



МИНИСТЕРСТВО ОБРАЗОВАНИЯ
РЕСПУБЛИКИ БЕЛАРУСЬ
Белорусский национальный
технический университет

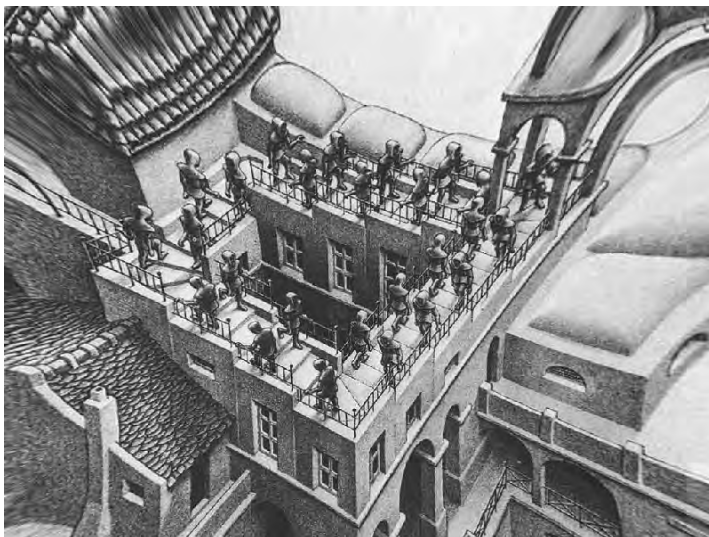
МЕЖДУНАРОДНЫЙ ИНСТИТУТ
ДИСТАНЦИОННОГО ОБРАЗОВАНИЯ

Кафедра «Информационные системы и технологии»

В. В. СТАРОВОЙТОВ
Ю. И. ГОЛУБ

**ПОЛУЧЕНИЕ И ОБРАБОТКА
ИЗОБРАЖЕНИЙ НА ЭВМ**

Учебно-методическое пособие



Минск
БНТУ
2018

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
Белорусский национальный технический университет

МЕЖДУНАРОДНЫЙ ИНСТИТУТ
ДИСТАНЦИОННОГО ОБРАЗОВАНИЯ

Кафедра «Информационные системы и технологии»

В. В. СТАРОВОЙТОВ
Ю. И. ГОЛУБ

ПОЛУЧЕНИЕ И ОБРАБОТКА ИЗОБРАЖЕНИЙ НА ЭВМ

Учебно-методическое пособие

*Рекомендовано учебно-методическим объединением
по образованию в области автоматизации
технологических процессов, производств и управления*

Минск
БНТУ
2018

УДК 004.932(075.8)

ББК 32.97я7

C77

Р е ц е н з е н т ы:

канд. техн. наук, доцент кафедры систем автоматизированного проектирования *И.Л. Ковалева*;
д-р техн. наук, профессор, заведующий кафедрой компьютерных технологий и систем БГУ *А.М. Недзьведь*

Старовойтов, В. В.

C77

Получение и обработка изображений на ЭВМ : учебно-методическое пособие / В.В. Старовойтов, Ю.И. Голуб. – Минск : БНТУ, 2018. – 204 с.

ISBN 978-985-550-770-4.

Цель данной книги – систематизация основных понятий и способов формирования цифровых изображений, а также основ их цифровой обработки. В пособии кратко описываются некоторые понятия, которые пока не вошли ни в один учебник (например, получение гиперспектральных изображений); приведены примеры решения прикладных задач, в основе которых лежит последовательное применение описанных методов.

Книга написана на базе проводимых авторами лекционных и практических занятий по обработке цифровых изображений. Ориентирована на студентов, магистрантов и аспирантов, обучающихся информатике.

УДК 004.932(075.8)

ББК 32.97я7

ISBN 978-985-550-770-4

© Старовойтов В. В., Голуб Ю. И., 2018

© Белорусский национальный
технический университет, 2018

ОГЛАВЛЕНИЕ

ПРЕДИСЛОВИЕ	8
Раздел I. ПОЛУЧЕНИЕ ЦИФРОВЫХ ИЗОБРАЖЕНИЙ	9
Глава 1. ЧТО ТАКОЕ ИЗОБРАЖЕНИЕ	9
1.1. Понятие изображения	9
1.2. Изображение как сигнал	9
1.3. Зрительная система человека	10
1.4. Описание изображений	15
1.4.1. Трехкомпонентная теория цвета	15
1.4.2. Оппонентная теория цвета	19
Глава 2. ФОРМИРОВАНИЕ ЦИФРОВЫХ ИЗОБРАЖЕНИЙ	20
2.1. Типы цифровых изображений	20
2.2. Преобразование непрерывного изображения в цифровое	21
2.2.1. Дискретизация изображений	21
2.2.2. Квантование изображений	23
2.2.3. Система кодирования цифрового изображения	27
Глава 3. СЖАТИЕ ИЗОБРАЖЕНИЙ	28
3.1. Алгоритмы сжатия без потерь	29
3.2. Алгоритмы сжатия с потерями	29
Глава 4. ФОРМАТЫ ПРЕДСТАВЛЕНИЯ ЦИФРОВЫХ ИЗОБРАЖЕНИЙ	36
Глава 5. УСТРОЙСТВА РЕГИСТРАЦИИ ЦИФРОВЫХ ИЗОБРАЖЕНИЙ	40
5.1. Сканер	40
5.2. Цифровой фотоаппарат	40
5.3. Получение изображений с помощью ультразвука	46
5.4. Регистрация рентгеновского изображения	48
5.5. Томографические изображения	49
5.5.1. Магнитно-резонансная томография	49
5.5.2. Компьютерная томография	51
5.5.3. Отличия изображения компьютерной томографии от магнитно-резонансной томографии	52
5.6. Формирование изображений дистанционного зондирования Земли	53

5.6.1. Формирование мультиспектральных изображений.....	54
5.6.2. Формирование гиперспектральных изображений.....	55
Глава 6. ВИДЕО И ФОРМАТЫ ВИДЕО	57
Выводы по разделу I	59
Раздел II. ОБРАБОТКА ЦИФРОВЫХ ИЗОБРАЖЕНИЙ	60
Глава 7. КЛАССИФИКАЦИЯ МЕТОДОВ ЦИФРОВОЙ ОБРАБОТКИ ИЗОБРАЖЕНИЙ	60
Глава 8. ГЕОМЕТРИЧЕСКИЕ ПРЕОБРАЗОВАНИЯ ИЗОБРАЖЕНИЙ	64
8.1. Аффинные преобразования.....	65
8.1.1. Сдвиг изображения.....	66
8.1.2. Отражение изображения относительно оси	67
8.1.3. Масштабирование изображения.....	67
8.1.4. Скос изображения.....	68
8.1.5. Поворот изображения.....	68
8.2. Проективные преобразования.....	70
8.3. Уточнение координат и яркости пикселей после преобразования	73
8.4. Геометрическая коррекция изображений	74
8.4.1. Построение мозаики из изображений.....	74
8.4.2. Коррекция проектированных искажений	79
8.5. Нелинейная геометрическая коррекция	80
8.5.1. Общий подход к вычислению параметров преобразования	80
8.5.2. Коррекция дисторсии	83
Глава 9. ПРЕОБРАЗОВАНИЯ ИЗОБРАЖЕНИЙ ТОЧЕЧНОГО ТИПА	87
9.1. Логические операции над изображениями	87
9.2. Линейные преобразования яркости полутоновых изображений	88
9.2.1. Препарирование изображений	88
9.2.2. Бинаризация изображений.....	90
9.2.3. Линейное растяжение гистограммы изображений.....	92
9.3. Арифметические операции над изображениями.....	94

9.4. Нелинейная коррекция яркости изображений.....	96
9.4.1. Гамма-коррекция	96
9.4.2. Логарифмическое преобразование яркости.....	97
9.4.3. Соляризация изображений.....	97
9.4.4. Приведение среднего к заданному значению	98
9.4.5. Табличный метод изменения яркости	100
9.4.6. Интерактивное изменение функции яркости изображения.....	101
9.5. Преобразования яркости на базе гистограммы изображения.....	103
9.6. Локально-адаптивная обработка изображений	107
9.6.1. Адаптивное повышение контраста	108
9.6.2. Адаптивное преобразование гистограммы	110
Глава 10. ПРЕОБРАЗОВАНИЯ ЛОКАЛЬНОГО ТИПА	112
10.1. Типы шумов.....	112
10.1.1. Импульсный шум	112
10.1.2. Аддитивный шум.....	113
10.1.3. Мультипликативный шум.....	115
10.1.4. Шум квантования	115
10.1.5. Спекл-шум.....	115
10.2. Фильтрация изображений.....	115
10.3. Низкочастотные фильтры.....	119
10.3.1. Арифметический усредняющий фильтр.....	119
10.3.2. Геометрический усредняющий фильтр	120
10.3.3. Гармонический усредняющий фильтр	121
10.3.4. Контргармонический усредняющий фильтр.....	121
10.3.5. Фильтр Гаусса.....	122
10.4. Высокочастотные фильтры	123
10.4.1. Фильтр Робертса	124
10.4.2. Фильтр Превитта	125
10.4.3. Фильтр Собела	125
10.4.4. Фильтр Лапласа	126
10.4.5. Выделение краев методом Кэнни.....	127
10.5. Нелинейная фильтрация	129
10.5.1. Медианная фильтрация.....	130
10.5.2. Адаптивная медианная фильтрация.....	133
10.5.3. Ранговая фильтрация.....	135

10.6. Внесение эффектов в изображение.....	135
10.6.1. Повышение резкости изображения.....	135
10.6.2. Создание эффекта тиснения.....	136
10.6.3. Создание эффекта акварелизации.....	137
10.7. «Продвинутые» алгоритмы.....	137
10.8. Сравнение фильтров.....	139
Глава 11. ОПЕРАЦИИ МАТЕМАТИЧЕСКОЙ	
МОРФОЛОГИИ	140
11.1. Математическая морфология бинарных	
изображений.....	141
11.1.1. Эрозия.....	141
11.1.2. Дилатация.....	143
11.1.3. Свойства дилатации и эрозии.....	144
11.1.4. Применение эрозии и дилатации.....	147
11.1.5. Отмыкание и замыкание.....	149
11.1.6. Применение операций отмыкания	
и замыкания.....	150
11.1.7. Операция утоньшения.....	150
11.2. Математическая морфология	
полутоновых изображений.....	153
Глава 12. МЕТОДЫ ОБРАБОТКИ ЦВЕТНЫХ	
ИЗОБРАЖЕНИЙ	155
12.1. Изменение цветового баланса.....	155
12.2. Идеальный отражатель.....	156
12.3. Преобразование «Серый мир».....	157
12.4. Линейное растяжение контраста	
цветного изображения.....	158
12.5. Автоматическое вычисление порога	
бинаризации цветного изображения.....	159
12.6. Повышение четкости цветного портрета.....	162
Глава 13. СЕГМЕНТАЦИЯ ИЗОБРАЖЕНИЙ	165
13.1. Сегментация изображений по яркости	
на основе пороговой обработки.....	166
13.1.1. Вычисление порогов по гистограмме.....	166
13.1.2. Алгоритм сегментации методом К-средних.....	168
13.1.3. Сегментация по принципу Вебера.....	168
13.1.4. Алгоритм сегментации Вежневца.....	170
13.1.5. Сегментация по цвету кожи.....	171

13.2. Статистический подход к текстурной сегментации.....	173
13.3. Сегментация методом морфологического водораздела	177
Выводы по разделу II	178
Раздел III. ПРИМЕРЫ ПРИМЕНЕНИЯ ЦИФРОВОЙ ОБРАБОТКИ ИЗОБРАЖЕНИЙ	179
Глава 14. РЕШЕНИЕ ПРИКЛАДНЫХ ЗАДАЧ	179
14.1. Повышение контраста рентгеновских изображений.....	179
14.2. Морфологическая фильтрация бинарных изображений.....	182
14.3. Коррекция фона при неравномерном освещении сцены при анализе изображений.....	182
14.4. Повышение разрешения мультиспектрального изображения	184
14.5. Детекция дорожных знаков.....	187
14.6. Автоматическое определение нефтяных пятен	189
14.7. Контроль отверстий в балках	190
14.8. Улучшение изображений в системах видеонаблюдения	192
14.8.1. Усиление контрастности.....	192
14.8.2. Коррекция динамического диапазона яркости	193
Глава 15. ПРОГРАММНЫЕ СРЕДСТВА ОБРАБОТКИ ЦИФРОВЫХ ИЗОБРАЖЕНИЙ	195
15.1. Писать код или взять из готовой библиотеки?	195
15.2. Система MATLAB.....	198
15.3. Открытая библиотека OpenCv	201
Выводы по разделу III.....	202
ЗАКЛЮЧЕНИЕ	203
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	204

ПРЕДИСЛОВИЕ

Что такое изображение? *Изображение* – это то, что мы видим, т. е. то, что может «зарегистрировать» зрительная система человека: например, окружающий нас мир, рисунок, чертеж, текст, телевизионное изображение, кривая на экране осциллографа и многое другое.

