * -1.5372 + var_Z * -0.4986
var_G = var_X * -0.9689 + var_Y * 1.8758 + var_Z * 0.0415
var_B = var_X * 0.0557 + var_Y * -0.2040 + var_Z * 1.0570
```

```

if (var_R > 0.0031308) var_R = 1.055 * (var_R ^ (1 / 2.4)) -0.055
else var_R = 12.92 * var_R
if (var_G > 0.0031308) var_G = 1.055 * (var_G ^ (1 / 2.4)) -0.055
else var_G = 12.92 * var_G
if (var_B > 0.0031308) var_B = 1.055 * (var_B ^ (1 / 2.4)) -0.055
else var_B = 12.92 * var_B
R = var_R * 255
G = var_G * 255
B = var_B * 255

```

### 3. XYZ —> CIE-L\*ab

var_X = X / ref_X	// ref_X= 95.047
var_Y = Y / ref_Y	// ref_Y = 100.000
var_Z = Z / ref_Z	// ref_Z = 108.883

```

if (var_X > 0.008856) var_X = var_X ^ (1/3)
else var_X = (7.787 * var_X) + (16 / 116)
if (var_Y > 0.008856) var_Y = var_Y ^ (1/3)
else var_Y = (7.787 * var_Y) + (16 / 116)
if (var_Z > 0.008856) var_Z = var_Z ^ (1/3)
else var_Z = (7.787 * var_Z) + (16 / 116)

```

**CIE-L\*** = (116 \* var\_Y) -16

**CIE-a\*** = 500 \* (var\_X - var\_Y)

**CIE-b\*** = 200 \* (var\_Y - var\_Z)

### 4. CIE-L\*ab —> XYZ

var\_Y = (**CIE-L\*** + 16) / 116

var\_X = **CIE-a\*** / 500 + var\_Y

var\_Z = var\_Y - **CIE-b\*** / 200

```

if (var_Y^3 > 0.008856) var_Y = var_Y^3
else var_Y = (var_Y - 16 / 116) / 7.787
if (var_X^3 > 0.008856) var_X = var_X^3
else var_X = (var_X - 16 / 116) / 7.787
if (var_Z^3 > 0.008856) var_Z = var_Z^3
else var_Z = (var_Z - 16 / 116) / 7.787

```

X = ref_X * var_X	// ref_X= 95.047
Y = ref_Y * var_Y	// ref_Y = 100.000
Z = ref_Z * var_Z	// ref_Z = 108.883

### 5. XYZ —> CIE-L\*uv

var\_U = (4 \* X) / (X + (15 \* Y) + (3 \* Z))

var\_V = (9 \* Y) / (X + (15 \* Y) + (3 \* Z))

var\_Y = Y / 100

```

if (var_Y > 0.008856) var_Y = var_Y ^ (1/3)
else var_Y = (7.787 * var_Y) + (16 / 116)
ref_X = 95.047
ref_Y = 100.000
ref_Z = 108.883
ref_U = (4 * ref_X) / (ref_X + (15 * ref_Y) + (3 * ref_Z))
ref_V = (9 * ref_Y) / (ref_X + (15 * ref_Y) + (3 * ref_Z))

```

**CIE-L\*** = (116 \* var\_Y) - 16  
**CIE-u\*** = 13 \* **CIE-L\*** \* (var\_U - ref\_U)  
**CIE-v\*** = 13 \* **CIE-L\*** \* (var\_V - ref\_V)

#### 6. CIE-L\*uv —> XYZ

var\_Y = (**CIE-L\*** + 16) / 116