Таким образом, изображение – это отражение естественных (существующих в природе) сигналов, т. е. волн различной длины, которые могут быть зарегистрированы системой человеческого зрения (СЧЗ) или созданными человеком приборами. Разница между первыми и вторыми в том, что глаз человека воспринимает волны в достаточно узком участке электромагнитного спектра, который называется *видимым*. Все видимое глазом, а также регистрируемые приборами сигналы изначально представляют собой непрерывные сигналы. Они образуют так называемые аналоговые изображения, которые можно описать непрерывными функциями, а современные компьютеры обрабатывают информацию в цифровом виде.

Процесс регистрации отраженного непрерывного сигнала и его преобразование в цифровое представление называется *получением цифровых изображений*. Различные преобразования этих изображений называются *обработкой цифровых изображений*. Обработка выполняется по определенным алгоритмам, с помощью программ, реализующих эти алгоритмы.

Алгоритм – это набор точных инструкций, описывающих порядок действий исполнителя для достижения результата решения задачи за конечное время. 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 устройством управления вычислительной машины, т. е. это реализация алгоритма. Программа – это один из компонентов программного обеспечения.

Раздел I. ПОЛУЧЕНИЕ ЦИФРОВЫХ ИЗОБРАЖЕНИЙ

Глава 1. ЧТО ТАКОЕ ИЗОБРАЖЕНИЕ

1.1. Понятие изображения

В философском смысле слово «изображение» означает «отражение» и используется в смысле подобия тех или иных характеристик объектов. Уточним эти понятия.

Изображение – это образ, в той или иной степени подобный (но не идентичный) изображаемому объекту.

В области изобразительного искусства художественные произведения (рисунок, скульптура и проч.) являются, как правило, субъективным отражением реального мира.

Оптическое изображение – это картина, получаемая в результате прохождения через оптическую систему световых лучей, отраженных от объекта, и воспроизводящая его контуры и детали.

Зрение человека реализует процесс психофизиологической обработки изображения объектов окружающего мира, которое осуществляется зрительной системой, и позволяет получать представление о величине, форме и цвете предметов, их взаимном расположении и расстоянии между ними.

Итак, **изображение** – это то, что может целиком увидеть человеческий глаз или регистрирующая аппаратура. При этом следует различать понятия «изображение сцены» и «изображение объекта». Сцена состоит из реальных трехмерных объектов, либо на ней выделяются так называемые объекты интереса.

1.2. Изображение как сигнал

Так, наш глаз видит некий отраженный сигнал в определенном участке электромагнитного спектра.

Спектр (от лат. spectare – смотреть) – 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 в историческом смысле – распределение потока излучения или частиц по длинам волн или энергии.

Электромагнитные излучения подразделяются на (рис. 1.1) радиоволны (radio waves), инфракрасное излучение (infrared), видимый свет (visible light), ультрафиолетовое излучение (ultraviolet), рентгеновское (x-rays) и жесткое (гамма-излучение, gamma rays) излучение. Эта шкала называется *электромагнитным спектром*.

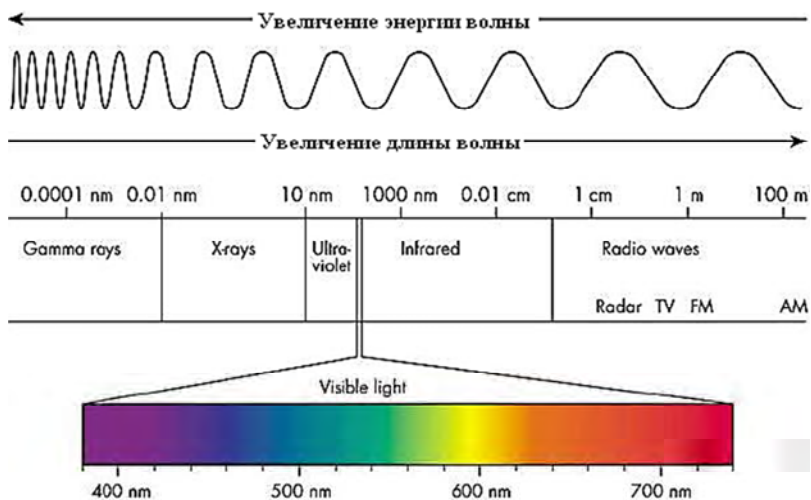


Рис. 1.1. Электромагнитный спектр

Водоем кажется синим потому, что молекулы воды поглощают все лучи видимого спектра, кроме синих. Но это заметно только при глубине более метра, поэтому лужи не выглядят синими.

1.3. Зрительная система человека

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

Свойства зрительной системы человека

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

обеспечивает адаптивную настройку 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 сетчатка, воспринимающая свет.

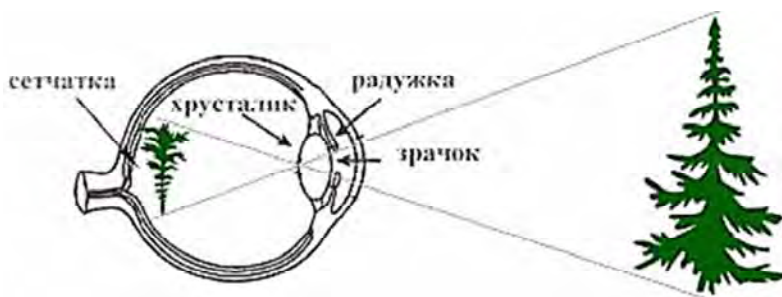


Рис. 1.2. Предмет, видимый глазом, отображается на сетчатке уменьшенным и перевернутым

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

Зрительная система по-разному реагирует на излучения, которые равны по мощности, но испускаются из различных диапазонов спектра. Такая чувствительность называется **спектральной**. Человеческий глаз воспринимает волны электромагнитного спектра в диапазоне от 380 до 750 нм.

Способность глаза различать минимальные различия яркости смежных областей изображения характеризуется **контрастной чувствительностью**. Также зрительная система характеризуется различной чувствительностью к цветовому тону, т. е. к излучениям из различных участков спектра. Еще зрительная система характеризуется **чувствительностью к насыщенности цвета**.

Приведенные выше типы чувствительности зрительной системы не являются постоянными, а зависят от многих факторов, в частности условий освещения. Например, при переходе из темной комнаты в светлую для восстановления светочувствительности глаза нужно некоторое время. Этот процесс называется **яркостной адаптацией глаза**.

Цветовосприятие характеризуется тремя основными характеристиками – светлота, цветовой тон и насыщенность.

У животных и человека органами зрения являются глаза. У человека и многих других животных существует бинокулярное зрение, обеспечивающее объемное изображение. В глазу человека содержатся два типа светочувствительных клеток (рецепторов): высокочувствительные палочки и менее чувствительные колбочки (рис. 1.3). Под влиянием света в палочках и колбочках происходит распад зрительных пигментов (родопсина и йодопсина). Палочки функционируют при свете слабой интенсивности, в сумерках. Зрительные ощущения, получаемые при этом, бесцветны. Колбочки функционируют днем и при ярком освещении. Их функция определяет ощущение цветности.

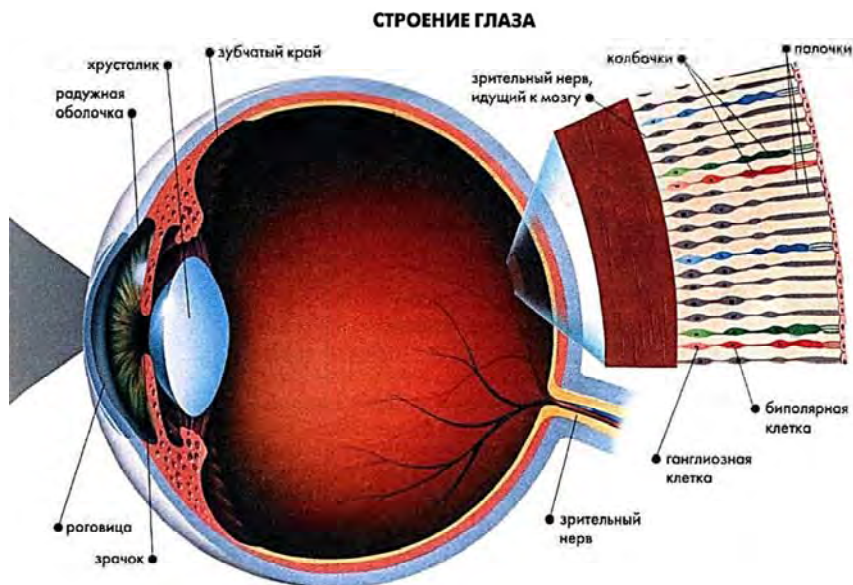


Рис. 1.3. Строение глаза человека

Палочки – это периферические отростки светочувствительных клеток сетчатки глаза, названные так за свою вытянутую цилиндрическую форму. В сетчатке глаза человека содержится около 120 миллионов палочек длиной 0,06 мкм и диаметром 0,002 мкм. Эти специализированные клетки преобразуют световые раздражения в нервное возбуждение.

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

Плотность размещения палочек на различных участках сетчатки глаза неравномерна и может составлять от 20 до 200 тысяч на квадратный миллиметр, 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 палочки 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), при этом палочки ответственны за сумеречное зрение, когда цвета предметов неразличимы.

Колбочки – другой тип фоторецепторов, названный так за свою коническую форму. Это также специализированные клетки, преобразующие световые раздражения в нервное возбуждение. Колбочки чувствительны к свету благодаря наличию в них пигмента йодопсина. В свою очередь, йодопсин состоит из нескольких зрительных пигментов. На сегодняшний день хорошо исследованы два пигмента: 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, диаметром от 1 до 4 мкм. Колбочки приблизительно в 100 раз менее чувствительны к свету, чем палочки, но гораздо лучше воспринимают быстрые движения.

По чувствительности к разным длинам волн света (цветам) (табл. 1.1, рис. 1.4) различают три вида колбочек. Колбочки S-типа чувствительны в фиолетово-синей (S от англ. *short* – коротковолно-

вый спектр), М-типа – в зелено-желтой (М от англ. *medium* – средневолновый) и L-типа – в желто-красной (L от англ. *long* – длинноволновый) частях спектра. Наличие трех видов колбочек и палочек (чувствительных в изумрудно-зеленой части спектра) дает человеку цветное зрение.

Таблица 1.1

Характеристики для различных видов колбочек

Тип колбочек	Воспринимаемые длины волн, нм	Максимум Чувствительности, нм
S	380–500	420
M	450–630	534
L	500–750	564

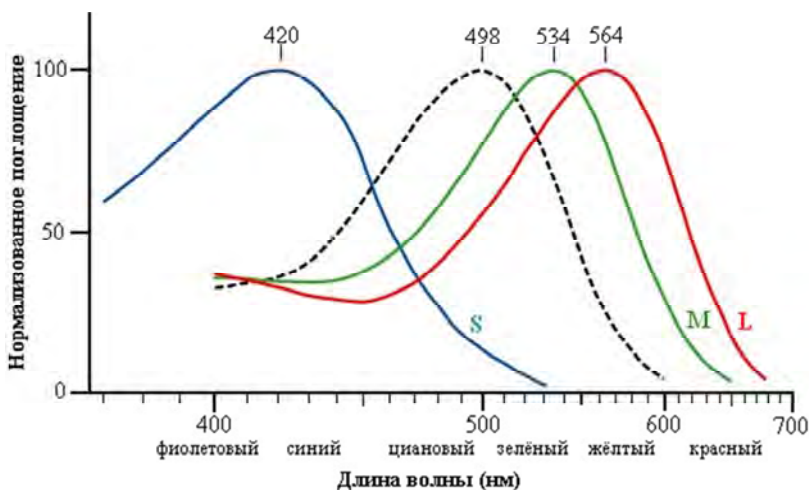


Рис. 1.4. Графики светочувствительности колбочек человеческого глаза S-, M-, L-типов. Пунктиром показана сумеречная, «черно-белая» восприимчивость палочек

В ночное время, когда поток фотонов недостаточен для нормальной работы колбочек, зрение обеспечивают только палочки, поэтому ночью человек не может различать цвета.

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

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

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

1.4. Описание изображений

1.4.1. Трехкомпонентная теория цвета

Трехкомпонентную теорию цветового зрения впервые описал в 1756 г. М. В. Ломоносов. 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 **цветовых моделей**. Термин «цветовая модель» означает абстрактную модель описания представления цветов в виде набора трех чисел, называемых цветовыми координатами. Вместе с методом интерпретации этих данных множество оттенков цвета цветовой модели определяет **цветовое пространство**.

Большинство цветов получается смешением нескольких основных цветов. Поэтому для описания цвета вводится понятие цвето-

вой модели как способа представления большого количества цветов посредством разложения его 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 человеком цвета Международная комиссия по освещению разработала стандарт CIE 1931 XYZ. Это цветовое пространство вмещало в себя все воспринимаемые человеком цвета, однако оно было существенно нелинейным.

В 1976 г. была разработана цветовая модель CIE Lab, которая является международным стандартом. В ней значение светлоты отделено от значения хроматической составляющей цвета (тон, насыщенность).

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 широко применяется при обработке изображений в качестве промежуточного цветового пространства, через которое происходит конвертирование данных между другими цветовыми пространствами (например, из RGB сканера в CMYK печатного процесса). 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, Saturation, Value – тон, насыщенность, значение) удобна для создания изображений художниками. В ней координатами цвета являются:

Hue – цветовой тон, обычно варьируемый в пределах $0-360^\circ$: 0° – красный, 120° – зеленый, 240° – синий (т. е. тон цвет задается в градусах);

Saturation – насыщенность, варьируемая в пределах $0-100$ или $0-1$;

Value – яркость, задаваемая в пределах $0-100$ или $0-1$.

Рассмотрим цветовую модель RGB. Если три составляющие имеют нулевые значения, получим черный цвет, если максимальные значения, – белый. Эта модель хороша для устройств с черным базовым фоном, например монитор (который излучает цвет). RGB – это аддитивная цветовая модель, предполагающая, что вся палитра цветов складывается из светящихся точек, ориентированная на устройства отображения излучательного типа. Отображение реального цвета в RGB на мониторе зависит от производителя монитора и настроек.

Если базовый фон белый, эту модель применять затруднительно: например, бумага поглощает цвет. В этом случае удобнее использовать субтрактивную модель, например CMY. «Субтрактивный» означает «вычитаемый», т. е. из белого вычитаются первичные цвета. Она является дополнительной к RGB и фактически определяет негативные оттенки, т. е. от максимального значения яркости в модели RGB вычитаются собственно оттенки цвета:

$$C = 255 - R; \quad M = 255 - G; \quad Y = 255 - B.$$

Эту модель активно используют в полиграфии при печати на белой бумаге. Теоретически равное количество краски $C = M = Y$ должно давать серый цвет, однако при печати такой цвет дает бурые оттенки и ведет к лишнему расходу краски. Поэтому на практике модель дополняют черным цветом, и она называется CMYK. Каждое из чисел, определяющее цвет в CMYK, обуславливает процент краски базового цвета, составляющей цветовую комбинацию. Например, для получения цвета хаки смешивают 30 частей голубой краски, 45 частей пурпурной краски, 80 – желтой и 5 – черной краски. 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: от размера пикселя на фотовыводе, растискивания краски, качества бумаги, влажности в цеху, спектральных характеристик источника освещения и других.

Рассмотрим важное понятие цветового охвата, который дает представление о том, насколько та или иная цветовая модель хорошо представляет цвета. *Цветовым охватом* называется диапазон цветов, который может быть воспроизведен или описан каким-либо образом.

Разными цветовыми охватами обладают электронно-лучевая трубка монитора или телевизора, цветовые модели, полиграфические краски и глаз человека. На рис. 1.5 показано сравнение цветовых охватов человеческого глаза (модель Lab), монитора (RGB) и печатной машины (СМУК) в системе координат $x = X/(X + Y + Z)$ и $y = Y/(X + Y + Z)$ в пространстве XYZ.

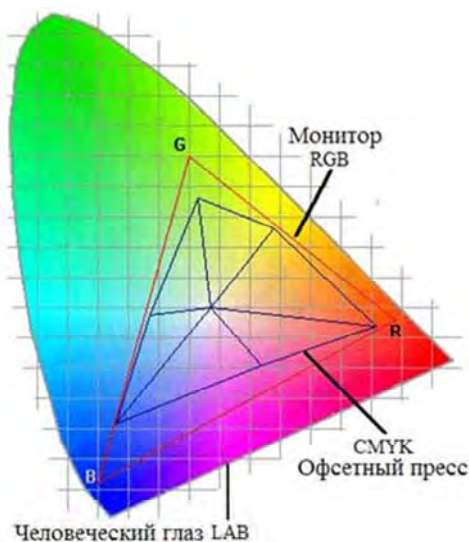


Рис. 1.5. Цветовой охват разных устройств

Разные цветовые модели используют 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, СМУК);
модели, не связанные с физикой оборудования, ориентированные на передачу данных;
математические модели для цветокоррекции, но не связанные с оборудованием (например, HSV).
Цветовые модели можно однозначно преобразовать друг в друга.

1.4.2. Оппонентная теория цвета

Параллельно трехкомпонентной существовала **оппонентная теория цвета** Эвальда Геринга. Ее развили Дэвид Хьюбел и Торстен Визел. За свое открытие они получили Нобелевскую премию 1981 г. Ученые предположили, что в мозг поступает информация не о красном (R), зеленом (G) и синем (B) цветах (как в теории цвета Гельмгольца). Мозг получает информацию о разнице яркости (I_{\max}) и черного (I_{\min}), разнице зеленого и красного цветов ($G - R$), разнице синего и желтого цветов ($B - Y$). Например, желтый цвет ($Y = R + G$) – это сумма красного и зеленого цветов, где R , G и B – яркости цветковых составляющих, красного R , зеленого G и синего B . Имеем систему коэффициентов:

$$K_{ч-б} = I_{\max} - I_{\min};$$

$$K_{gr} = G - R;$$

$$K_{brg} = B - R - G,$$

где $K_{ч-б}$, 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 называется **цветовой адаптацией**.

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

Глава 2. ФОРМИРОВАНИЕ ЦИФРОВЫХ ИЗОБРАЖЕНИЙ

2.1. Типы цифровых изображений

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 двумерный массив, элементы которого содержат информацию о цвете, и ориентировано на растровые устройства отображения. В цифровой обработке используются растровые изображения, поэтому векторные мы не рассматриваем.

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

Элементы **бинарного** (двухуровневого) изображения описываются одним битом, который принимает значения «0» и «1». Значение «0» соответствует черному цвету, «1» – белому цвету. Они получаются в результате обработки (бинаризации) полутоновых или цветных изображений. Бинарные изображения удобны при передаче изображений документов в факсимильной связи.

Полутоновое изображение состоит из пикселей, 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 пиксель.

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

В **палитровых** (индексированных) изображениях значение пикселя, т. е. индекс, фактически является ссылкой на ячейку палитры, которая представляет собой таблицу RGB компонентов для этого индекса. Для описания таких изображений необходимо хранить палитру (т. е. табл. 256 цветов) и собственно изображение (1 бит на пиксель). При использовании неравномерных алгоритмов квантования зрительная система человека практически не различает изображения, представленные в полноцветном и индексированном вариантах.

Выбор типа изображения зависит от решаемой задачи. В зависимости от типа изображения они по-разному представляются в разных форматах. Основные типы цифровых изображений и их характеристики представлены в табл. 2.1.

Таблица 2.1

Основные типы цифровых изображений

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

2.2. Преобразование непрерывного изображения в цифровое

Процесс преобразования аналогового сигнала в цифровую форму состоит из трех этапов: 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 меняются в непрерывном диапазоне).

2.2.1. 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, т. е. теоретически множество допустимых значений бесконечно.

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

Пусть $x(t_1, t_2)$ – непрерывное изображение, а $x(i_1, i_2)$ – соответствующее ему дискретное, полученное из непрерывного путем прямоугольной дискретизации. Это означает, что связь между ними определяется выражением

$$x(i_1, i_2) = x(i_1\Delta t_1, i_2\Delta t_2), \quad (2.1)$$

где $\Delta t_1, \Delta t_2$ – соответственно вертикальный и горизонтальный шаги или интервалы дискретизации.

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 восстановить непрерывный сигнал. С математической точки зрения вопрос, таким образом, заключается в восстановлении непрерывного сигнала в двумерных промежутках между узлами, в которых его значения известны или, иными словами, в осуществлении двумерной интерполяции.



Рис. 2.1. Влияние интервала дискретизации на представление изображения отпечатка пальца

На рис. 2.1 изображен один и тот же отпечаток при разных интервалах дискретизации. Правое изображение на рис. 2.1, *б* сильно расфокусировано и неверно передает структуру отпечатка, оно не пригодно для распознавания.

2.2.2. Квантование изображений

При цифровой обработке изображений непрерывный динамический диапазон значений яркости делится на ряд дискретных уровней и значения яркости, попавшие в один диапазон, округляются до одного значения, например середины этого диапазона. Такая процедура называется *квантованием по яркости*. Она преобразует непрерывную переменную x в дискретную переменную $x_{\text{кв}}$, принимающую конечное множество значений $\{r_1, \dots, r_L\}$. Эти значения называются *уровнями квантования*. В общем случае преобразование выражается ступенчатой функцией (рис. 2.2). Если яркость x отсчета изображения $d_j < x \leq d_{j+1}$, то исходный отсчет заменяется на уровень квантования r_j , где $d_j, j = \overline{1, L+1}$ – пороги квантования. При этом динамический диапазон значений яркости 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.

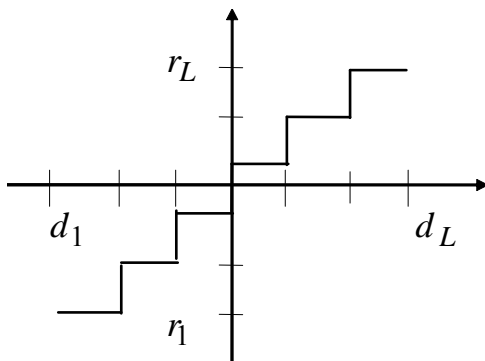


Рис. 2.2. Функция, описывающая равномерное квантование

Не следует путать процедуры квантования и дискретизации (и соответственно шаг квантования с частотой дискретизации). При дискретизации выполняется пространственное разбиение области изображения, при квантовании – дискретизации подвергается множество значений яркости.

Квантование можно выполнить разными способами.

Квантование методом равномерного разбиения цветового пространства

Рассмотрим самый простой вариант квантования – метод равномерного разбиения цветового пространства.

Пусть $I(x, y)$ – яркость пикселя исходного изображения с координатами (x, y) ;

$I_{new}(x, y)$ – яркость пикселя нового изображения;

$N1$ – количество оттенков в исходном изображении (обычно $N1 = 256$);

$N2$ – количество элементов в новой палитре (например $N2 = 16$).

Алгоритм равномерного разбиения цветового пространства

Шаг 1. В цикле для каждого пикселя (x, y) вычислить новое значение яркости.

Шаг 2. $I_{new}(x, y) = I(x, y) / (N1 / N2)$.

Шаг 3. Результат округлить до целого.

Для цветного изображения повторить алгоритм 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 RGB-пространстве. Каждый оттенок можно закодировать целым индексом от 0 до 255. Описав в таблице, какой цвет оно представляет, получают ***индексированное цветное изображение***.

Достоинства этого алгоритма – 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 атрибутов пикселей в исходном изображении. Это достигается путем последовательного разбиения цветового пространства на параллелепипеды со сторонами, параллельными осям цветового пространства RGB. Это неравномерное квантование.

Шаг 1. Задать N – размер палитры. Найти минимальный параллелепипед, включающий все значения цвета пикселей исходного изображения. Далее выполняется разбиение параллелепипеда 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, получают два параллелепипеда, которые содержат равное количество значений.

Предыдущую процедуру повторяют для каждого из параллелепипедов до тех пор, пока не сформируется N параллелепипедов, где N равно заданному размеру палитры.

Если же на каком-то этапе потребуются разделить параллелепипед, содержащий лишь одно значение, то получившийся пустой параллелепипед следует присоединить к самому большому параллелепипеду для последующего разделения.

Шаг 3. Заполнить палитру либо центральными точками параллелепипедов, либо средними арифметическими значений, попавших в данные параллелепипеды.

Шаг 4. Для каждого пикселя его RGB-атрибуты заменить параметрами параллелепипедов, вычисленными на *шаге 3*.

Результатом работы этого алгоритма является хорошее квантование, при этом скорость работы данного алгоритма высока.

Квантование изображений методами кластеризации

Квантование можно выполнять методами, производящими кластеризацию цветовых значений пикселей изображения для построения палитры. В общем случае **кластеризация** – это процесс разбиения объектов на группы (кластеры) на основе общих свойств объектов. Применительно к квантованию изображений это означает разбиение всех оттенков изображения на кластеры так, что внутри каждого кластера находятся лишь близкие значения. Это также способ неравномерного квантования.

Один из самых популярных алгоритмов для 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 сопоставляются кластерам так, что средние в кластерах максимально отличаются друг от друга.

Алгоритм кластеризации методом k -средних

Исходные данные: цветное изображение, число кластеров K , максимальное число итераций N .

Шаг 1. Определить начальные центры кластеров. Выбор координат центров можно осуществлять разными способами, например случайно, либо выбрать значения оттенков R , G , B первых K пикселей.

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

Шаг 2. Вычислить новые координаты центров всех K кластеров, которыми считаются среднее арифметическое всех значений R , G , B оттенков цвета, входящих в кластер.

Цвета всех пикселей изображения перераспределить по K кластерам, вычислив расстояния между ними и найденными центрами.