```

if (var_Y^3 > 0.008856) var_Y = var_Y^3
else var_Y = (var_Y - 16 / 116) / 7.787
ref_X = 95.047
ref_Y = 100.000
ref_Z = 108.883
ref_U = (4 * ref_X) / (ref_X + (15 * ref_Y) + (3 * ref_Z))
ref_V = (9 * ref_Y) / (ref_X + (15 * ref_Y) + (3 * ref_Z))
var_U = CIE-u* / (13 * CIE-L*) + ref_U
var_V = CIE-v* / (13 * CIE-L*) + ref_V

```

**Y** = var\_Y \* 100  
**X** = -(9 \* Y \* var\_U) / ((var\_U - 4) \* var\_V - var\_U \* var\_V)  
**Z** = (9 \* Y - (15 \* var\_V \* Y) - (var\_V \* X)) / (3 \* var\_V)

#### 7. RGB —> HSL

```

var_R = (R / 255) // RGB от 0 до 255
var_G = (G / 255)
var_B = (B / 255)
var_Min = min(var_R, var_G, var_B)
var_Max = max(var_R, var_G, var_B)
del_Max = var_Max - var_Min
L = (var_Max + var_Min) / 2

if (del_Max == 0)
{
    H = 0 // HSL от 0 до 1
    S = 0
}
else
{

```

```

if (L < 0.5) S = del_Max / (var_Max + var_Min)
else S = del_Max / (2 - var_Max - var_Min)

del_R = (((var_Max - var_R) / 6) + (del_Max / 2)) / del_Max
del_G = (((var_Max - var_G) / 6) + (del_Max / 2)) / del_Max
del_B = (((var_Max - var_B) / 6) + (del_Max / 2)) / del_Max
if (var_R == var_Max) H = del_B - del_G
else if (var_G == var_Max) H = (1 / 3) + del_R - del_B
else if (var_B == var_Max) H = (2 / 3) + del_G - del_R
if (H < 0) H += 1
if (H > 1) H -= 1
}

```

## 8. HSL —> RGB

```

if (S == 0) // HSL от 0 до 1
{
    R = L * 255 // RGB от 0 до 255
    G = L * 255
    B = L * 255
}
else
{
    if (L < 0.5) var_2 = L * (1 + S)
    else var_2 = (L + S) - (S * L)
    var_1 = 2 * L - var_2
    R = 255 * Hue_2_RGB(var_1, var_2, H + (1 / 3))
    G = 255 * Hue_2_RGB(var_1, var_2, H)
    B = 255 * Hue_2_RGB(var_1, var_2, H - (1 / 3))
}

Hue_2_RGB( v1, v2, vH ) // Function Hue_2_RGB
{
    if (vH < 0) vH += 1
    if (vH > 1) vH -= 1
    if ((6 * vH) < 1) return (v1 + (v2 - v1) * 6 * vH)
    if ((2 * vH) < 1) return (v2)
    if ((3 * vH) < 2) return (v1 + (v2 - v1) * ((2 / 3) - vH) * 6)
    return (v1)
}

```

## 9. RGB —> HSV

```

var_R = (R / 255) // RGB от 0 до 255
var_G = (G / 255)
var_B = (B / 255)

```

```

var_Min = min(var_R, var_G, var_B)
var_Max = max(var_R, var_G, var_B)
del_Max = var_Max - var_Min
V = var_Max

if (del_Max == 0)
{
    H = 0                                // HSV от 0 до 1
    S = 0
}
else
{
    S = del_Max / var_Max
    del_R = (((var_Max - var_R) / 6) + (del_Max / 2)) / del_Max
    del_G = (((var_Max - var_G) / 6) + (del_Max / 2)) / del_Max
    del_B = (((var_Max - var_B) / 6) + (del_Max / 2)) / del_Max
    if (var_R == var_Max) H = del_B - del_G
    else if (var_G == var_Max) H = (1 / 3) + del_R - del_B
    else if (var_B == var_Max) H = (2 / 3) + del_G - del_R
    if (H < 0) H += 1
    if (H > 1) H -= 1
}

```

#### 10. HSV → RGB