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

- 1) центры всех кластеров не меняются, т. е. цвета принадлежат тем же кластерам, что и на предыдущей итерации;
- 2) число итераций равно максимальному заданному числу итераций N .

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

Достоинством является высокая скорость работы. Применительно к квантованию изображений данный метод показывает очень хорошие результаты.

2.2.3. Система кодирования цифрового изображения

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

Понятие *пиксель* (англ. pixel, pel – сокращение от picture cell – элемент изображений) имеет два значения:

1) наименьший *логический* элемент двумерного цифрового изображения,

2) *физический* элемент светочувствительной матрицы и элемент матрицы дисплеев, формирующих изображение.

Пиксель представляет собой неделимый объект прямоугольной или круглой формы, характеризуемый определенным цветом (в плазменных панелях он может быть восьмиугольным).

Таким образом, цифровое изображение можно трактовать как матрицу, элементы которой – пиксели, содержащие целочисленные значения из некоторой палитры. Отметим, что эти значения безразмерны, поскольку имеют абстрактный характер.

Глава 3. СЖАТИЕ ИЗОБРАЖЕНИЙ

Изображения занимают большой объем памяти, и их компактное хранение предполагает сжатие исходных данных. Например, при дистанционном зондировании Земли с помощью спутников сжатие данных представляет серьезную проблему. По спутниковым каналам передается до 300 мегабит в секунду. Поэтому алгоритмы сжатия прежде всего ориентированы на устранение избыточности двумерных многоспектральных данных таким образом, чтобы иметь возможность в случае необходимости полностью восстановить исходное изображение. Методы сжатия изображений можно разбить на две группы:

сжатие без потерь, позволяющее точно восстанавливать исходное изображение;

сжатие с потерями (или сжатие с регулируемой степенью искажений).

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

кодирование длин серий (RLE);

метод группового кодирования (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. Таким образом, восстановить исходные значения яркости или цвета всех пикселей 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 информации. Этот метод применяется в телевидении. Разбиение непрерывного видеопотока на кадры – также пример исключения психофизиологической избыточности.

3.1. 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 (хотя может сохранять изображение и без архивации). Здесь 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 байт. Алгоритм Хаффмана обеспечивает сжатие с помощью предварительного вычисления частоты появления одинаковых байтов в изображении.

Говоря о лучших на сегодня алгоритмах сжатия изображений без потерь, следует отметить, что в 1994 г. был принят стандарт ISO/JPEG- Lossless. Он ориентирован на полноцветные 24- или 8-битовые в градациях яркости изображения без палитры. На 8-битовых тестовых изображениях нескольких категорий (космических, медицинских, фото- и компьютерно-графических, смешанных) средний уровень сжатого по JPEG-Lossless изображения составил 3,98 бит/пиксель.

3.2. Алгоритмы сжатия с потерями

The Joint Photographic Experts Group (JPEG), подразделение в рамках ISO – Международной организации по стандартизации, разработало международный стандарт JPEG для полноцветных (24-битовых) изображений. Он содержит один из мощных алгоритмов сжатия. Для него реализованы четыре варианта сжатия:

– последовательное сжатие (на базе дискретного косинусного преобразования (ДКП));

- прогрессивный вариант ДКП-сжатия, использующий пирамидальное представление изображений;
- последовательное, основанное на предсказании сжатия без потерь (JPEG-Losless);
- иерархическое сжатие, соответственно с потерями и без потерь.

Оперирует алгоритм областями 8×8 , на которых яркость и цвет меняются сравнительно плавно. Вследствие этого при разложении матрицы такой области в двойной ряд по косинусам значимыми оказываются только первые коэффициенты. Кроме того, система цветового восприятия человека слабо распознает определенные частоты (цветовое изменение картинки). Поэтому можно аппроксимировать некоторые коэффициенты более грубо без заметной потери качества изображения, используя для этого квантование коэффициентов. При таком преобразовании теряется часть информации, но могут достигаться большие коэффициенты сжатия.

Основные шаги алгоритма для сжатия 24-битового цветного изображения.

Шаг 1. Преобразовать изображение из цветового пространства RGB в пространство YCrCb. В нем Y – яркостная составляющая, а Cr, Cb – компоненты, отвечающие за цвет (хроматический красный и хроматический синий). Человеческий глаз менее чувствителен к вариациям цветовой компоненты, чем к яркости, поэтому можно проредить массивы Cr и Cb. Каждому блоку из четырех пикселей (2×2) яркостного канала Y ставятся в соответствие усредненные значения Cb и Cr (схема прореживания $4 : 2 : 0$). При этом для каждого блока 2×2 вместо 12 значений (4 Y, 4 Cb и 4 Cr) используется всего 6 (4 Y и по одному усредненному Cb и Cr). Если к качеству восстановленного после сжатия изображения предъявляются повышенные требования, прореживание может выполняться лишь в каком-то одном направлении – по вертикали (схема $4 : 4 : 0$) или по горизонтали ($4 : 2 : 2$) или не выполняться вовсе ($4 : 4 : 4$). Стандарт допускает и другие варианты прореживания.

Перевод из RGB в YCrCb выполняется по формуле.

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0,299 & 0,587 & 0,114 \\ 0,5 & -0,4187 & 0,0813 \\ 0,1687 & -0,3313 & 0,5 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}. \quad (3.1)$$

Обратное преобразование осуществляется умножением вектора YUV на обратную матрицу по формуле.

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1,402 \\ 1 & -0,34414 & -0,71414 \\ 1 & 1,722 & 0 \end{bmatrix} \times \begin{bmatrix} Y \\ U \\ V \end{bmatrix} - \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}. \quad (3.2)$$

Шаг 2. Разбить исходное изображение на матрицы 8×8 . Сформировать из каждой три рабочие матрицы ДКП – по 8 бит отдельно для каждой компоненты.

Шаг 3. Применить ДКП к каждой рабочей матрице. При этом получаем матрицу, в которой коэффициенты в левом верхнем углу соответствуют низкочастотной составляющей изображения, а в правом верхнем – высокочастотной.

Шаг 4. Выполнить квантование коэффициентов делением рабочей матрицы на матрицу квантования поэлементно. Для каждой компоненты (Y, U и V) в общем случае задать свою матрицу квантования (МК) $q[u, v]$, например, для Y:

$$Y_q[u, v] = \text{IntegerRound} \left(\frac{Y[u, v]}{q[u, v]} \right). \quad (3.3)$$

На этом шаге осуществляется управление степенью сжатия и происходят самые большие потери. Задавая МК с большими коэффициентами, получаем больше нулей и, следовательно, большую степень сжатия. В стандарт JPEG включены рекомендованные разработчиками МК, построенные опытным путем. Матрицы для большего или меньшего коэффициентов сжатия получают путем умножения исходной матрицы на некоторое (рекомендованное) число *gamma*.

С квантованием связаны и специфические эффекты алгоритма. При больших значениях числа *gamma* потери в низких частотах могут быть настолько велики, что восстановленное изображение распадется на квадраты 8×8 . Потери в высоких частотах могут проявиться в так называемом эффекте Гиббса, когда вокруг контуров с резким переходом цвета образуется своеобразный нимб.

Шаг 5. Перевести матрицу квантованных коэффициентов 8×8 в 64-элементный вектор при помощи «зигзаг»-сканирования, когда берутся элементы с индексами [0, 0], [0, 1], [1, 0], [2, 0], [1, 1], [0, 2], [0, 3], [1, 2]... В результате в начале вектора получить трансформанты, соответствующие низким частотам, а в конце – высоким.

Шаг 6. Свернуть вектор с помощью алгоритма группового кодирования (RLE). При этом получаются высокие частоты типа «пропустить, число», где «пропустить» является счетчиком пропускаемых нулей, а число – значением, которое необходимо поставить в следующую ячейку. Так, вектор 42 3000–2 0000 1 ... будет свернут в $D_s(0, 42)(0, 3)(3, 2)(4, 1)$.

Шаг 7. Свернуть полученные D_s кодированием по алгоритму Хаффмана с фиксированной кодовой таблицей.

При сохранении изображения в файле указывается параметр качества, задаваемый в условных единицах, например от 1 до 100 или от 1 до 10. 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 не будет в точности совпадать с исходным, что связано как с конечной точностью вычисления ДКП, а также с необходимостью округления значений Y , C_b , C_r и коэффициентов ДКП до целых значений.

Вариант Lossless JPEG не использует ДКП, что обеспечивает точное совпадение восстановленного и исходного изображений, однако его малая эффективность (коэффициент сжатия редко превышает 2) и отсутствие поддержки со стороны разработчиков программного обеспечения не способствовали популярности Lossless JPEG.

Процесс восстановления изображения в этом алгоритме полностью симметричен.

Метод позволяет сжимать некоторые изображения в 10–15 раз без визуально малозаметных потерь.

Алгоритм JPEG разработан для сжатия фотографий, содержащих сцены с плавными переходами яркости и цвета. Наибольшее распространение JPEG получил в сети Интернет. Он малоприменим для сжатия чертежей и текста, имеющих резкий контраст между соседними пикселями, что приводит к появлению заметных артефактов. Такие изображения целесообразно сохранять в форматах без потерь, таких как TIFF, GIF или PNG.

JPEG не должен использоваться и в тех случаях, когда недопустимы даже минимальные потери, например, при сжатии астрономических или медицинских изображений. В таких случаях можно использовать предусмотренный стандартом JPEG режим сжатия Lossless JPEG.

Стандарт JPEG предусматривает два основных способа представления кодируемых данных.

Наиболее распространенным, поддерживаемым большинством доступных кодеков, является последовательное представление данных, предполагающее последовательный обход кодируемого изображения поблочно слева направо, сверху вниз. Над каждым кодируемым блоком изображения осуществляются описанные выше операции, а результаты кодирования помещаются в выходной поток в виде единственного «скана», т. е. массива кодированных данных, соответствующего последовательно пройденному («просканированному») изображению.

Второй способ (прогрессивный JPEG) означает, что сжатые данные записываются в выходной поток в виде набора сканов, каждый из которых описывает изображение полностью со все большей степенью детализации. Это достигается либо путем записи в каждый скан не полного набора коэффициентов ДКП, а лишь какой-то их части: сначала – низкочастотных, в следующих сканах – высокочастотных, либо путем последовательного, от скана к скану, уточнения коэффициентов ДКП. Прогрессивное представление данных позволяет увидеть грубый вариант изображения уже после передачи незначительной части JPEG-файла.

JPEG 2000 (или **jp2**) – алгоритм, в котором вместо ДКП (применяемого в формате JPEG) используется вейвлет-преобразование, которое основано на представлении сигнала в виде суперпозиции базовых функций – вейвлетов.

В результате сжатия этим алгоритмом после восстановления изображение получается более гладким и четким, а размер сжатого файла по сравнению с JPEG при одинаковом качестве оказывается меньшим. JPEG 2000 полностью свободен от главного недостатка JPEG: благодаря использованию вейвлетов изображения, при высоких степенях сжатия, не содержат артефактов в виде «решетки» из блоков размером 8×8 пикселей. Формат JPEG 2000, как и JPEG, поддерживает так

называемое прогрессивное сжатие, позволяющее по мере загрузки видеть сначала размытое, а затем все более четкое изображение.

Основные преимущества JPEG 2000 по сравнению с JPEG

1. Большая степень сжатия: на высоких битрейтах, где артефакты незначительны, JPEG 2000 имеет степень сжатия в среднем на 20 % больше, чем JPEG. На низких битрейтах JPEG 2000 также имеет преимущество над основными режимами JPEG. Большая степень сжатия достигается благодаря использованию дискретного вейвлет-преобразования и более сложного энтропийного кодирования.

2. Масштабируемость фрагментов изображений: JPEG 2000 обеспечивает бесшовное сжатие разных компонентов изображения, с каждым компонентом хранится от 1 до 16 бит на семпл. Благодаря разбиению на блоки изображения разных разрешений можно хранить в одном кодовом потоке.

3. Произвольный доступ к кодовому потоку, также иногда называемый доступом к областям интереса: кодовый поток JPEG 2000 обеспечивает несколько механизмов для поддержки произвольного доступа, также поддерживается несколько степеней разбиения на части (области интереса).

4. Гибкий формат файла: форматы файлов JP2 и JPX обеспечивают хранение информации о цветовых пространствах, метаданных и информации для согласованного доступа в сетевых приложениях, взаимодействующих с помощью протокола JPEG Part 9 JPIP.

Артефакты, возникающие при сжатии по алгоритму JPEG 2000 с высокой степенью компрессии (потерь), качественно мало отличаются от артефактов, возникающих при сжатии компрессором JPEG в тех местах, где оригинальное изображение имело плавные цветовые переходы: они становятся еще более плавными (размытыми). В тех же местах, 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 пикселей) отступающего от более темного участка перехода. Различия в артефактах:

нет сетки в 8×8 пикселей;

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 фр. *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 эффективным, если необходимо передать все нюансы оформления, например исторических документов, где важное значение имеет 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 специальная технология, разделяющая исходное изображение на три слоя: передний план, задний план и черно-белую (однобитовую) маску. Маска сохраняется с разрешением исходного файла; 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.

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 исходное изображение. В присутствии сложного заднего плана выигрыш в объеме обычно составляет 4–10 раз. Однако при стандартных настройках в DjVu используется сжатие данных с потерями, поэтому для особо важных документов обычно используются форматы сжатия без потерь: PNG, JPEG 2000, TIFF и т. п. В DjVu также можно использовать и сжатие данных без потерь. Так, например, утилита CJB2 предоставляет сжатие без потерь.

В основе алгоритма DjVu лежат несколько алгоритмов: алгоритм отделения текста от фона на отсканированном изображении;

вейвлетный алгоритм сжатия фона IW44;
алгоритм сжатия черно-белых изображений JB2;
универсальный алгоритм сжатия ZIP;
алгоритм распаковки «по запросу»;
алгоритм «маскировки» изображений.

Глава 4. ФОРМАТЫ ПРЕДСТАВЛЕНИЯ ЦИФРОВЫХ ИЗОБРАЖЕНИЙ

Изображения должны эффективно храниться, обрабатываться и экспортироваться. 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 файла часто указывается в расширении имени файла, отделенном точкой. Например, окончание имени (расширение) «.tif» используют для обозначения файлов, содержащих изображение, описанное в формате TIFF, а «.doc» – содержащих текстовую информацию, структурированную в соответствии со стандартами программы Microsoft Word.

Для многих форматов файлов существуют опубликованные спецификации, в которых подробно описаны структура файлов данного формата и то, как программы должны кодировать данные для записи в этот формат и как декодировать их при чтении.

Для того чтобы правильно работать с файлами, программы должны иметь возможность определять их тип. Исторически в разных операционных системах используются разные подходы для решения этой задачи.

На данный момент времени существует большое количество различных форматов графических файлов, предназначенных для разнообразных целей. Наиболее распространенные форматы и их особенности представлены в табл. 4.1.

**Форматы графических файлов, их описание
и область применения**

Формат	Особенности	Применение
RAW	Содержит необработанную информацию с матрицы фотокамеры, может быть сжат без потери качества	
BMP	Хранит данные с глубиной цвета до 48 бит и максимальным размером 65 535 × 65 535 пикселей 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 проч. с глубиной цвета (8, 16, 32 и 64 бит)	Полиграфия, geoTIFF используется для хранения мультиспектральных спутниковых изображений
JPEG	JPEG поддерживает до 16 млн цветов и сохраняет их в фотографиях. JPEG 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	Формат PNG-24 также поддерживает 16 млн цветов. Подобно формату JPEG, сохраняет яркость и оттенки цветов в фотографиях. Подобно GIF и формату PNG-8, сохраняет детали изображения (например, в линейных рисунках, логотипах или иллюстрациях)	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. Важным достоинством этого формата является его 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, в случае необходимости TIFF позволяет сохранять разнообразную дополнительную информацию о изображении (контуры, альфа-каналы и т. п.). В формате TIFF может быть использовано сжатие по алгоритму LZW.

TIFF является лучшим выбором при хранении изображений с высоким качеством, предназначенных для дальнейшего анализа и обработки, импорте растровой графики в векторные программы, издательские системы и т. п.

Не зависящий 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): в него были добавлены поддержка прозрачности и анимации изображений. Формат GIF использует LZW-компрессию и имеет ограниченную палитру в 256 цветов. Таким образом, 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 утрачена).

GIF используется как один из основных растровых форматов для представления графической информации в интернете.

Формат JPEG ориентирован на одноименный алгоритм сжатия для компрессии изображений. Он описывает структуру представления после их сжатия алгоритмом JPEG и предназначен для хранения полноцветной графики с высоким коэффициентом сжатия без заметной для глаза человека потери качества. При этом, как было сказано выше, изображение не может быть полностью восстановлено в исходном виде. При сохранении графики в формате JPEG можно управлять параметром качества (quality), измеряемым в относи-

тельных единицах – чаще всего от 0 (максимальное сжатие) до 100 (максимальное качество).

Формат JPEG следует использовать для хранения полноцветных изображений (в первую очередь фотографических и т. п.), которые не предназначены для дальнейшей обработки. Его не следует применять для хранения изображений с небольшим количеством цветов (менее 256) и с резким переходами между цветами, так как это может привести к проявлению различных глазом дефектов.

Формат BMP (Windows Device Independent Bitmap, от BitMaP – битовая карта) является собственным растровым форматом операционной системы Windows и способен хранить как изображения в индексированном виде (с палитрой до 256 цветов), так и полноцветные RGB-изображения (с глубиной цвета 24 бита). Возможно применение алгоритма сжатия RLE. Применение данного формата ограничено тем, что он не поддерживается графическими программами под управлением большинства операционных систем, кроме Windows. Формат используется в основном для хранения картинок при организации пользовательских интерфейсов.

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

Глава 5. УСТРОЙСТВА РЕГИСТРАЦИИ ЦИФРОВЫХ ИЗОБРАЖЕНИЙ

5.1. Сканер

Сканер – это устройство, которое при помощи аналого-цифрового преобразователя создает цифровое описание изображения внешнего для ЭВМ объекта и передает его посредством системы ввода/вывода в ЭВМ. Процесс получения этой копии называется *сканированием*.

Сканеры бывают ручные, рулонные, планшетные и проекционные. Разновидностью проекционных сканеров являются слайд-сканеры, предназначенные для сканирования фотопленок. В высококачественной полиграфии применяются барабанные сканеры, в которых в качестве светочувствительного элемента используется фотоэлектронный умножитель.

Принцип работы однопроходного планшетного сканера состоит в том, что вдоль сканируемого изображения, 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 три расположенных параллельно друг другу фоточувствительных полупроводниковых элемента, каждый из которых принимает информацию о компонентах изображения.

5.2. Цифровой фотоаппарат

Цифровой фотоаппарат по конструкции схож с обычным фотоаппаратом, но вместо фотопленки, на которой фиксируется изображение, у него имеются специальные датчики (фотоматрицы): матричные приборы с зарядовой связью – ПЗС (Charge Coupled Device, CCD) или комплементарные металлооксидные полупроводниковые приборы КМОП (Complementary Metal Oxide Semiconductor, CMOS). Эти датчики преобразуют изображение, проецируемое на них объективом, в цифровую форму.

Две качественные характеристики, напрямую влияющие на результат съемки, – **динамический диапазон** сенсора и **разрядность** представления цвета. Первая из этих характеристик отражает способность матрицы передавать световые оттенки, вторая относится

не только к самому сенсору, но и к аналого-цифровому преобразователю, переводящему электрические сигналы с выводов матрицы в цифровой код.

Динамический диапазон – это количество оттенков серого (т. е. уровней яркости), которые способен различить светочувствительный материал (фотопленка или сенсор цифровой камеры) между абсолютно черным и абсолютно белым цветом. Чем выше динамический диапазон, тем выше достоверность изображения на экспонированном носителе. Самым высоким динамическим диапазоном обладает негативная фотопленка. Поэтому до сих пор, несмотря на достижения цифровых технологий, для демонстрации фильмов в кинотеатрах используются обычные пленочные, а не цифровые проекторы.

Существует несколько вариантов конструкции цифровых фотоаппаратов: камеры с задней разверткой (рис. 5.1), трехкадровые камеры (рис. 5.2) и однокадровые камеры с одной или тремя фотоматрицами (рис. 5.3 и 5.4). В основном применяются однокадровые фотокамеры.

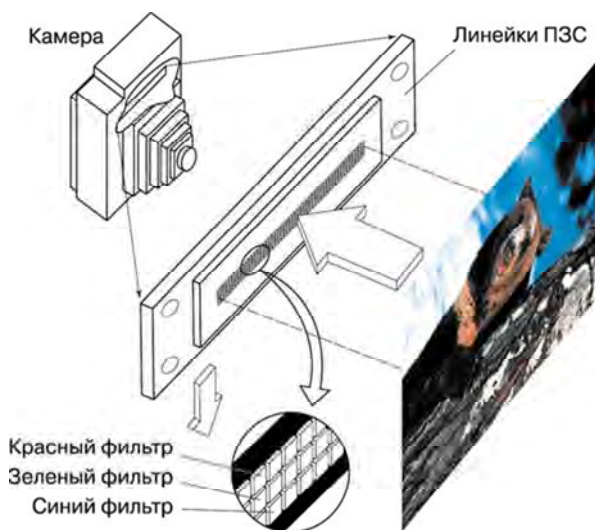


Рис. 5.1. Принцип действия камеры с задней разверткой

Камера с задней разверткой производит сканирование в плоскости изображения. Принцип работы такой камеры (см. рис. 5.1) напо-

минает технологию сканирования, реализованную в сканерах с построчным считыванием информации. Сканирующая головка, содержащая линейку светочувствительных ПЗС, перемещается с небольшим шагом вдоль задней фокальной плоскости камеры поперек изображения, за каждый шаг регистрируя одну строку пикселей. Камеры, в которых применяется такой принцип, позволяют получать изображения с высоким разрешением, но время экспозиции может достигать нескольких минут, что делает технику задней развертки непригодной для съемки движущихся объектов или при работе со вспышкой. Во время сканирования затвор камеры остается открытым, поэтому необходимо применять постоянное освещение, так как ни вспышка, ни стробоскоп в данном случае не годятся.

Трехкадровая камера предназначена для регистрации цветных изображений неподвижных объектов (рис. 5.2). В качестве светочувствительного датчика используется двумерная матрица ПЗС. Плоские двумерные матрицы имеют гораздо меньшее разрешение, чем линейные. Каждый элемент матрицы формирует одну точку изображения. Экспозиция производится с такой скоростью, что можно пользоваться освещением от обычной вспышки.

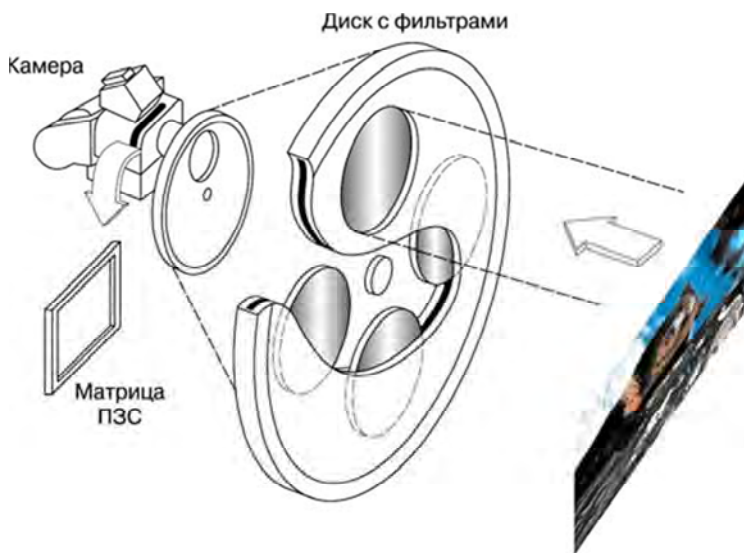


Рис. 5.2. Принцип действия трехкадровой камеры

Для регистрации цветного изображения нужно сделать три отдельных снимка через три светофильтра (красный, зеленый и синий). Между экспонированиями диск со светофильтрами поворачивается таким образом, чтобы в момент съемки перед матрицей находился или красный, или зеленый, или синий фильтр.

Технология, реализованная в однокадровых камерах с одной матрицей, обеспечивает высокую скорость оцифровывания 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. 5.3) применяется плоская матрица, но данные о цвете регистрируются не через отдельные фильтры, а через нанесенный на поверхность ПЗС-матрицы пленочный фильтр, состоящий из красных, зеленых и синих элементов. Данные о каждой точке 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.

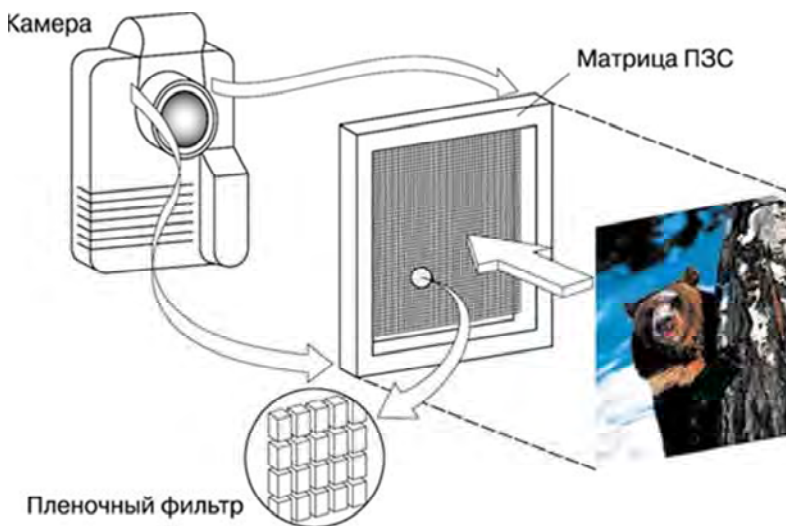


Рис. 5.3. Принцип действия однокадровой камеры с одной матрицей

Принцип действия однокадровой камеры с тремя матрицами (камеры для однокадровой цветной съемки), в состав которой входят три матрицы, заключается в расщеплении приходящего света на красную, зеленую и синюю составляющие, причем каждая из них направляется на свою матрицу (рис. 5.4). 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.