```

if (S == 0)                                // HSV от 0 до 1
{
    R = V * 255
    G = V * 255
    B = V * 255
}
else
{
    var_h = H * 6
    if (var_h == 6) var_h = 0      // H должно быть < 1
    var_i = int(var_h)           // или ... var_i = floor(var_h)
    var_1 = V * (1 - S)
    var_2 = V * (1 - S * (var_h - var_i))
    var_3 = V * (1 - S * (1 - (var_h - var_i)))
    if (var_i == 0) {var_r = V; var_g = var_3; var_b = var_1}
    else if (var_i == 1) {var_r = var_2; var_g = V; var_b = var_1}
    else if (var_i == 2) {var_r = var_1; var_g = V; var_b = var_3}
    else if (var_i == 3) {var_r = var_1; var_g = var_2; var_b = V}
    else if (var_i == 4) {var_r = var_3; var_g = var_1; var_b = V}
    else {var_r = V; var_g = var_1; var_b = var_2}
    R = var_r * 255
}

```

```

// значения RGB имеют диапазон от 0 до 255
    G = var_g * 255
    B = var_b * 255
}

```

### **11. RGB → CMY**

```

// значения RGB имеют диапазон от 0 до 255
// значения CMY имеют диапазон от 0 до 1
C = 1 - (R / 255)
M = 1 - (G / 255)
Y = 1 - (B / 255)

```

### **12. CMY → RGB**

```

// значения CMY имеют диапазон от 0 до 1
// значения RGB имеют диапазон от 0 до 255
R = (1 - C) * 255
G = (1 - M) * 255
B = (1 - Y) * 255

```

### **13. CMY → CMYK**

```

// все величины моделей CMYK и CMY имеют диапазон значений от 0 до 1
var_K = 1

```

```

if (C < var_K) var_K = C
if (M < var_K) var_K = M
if (Y < var_K) var_K = Y
if (var_K == 1) { //Black
    C = 0
    M = 0
    Y = 0
}
else {
    C = (C - var_K) / (1 - var_K)
    M = (M - var_K) / (1 - var_K)
    Y = (Y - var_K) / (1 - var_K)
}

```

K = var\_K

### **14. CMYK → CMY**

```

// все величины моделей CMYK и CMY имеют диапазон значений от 0 до 1

```

```

C = (C * (1 - K) + K)
M = (M * (1 - K) + K)
Y = (Y * (1 - K) + K)

```

Таблица 2  
Диапазоны значений HSL, HSB, HSV в различных приложениях

Приложение	Модель	Диапазон Н	Диапазон S	Диапазон L/V/B	
Paint Shop Pro	HSL	0–255	0–255	L	0–255
Gimp	HSV	0–360°	0–100	V	0–100
Photoshop	HSB	0–360°	0–100 %	B	0–100 %
Windows	HSL	0–240	0–240	L	0–240
Linux / KDE	HSV	0–360°	0–255	V	0–255
GTK	HSV	0–360°	0–1.0	V	0–1.0
Java (awt.Color)	HSB	0–1.0	0–1.0	B	0–1.0
Apple	HSL	0–360°	0–100 %	L	0–100 %

*Научное издание*  
**Старовойтов Валерий Васильевич**  
**Голуб Юлия Игоревна**

**ЦИФРОВЫЕ ИЗОБРАЖЕНИЯ:  
ОТ ПОЛУЧЕНИЯ ДО ОБРАБОТКИ**

Ответственный за выпуск Рудая Н.А.  
Редактор Мойсейчик С.С.  
Корректор Михайлова А.А.

Подписано в печать 12.05.2014. Формат 60x84 1/16.  
Бумага офсетная. Усл. печ. л. 11,7. Уч.-изд. 14,0. Тираж 100 экз. Заказ 17.

---

Издатель и полиграфическое исполнение:  
Государственное научное учреждение “Объединенный институт проблем  
информатики Национальной академии наук Беларусь”.  
Свидетельство о государственной регистрации издателя, изготовителя,  
распространителя печатных изданий  
№ 1/274 от 04.04.2014.  
Ул. Сурганова, 6, 220012, Минск.