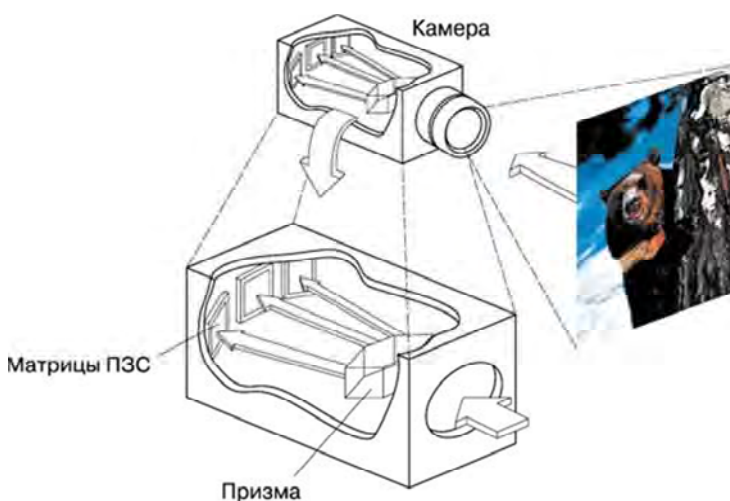


Рис. 5.4. Принцип действия однокадровой камеры с тремя матрицами

При использовании трехслойных фотоматриц ХЗ (рис. 5.5) компании Foveon цветоразделение на основные цвета RGB проводится в толщине полупроводникового материала послойно, с применением физических свойств кремния, заключающихся в том, что с увеличением длины волны световых волн растет и глубина их проникновения в кремний (рис. 5.5, а). Фотодиоды, созданные чередованием зон проникновения света, размещают один под другим на характерных глубинах для улавливания фотонов синего, зеленого и красного

цветов. Синяя часть спектра поглощается верхним слоем (толщина 0,2 мкм), зеленая – средним (толщина 0,4 мкм), а красная – нижним (более 2 мкм). Толщина каждого слоя выбрана по результатам экспериментальных исследований глубины проникновения в кремний квантов соответствующего спектрального диапазона. 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.

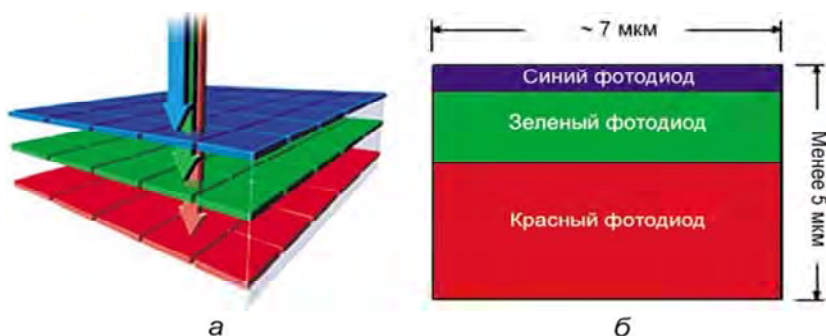


Рис. 5.5. «Трехслойная» фотоматрица: *а* – принцип действия; *б* – структура фотоматрицы

Достоинствами трехслойных фотоматриц является то, что они не требуют установки цветных фильтров и в связи с этим повышаются отношения «сигнал/шум», так как фильтры поглощают примерно 2/3 светового сигнала, а также потенциально более высокое разрешение.

Недостатками трехслойных фотоматриц следует считать невысокую точность цветопередачи, так как в наибольшей степени она определяется свойствами кремния, а также относительно высокий уровень цифрового шума из-за того, что часть фотонов поглощается не в «своей» области.

Отметим, что пока цифровые фотоаппараты не превосходят пленочные. Цифровой фотоаппарат не может 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.

5.3. Получение изображений с помощью ультразвука

Ультразвуковое исследование (УЗИ) – это неинвазивное исследование живого организма с помощью ультразвуковых волн (рис. 5.6 и 5.7).

Физическая основа УЗИ – пьезоэлектрический эффект. При деформации монокристаллов некоторых химических соединений под воздействием ультразвуковых волн на поверхности этих кристаллов возникают противоположные по знаку электрические заряды – 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. Эта часть в ультразвуковых аппаратах называется акустическим преобразователем.

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 существенные изменения: одна его часть продолжает распространяться в новой среде, в той или иной степени поглощаясь ею, другая отражается. Коэффициент отражения зависит от разности величин акустического сопротивления граничащих друг с другом тканей: чем это различие больше, тем больше отражение и амплитуда зарегистрированного сигнала, а значит, он светлее и ярче будет выглядеть на экране аппарата. Полным отражателем является граница между тканями и воздухом.

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 ор-

ганов брюшной полости и забрюшинного пространства, а также полости малого таза используется частота 2,5–3,5 МГц, для исследования щитовидной железы используется частота 7,5 МГц.

Генератор работает в импульсном режиме, посылая около 1000 импульсов в секунду. В промежутках между генерированием ультразвуковых волн пьезодатчик фиксирует отраженные сигналы. Отраженные эхосигналы поступают в усилитель и специальные системы реконструкции, после чего появляются на экране монитора в виде изображения срезов тела, имеющих различные оттенки черно-белого цвета, т. е. 2D-полутоновое изображение (см. рис. 5.6). Оптимальным является наличие не менее 64 оттенков серого.

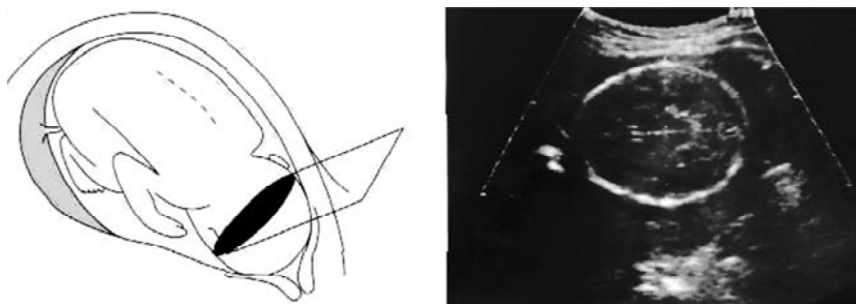


Рис. 5.6. Измерение размера головки плода:
а – схема; *б* – изображение



Рис. 5.7. Ребенок на 20-й неделе беременности

Сейчас можно получить 3D-УЗИ – статичные трехмерные изображения, получаемые в результате объемного сканирования

(см. рис. 5.7). УЗИ 4D – это движущиеся изображения в формате 3D, которые можно наблюдать в режиме реального времени. В качестве четвертого измерения выступает время.

5.4. Регистрация рентгеновского изображения

Получение изображения основано на ослаблении рентгеновского излучения при его прохождении через различные ткани с последующей его регистрацией на рентгеночувствительную пленку. При рентгенографии пациент размещается между рентгеновской трубкой и пленкой (детектором). В результате прохождения через образования разной плотности и состава пучок излучения рассеивается и тормозится, в связи с чем на пленке формируется изображение разной степени интенсивности (рис. 5.8). В результате на пленке получается усредненное изображение всех тканей (тень). Из этого следует, что для получения адекватного рентгеновского снимка необходимо проводить исследование рентгенологически неоднородных образований. В современных цифровых аппаратах регистрация выходного излучения может производиться на специальную электронную матрицу. Аппараты, обладающие электронной чувствительной матрицей, дороже аналоговых устройств. При этом изображение выводится на монитор и сохраняется в базе данных.

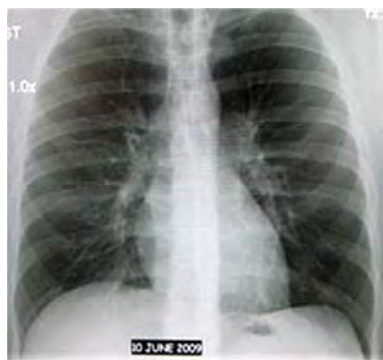


Рис. 5.8. Пример рентгеновского цифрового снимка

Качество полученного рентгеновского снимка определяется тремя основными параметрами: напряжением, подаваемым на рентге-

новскую трубку, силой тока и временем работы трубки. В зависимости от исследуемых анатомических образований и массогабаритных данных больного эти параметры могут существенно изменяться. Существуют средние значения для разных органов и тканей, но фактические значения отличаются в зависимости от аппарата, где проводится исследование, и больного, которому проводится рентгенография. Для каждого аппарата составляется индивидуальная таблица значений. Значения эти не абсолютные и корректируются по мере выполнения исследования.

Обычно разрешающая способность достигает 0,5 мм (одна пара линий на миллиметр соответствует 2 пикселям/мм), но можно получить и более высокое разрешение: до 26 пар линий на миллиметр, что соответствует 0,02 мм.

Недостатки рентгенографии:

информативность классической рентгенографии значительно ниже таких методов визуализации, как компьютерная (КТ) и магнитно-резонансная томография (МРТ). Обычные рентгеновские изображения отражают проекционное наложение 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.

5.5. Томографические изображения

Томографические изображения получают с помощью устройств двух типов на базе рентгеновского излучения и ядерного магнитного резонанса.

5.5.1. Магнитно-резонансная томография

Магнитно-резонансная томография (МРТ) – это томографический метод исследования внутренних органов и тканей с использованием физического явления ядерного магнитного резонанса. Метод основан на измерении электромагнитного отклика ядер атомов водорода на возбуждение их определенной комбинацией электромагнитных волн в постоянном магнитном поле высокой напряженности (рис. 5.9).



Рис. 5.9. МРТ-изображение головы человека

В 1973 г. Пол Лотербур опубликовал статью «Создание изображения с помощью индуцированного локального взаимодействия. Примеры на основе магнитного резонанса», и этот год считается годом основания магнитно-резонансной томографии. Позже Питер Мэнсфилд усовершенствовал математические алгоритмы получения изображения.

В СССР способ и устройство для ЯМР-томографии этого типа предложил В.А. Иванов в 1960 г. (в 1986 году термин «ЯМР-томография» был заменен на МРТ).

За изобретение метода МРТ Питер Мэнсфилд и Пол Лотербур в 2003 году получили Нобелевскую премию в области медицины.

В создание магнитно-резонансной томографии известный вклад внес также американо-армянский ученый Реймонд Дамадьян, один из первых исследователей принципов МРТ, держатель патента на МРТ и создатель первого коммерческого МРТ-сканера.

Томография позволяет с высоким качеством визуализировать головной, спинной мозг и другие внутренние органы.

Эффект ядерно-магнитного резонанса 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 сопро-

вождается выделением энергии в виде импульсов определенной (так называемой резонансной) частоты. Эту энергию можно измерить при помощи приемной катушки. Для получения изображения эти сигналы обрабатываются с помощью преобразования Фурье.

МРТ позволяет получать изображение в сагиттальной, вертикальной, горизонтальной плоскостях, а также под углом к ним, не изменяя положения больного. Изображение в каждой плоскости формируется за 5–10 мин. В отличие от КТ изменение положения больного во время исследования искажает все изображения.

5.5.2. Компьютерная томография

Идея компьютерной томографии (КТ), заключается в компьютерной реконструкции изображения поперечного среза тела на основании анализа поглощения рентгеновского излучения.

Трубка испускает тонкий веерообразный пучок рентгеновских лучей, перпендикулярный 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 фотоны 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. Таким образом, они чувствительнее и при определении различий в ослаблении.

Получение компьютерной томограммы (среза) на выбранном уровне основывается на выполнении следующих операций:

- 1) формирование требуемой ширины рентгеновского луча (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 отображения его на экране монитора.

Воздух или газ, вне зависимости от их места нахождения, выглядят черными, а кости – белыми. Другие органы, ткани и участки патологических изменений отображаются оттенками серого между этими двумя крайностями.

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, а применение большей первичной матрицы – к увеличению времени обработки среза или необходимости устанавливать дополнительные специальные процессоры видеоизображения.

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.

На МРТ лучше видны мягкие ткани, и на ранних стадиях поражения центрально нервной системы (ЦНС) она чувствительнее КТ. Компьютерная томография используется для визуализации позвоночника, основания черепа и височных костей. Острые внутримозговые и субарахноидальные кровоизлияния, а также стеноз позвоночного канала, остеохондроз и остеоартроз позвоночника на КТ лучше видны, чем на МРТ, хотя последняя предпочтительнее у больных с неврологическими нарушениями.

5.5.3. Отличия изображения компьютерной томографии от магнитно-резонансной

На обычном рентгеновском снимке лучше всего различимы кости. Поэтому КТ идеально подходит для диагностирования костных повреждений. Кроме того, на КТ хорошо видно свежее кровотечение, поэтому КТ 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 МРТ кости тоже видны, но еще лучше получаются мягкие ткани (хрящи, мышцы, связки, мозг и т. д.). Поэтому МРТ, например, показана при подозрении на разрыв связок и для исключения грыжи межпозвонкового диска. МРТ широко используют в нейрохирургии и неврологии (старые травмы головного мозга, инсульты в поздней стадии, подозрения на опухоли спинного и головного мозга и т. д.).

5.6. Формирование изображений дистанционного зондирования Земли

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

фотографические системы;

сканирующие системы видимого и ИК-диапазона (телевизионные, сканирующие радиометры и многоспектральные сканеры);

телевизионные оптические системы;

радиолокационные системы бокового обзора;

сканирующие СВЧ-радиометры.

Атмосфера поглощает часть отраженных лучей электромагнитного спектра, но в ней есть несколько «окон прозрачности», которые пропускают электромагнитные волны с минимальной степенью искажений. Чаще всего в методах ДЗЗ используется окно прозрачности, соответствующее оптическому (видимому) диапазону, объединяющему видимую (380–720 нм), ближнюю инфракрасную (720–1300 нм) и среднюю инфракрасную (1300–3000 нм) области. Полученные снимки классифицируются по пространственному разрешению:

снимки низкого разрешения (300–1 000 м);

снимки среднего разрешения (50–200 м);

снимки относительно высокого разрешения (20–40 м);

снимки высокого разрешения (10–20 м);

снимки очень высокого разрешения (1–10 м);

снимки сверхвысокого разрешения (0,3–0,9 м).

5.6.1. Формирование мультиспектральных изображений

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 полутонового изображения. Число фильтров определяет количество каналов мультиспектрального изображения (табл. 5.1).

Таблица 5.1

Спутники со съемочными системами высокого разрешения

Спутник	Pan, мм	B, мм	G, мм	R, мм	NIR, мм	Др.	Разрешение, м	Бит
WorldView-3	.45-.80	.4-.5 фиол и .45-51	.51-.58 и .585-.625 жел.	.63-.69 и .705-.745	.77-.895 и .86-1.04	Нет	.31 и 1.24	11
WorldView-2	-/-	-/-	-/-	-/-	-/-	Нет	.5 и 2	11
GeoEye-2	.45-.90	.45-.52	.52-.60	.625-.695	.76-.90	Нет	.25 и 1.65	11
Ikonos	.45-.90	.45-.52	.51-.60	.63-.70	.76-.90	Нет	1 и 4	11
QuickBird	.45-.90	.45-.52	.52-.60	.63-.69	.76-.90	Нет	.61 и 2.44	11
OrbView-3	.45-.90	.45-.52	.52-.60	.625-.695	.76-.90	Нет	1 и 4	11
Spot-6 и 7	.45-.745	.45-.52	.53-.59	.625-.695	.76-.89	Нет	2 и 6	12
СИЧ-2, Украина	.50-.90	Нет	.50-.60	.61-.68	.80-.89	1,55-.170	7,8 и 39,5	8
БелКА и Канопус-В	.45-.85	.46-.52	.52-.60	.63-.69	.75-.86	Нет	2,1 и 10,5	10

Отметим следующие факты:

- 1) чем шире спектральный диапазон отражаемого сигнала, тем выше разрешение (пример: панхроматический режим);
- 2) мультиспектральные диапазоны могут не пересекаться и наоборот.

5.6.2. Формирование гиперспектральных изображений

Ключевым моментом при построении гиперспектральных систем является способ формирования трехмерной структуры данных на двух- или одномерный массив фотоприемников. Гиперкубом называют трехмерный массив данных, два измерения которого соответствуют пространственным координатам x , y , третья – спектральной координате λ (рис. 5.10).

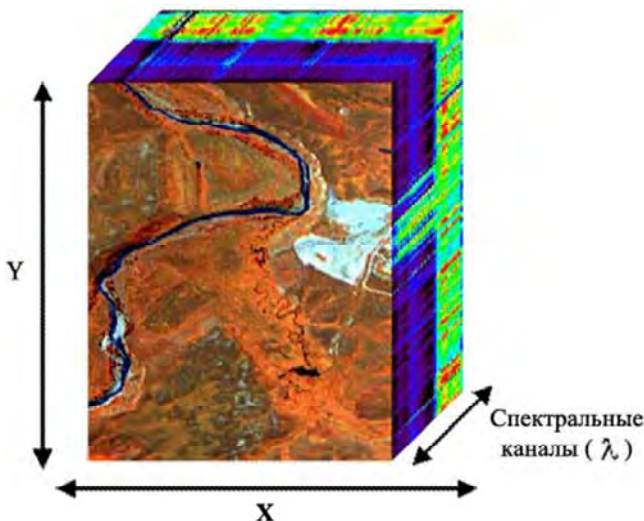


Рис. 5.10. Сканирование гиперспектрометром пространства вдоль оси x (или λ)

Гиперкуб получают при съемке кадра путем фиксации одной из координат: либо пространственной, например x , либо спектральной λ . За один кадр фотоприемная матрица формирует двумерный срез гиперкуба с фиксированной третьей координатой. Поэтому для построения полного гиперкуба приходится использовать сканирование по одной из координат – пространственной либо спектральной λ .

Наиболее распространенными на сегодняшний день являются гиперспектрометры, которые в каждый момент времени регистрируют узкий отрезок поверхности под собой. Такие гиперспектрометры относятся к типу pushbroom (рис. 5.11).

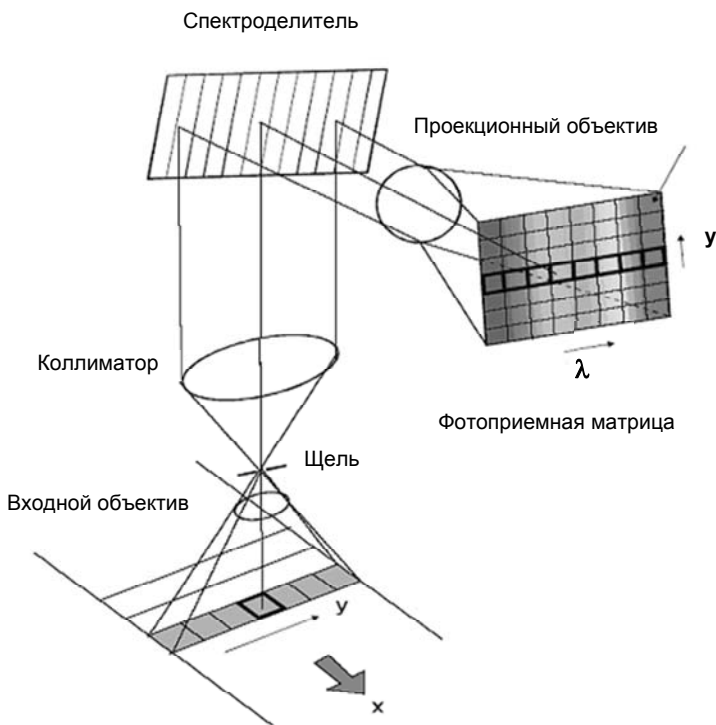


Рис. 5.11. Принципиальная схема гиперспектрометра

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

После коллимирующего объектива отраженный сигнал в параллельных лучах попадает на призму. **В призме происходит его разложение в спектр**, и затем он проецируется на фотоприемную матрицу. Таким образом, на матрице формируется срез гиперкуба для фиксированной пространственной координаты x .

Большинство спектрометров регистрируют 100–200 каналов шириной по 0,4–2,5 нм каждый.

Таким образом, мультиспектральные и гиперспектральные снимки могут покрывать одинаковый спектральный диапазон, но отличаются по оптическому способу регистрации.

Глава 6. ВИДЕО И ФОРМАТЫ ВИДЕО

Цифровое видео базируется на множестве технологий записи, обработки, передачи, хранения и воспроизведения визуального или аудиовизуального материала 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 одно- и трехматричные. Первые считывают изображение со светочувствительного элемента камеры (ПЗС или КМОП-матрица). Для получения цветного видеоизображения применяют специальную RGB-фильтрацию элементов изображения, чтобы на каждый пиксель приходилось по три значения цвета – красного, зеленого и синего. Такой метод применяется в недорогих одноматричных видеокамерах и фотоаппаратах с поддержкой видеозаписи.

Для профессиональной видеосъемки применяются трехматричные и более системы, где оптическое цветоделение производится при помощи дихроичной призмы. Каждое из цветоделенных изображений попадает на отдельную матрицу, с которой считываются 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 используется только 220 (диапазон 16–235), а остальные – для сигналов синхронизации. При 10-битном кодировании используется 877 уровней. Для цветовой компонент Cr, Cb используется 225 уровней в 8-битной системе и 897 уровней в 10-битной системе.

Для цифровых видеостандартов принято базовое соотношение $4 : 2 : 2$, которое означает, что цветоразностные компоненты Cr, Cb передаются с пространственным разрешением, в два раза меньшим разрешения по яркостному сигналу, потому что человеческий глаз более чувствителен к изменению яркости, чем цвета. При этом частота дискретизации для яркостного сигнала Y устанавливается рав-

ной 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 720×576 пикселей, для североамериканского стандарта NTSC – 720×480 , для видео высокой четкости (HD 720p) – 1280×720 , а для стандарта HDTV (Full HD) – 1920×1080 точек. Минимальный показатель, при котором движение будет восприниматься однородным, – 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 двумерного 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 развертке 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 720×576 пикселей и 16-битной глубиной цвета потребуется около полутора гигабайт свободного дискового пространства без учета звуковой дорожки. Существует несколько десятков форматов сжатия, которые используют разные алгоритмы компрессии. Для кодирования видеосигнала используют форматы видео MPEG-2, MPEG-4, H.264, DV, AVC и др. При этом, используя определенный алгоритм преобразования данных, можно сжать видео разными программными средствами, что на выходе дает различные результаты. Эти отличия определяются **кодеком** – программой, осуществляющей сжатие исходных сигналов. При этом каждый кодек использует собственный алгоритм, который влияет на качество и скорость кодирования.

Фильм длительностью 1,5 часа, записанный в формате DV, занимает 20 Гб, в MPEG-2 – 4,7 Гб, в MPEG-4 – 700 Мб.

Выводы по разделу I

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

Цифровое изображение – это дискретная функция двух пространственных переменных $f(x, y)$, определенная в конечной области плоскости XU , принимающая множество неотрицательных целых значений из диапазона $[0, 2n-1]$, где n – целое. То есть $f(x, y) \geq 0, 0 \leq x \leq a, 0 \leq y \leq i$, 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 называются **пикселями** (от англ. *pixel* – *picture element*).

На логическом уровне представления бинарные, полутоновые и индексированные изображения задаются двумерной таблицей, цветные и многоспектральные изображения – трехмерной матрицей. Для хранения и передачи изображение может 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 полутоновое, матрица двумерна. Если изображение цветное или многоспектральное, она трехмерна, но каждый ее слой, соответствующий отражению в определенном диапазоне электромагнитного спектра, представляет собой полутоновое изображение.

Такие матрицы, содержащие информацию о яркости, представляют цифровые изображения, которые являются исходными данными для методов обработки, описанных в разд. II.

Раздел II. ОБРАБОТКА ЦИФРОВЫХ ИЗОБРАЖЕНИЙ

Глава 7. КЛАССИФИКАЦИЯ МЕТОДОВ ЦИФРОВОЙ ОБРАБОТКИ ИЗОБРАЖЕНИЙ

Цифровая обработка изображений – это любой способ изменения данных, представленных цифровыми изображениями, результатом обработки также является цифровое изображение (рис. 7.1).



Темное или слабоконтрастное



Неправильные цвета



Шумное



Размытое



Неравномерно освещенное

Рис. 7.1. Примеры изображений, требующих корректировки

Цель обработки изображений:

преобразование изображений, ориентированное на визуальное восприятие человеком (в основном повышение качества, коррекция цвета, контраста, исправления мелких помех или искажение изображения),

преобразование изображений, ориентированное на их автоматизированный анализ (например, фильтрация, математическая морфология, сегментация на области определенных классов, выделение объектов и т. п.).

Среди задач обработки изображений можно выделить две: улучшение и восстановление изображений.

Главная цель **улучшения** заключается в таком изменении изображения, чтобы результат оказался наиболее подходящим с точки зрения конкретного применения. Слово *конкретное* является важным, поскольку, например, метод, будучи полезным для улучшения рентгеновских изображений, как правило, не подходит для обработки цветных фотографий, и наоборот.

Общей теории улучшения изображений не существует.

Если изображение обрабатывается для визуальной интерпретации, наблюдатель окончательно судит о том, насколько хорош примененный метод. Визуальное оценивание качества изображения – это субъективный процесс, а понятие «хорошее изображение» невозможно формализовать или оценить численно. Если цель – обработка изображения для машинной обработки, задача оценивания проще. Например, в задаче распознавания символов лучшим будет тот метод обработки изображения, который приводит к более точным результатам машинного распознавания.

Целью **восстановления** также является повышение качества изображения, но для других применений. Несмотря на пересечение областей использования 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, искаженного в процессе регистрации – например, смазанного, – путем использования априорной информации о явлении, которое вызвало ухудшение изображения. Поэтому методы восстановления основаны на моделировании процессов искажения, применения обратных процедур для воссоздания исходного изображения, и они могут быть оценены более объективно.

Таким образом, улучшение изображений выполняется как для визуального восприятия результата человеком, так и для автоматического анализа и распознавания изображений. В чем разница? Человеческий глаз не отличит на полутоновом изображении яркости 10, 14 и 18, а при автоматическом анализе придется решать, соответствует ли яркость этих пикселей одной однородной области.

Методы обработки изображений классифицируют по-разному:

1. Исправление дефектов изображения:

шум (случайные погрешности цвета в каждой точке изображения);
недостаточная или избыточная яркость;
недостаточная или избыточная контрастность, избыточный динамический диапазон изображения;
неправильный цветовой тон;
нерезкость (расфокусировка);
искажения из-за пыли, царапин на сканируемом документе;
«битые пиксели»;
устранение дисторсии (искривления);
ретушь фотопортретов:
ретушь кожи (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:

кадрирование;
создание панорам;
устранение ненужных деталей изображения;
фотомонтаж (создание из нескольких изображений нового изображения);
вставка чертежей, надписей, символов, указателей и пр.;
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 также атмосферную коррекцию изображений.

Отдельную нишу занимают методы обработки и анализа изображений с помощью математической морфологии.

Методы обработки можно условно разделить 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).

Методы обработки изображений можно разделить на три класса согласно количеству пикселей, участвующих в преобразовании яркости одного пикселя нового изображения:

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

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

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

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

Линейный функционал – это линейное отображение множества функций в множество чисел со свойствами:

$$\begin{aligned} F[f + g] &= F[f] + F[g]; \\ F[cf] &= cF[f], \end{aligned} \tag{7.1}$$

где F – линейный функционал;

f и g – функции из его области определения;

c – константа.

В соответствии с этим определением методы обработки изображений можно разделить на линейные и нелинейные. **Линейные** методы допускают упрощение вычислений (и соответственно ускорение), а также реализацию с помощью операции *свертки*.

Обработка изображений может осуществляться рекурсивными и нерекурсивными методами. В *рекурсивных* методах используется результат обработки части соседних пикселей, при *нерекурсивных* не используется. В обработке изображений большинство методов нерекурсивное.

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

Глава 8. ГЕОМЕТРИЧЕСКИЕ ПРЕОБРАЗОВАНИЯ ИЗОБРАЖЕНИЙ

Геометрические преобразования заключаются в пространственном преобразовании положения совокупности пикселей из одной двумерной системы координат в другую. При этом изображение удобно рассматривать в виде прямоугольной сетки раstra (пикселей), заданного в прямоугольной системе координат. При выполнении геометрических преобразований исходное множество пикселей с координатами (x, y) преобразуется в новое множество (т. е. «сетка» 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 имеются евклидово пространство с ортонормированной декартовой системой координат с взаимно ортогональными осями. Тогда каждому пикселю изображения ставится в соответствие пара (x, y) декартовых координат: их можно интерпретировать как двумерный вектор X , представленный отрезком, проведенным из точки $(0,0)$ до точки (x, y) .

Двумерные преобразования на плоскости можно интерпретировать как движение точек, соответствующих множеству пикселей. Выделим **линейные преобразования**, представляемые матрицами, т. е. преобразования, при которых новые координаты точки линейно зависят от исходных координат этой точки следующим образом:

$$\mathbf{x}' = \mathbf{T}\mathbf{x}. \quad (8.1)$$

Для удобства матричного описания и комбинирования геометрических преобразований в 2D-плоскости используется формализм, называемый однородными координатами. Точки плоскости описываются трехмерными векторами вида $(x, y, 1)$, а преобразование – в матричном виде:

$$\begin{pmatrix} \bar{x}' \\ \bar{y}' \\ w' \end{pmatrix} = \begin{pmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{pmatrix} \begin{pmatrix} \bar{x} \\ \bar{y} \\ w \end{pmatrix}. \quad (8.2)$$

К линейным преобразованиям можно отнести кусочно-линейные, т. е. такие, при которых изображение разбивается на части, к каждой из которых применяются разные линейные преобразования.

Нелинейные преобразования: полиномиальные (посредством полиномиальных функций), синусоидальные преобразования и т. п. в данном издании не рассматриваются.

8.1. Аффинные преобразования

Основные типы **аффинных преобразований** и их матричные представления показаны на рис. 8.1.

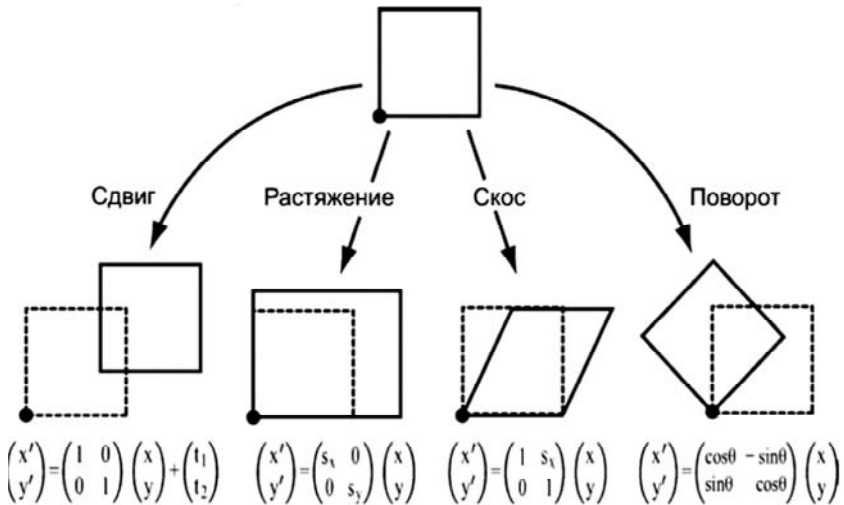


Рис. 8.1. Базовые преобразования аффинного типа

В общем виде любое аффинное преобразование на плоскости описывается уравнениями:

$$\begin{aligned}x' &= ax + bx + c; \\y' &= dx + ey + f,\end{aligned}\tag{8.3}$$

или в матричном виде

$$\begin{aligned}x' &= Ax + c; \\x' &= \begin{pmatrix} x' \\ y' \end{pmatrix}; \\A &= \begin{pmatrix} a & b \\ d & e \end{pmatrix}; \\c &= \begin{pmatrix} c \\ f \end{pmatrix}.\end{aligned}\tag{8.4}$$

В однородных координатах это преобразование можно описать одной матрицей T или

$$\begin{vmatrix} x' & y' & 1 \end{vmatrix} = \begin{vmatrix} x & y & 1 \end{vmatrix} \cdot \begin{vmatrix} t_{11} & t_{21} & 0 \\ t_{12} & t_{22} & 0 \\ x'_0 & y'_0 & 1 \end{vmatrix}.\tag{8.5}$$

Опишем параметры базовых аффинных преобразований (масштабирование, поворот, перенос, отражение относительно оси).

8.1.1. Сдвиг изображения

Сдвиг задается соотношениями

$$\begin{aligned}x' &= x + c; \\y' &= y + f.\end{aligned}\tag{8.6}$$

Матрица сдвига имеет вид

$$\mathbf{T}_{af} = \begin{bmatrix} 1 & 0 & c \\ 0 & 1 & f \\ 0 & 0 & 1 \end{bmatrix}.\tag{8.7}$$

8.1.2. Отражение изображения относительно оси

Отражение относительно оси абсцисс задается формулами:

$$\begin{aligned}x' &= x; \\ y' &= -y.\end{aligned}\tag{8.8}$$

Матрица отражения имеет вид

$$T_{af} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.\tag{8.9}$$

8.1.3. Масштабирование изображения

Преобразование задается в виде:

$$\begin{aligned}x' &= \alpha x, \quad \alpha > 0; \\ y' &= \beta y, \quad \beta > 0.\end{aligned}\tag{8.10}$$

В однородных координатах матрица преобразования имеет вид:

$$T_{af} = \begin{bmatrix} \alpha & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & 1 \end{bmatrix}.\tag{8.11}$$

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

8.1.4. Скол изображения

Скол вдоль оси OX описывается формулами

$$\begin{aligned}x' &= x + sy; \\ y' &= y.\end{aligned}\tag{8.12}$$

Матрица преобразования в однородных координатах имеет вид:

$$T_s = \begin{bmatrix} 1 & s & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.\tag{8.13}$$

8.1.5. Поворот изображения

Поворот вокруг начала координат на угол φ описывается формулами:

$$\begin{aligned}x' &= x \cos \varphi - y \sin \varphi; \\ y' &= x \sin \varphi + y \cos \varphi.\end{aligned}\tag{8.14}$$

Матрица вращения в однородных координатах имеет вид

$$T_{af} = \begin{bmatrix} \cos \varphi & -\sin \varphi & 0 \\ \sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}.\tag{8.15}$$

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

$$\begin{aligned}x' &= -y; \\ y' &= x.\end{aligned}\tag{8.16}$$

При повороте изображения координаты пикселя могут принять нецелые значения и их округляют до целого. Это создает в повернутом изображении пиксели с неопределенной яркостью. На рис. данные пиксели отмечены черными точками.

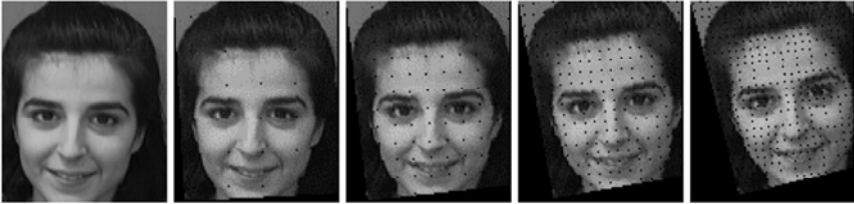


Рис. 8.2. Изображение, повернутое на 3, 6, 10, 14° вокруг верхнего левого угла. В черных пикселях яркость не определена

Пикселям с неопределенной яркостью можно присвоить, например, значение яркости ближайшего пикселя. Альтернативой является поворот посредством выполнения трех операций сдвига вдоль оси OX , затем OY и еще раз вдоль OX . Матрица вращения описывается в виде декомпозиции трех матриц:

$$R(\alpha) = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} = \begin{bmatrix} 1 & -\operatorname{tg} \alpha / 2 \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 \\ \sin \alpha & 1 \end{bmatrix} \times \begin{bmatrix} 1 & -\operatorname{tg} \alpha / 2 \\ 0 & 1 \end{bmatrix}. \quad (8.17)$$

После каждой операции сдвига выполняется округление координат до целых значений.

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

Любое аффинное преобразование имеет **обратное**, которое также является аффинным. Произведение прямого и обратного преобразований дает **единичное преобразование**, оставляющее все точки на месте.

Свойства аффинных преобразований:

сохраняются прямые линии;

сохраняются отношения длин отрезков, лежащих на одной прямой (или на параллельных прямых);

сохраняются отношения площадей фигур;

параллельные прямые переходят в параллельные.

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

$$\text{евклидово преобразование} \subset \text{аффинное} \subset \text{проективное преобразование}$$

8.2. Проективные преобразования

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

$$\begin{pmatrix} \bar{x}' \\ \bar{y}' \\ 1 \end{pmatrix} = \begin{pmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}. \quad (8.18)$$

При представлении в обычных координатах соотношение (8.18) будет иметь нелинейный вид:

$$x' = \frac{t_{11}x + t_{12}y + t_{13}}{t_{31}x + t_{32}y + t_{33}};$$
$$y' = \frac{t_{21}x + t_{22}y + t_{23}}{t_{31}x + t_{32}y + t_{33}}. \quad (8.19)$$

В общем случае проективные преобразования не сохраняют параллельности линий. Свойством, сохраняющимся при проективном преобразовании, является **коллинеарность** точек: три точки, лежащие на одной прямой (т. е. коллинеарные), после преобразования остаются лежать на одной прямой.

Проективное преобразование связано с отображением трехмерной визуальной информации на двумерную плоскость. С математической точки зрения мир удобно рассматривать как трехмерное проективное пространство P^3 , а плоскость изображения, включенную 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

$$p = (X, Y, Z)^t$$

в точку изображения

$$x = (x, y)^t$$

и выраженное в однородных координатах, задается в виде

$$\begin{pmatrix} \bar{x} \\ \bar{y} \\ w \end{pmatrix} = \begin{pmatrix} wx \\ wy \\ w \end{pmatrix} = P \begin{pmatrix} WX \\ WY \\ WZ \\ W \end{pmatrix} = \begin{pmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{pmatrix} \begin{pmatrix} \bar{X} \\ \bar{Y} \\ \bar{Z} \\ W \end{pmatrix}. \quad (8.20)$$

Изображения на рис.8.3, б и в получены следующими проективными преобразованиями:

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

и

(8.21)

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

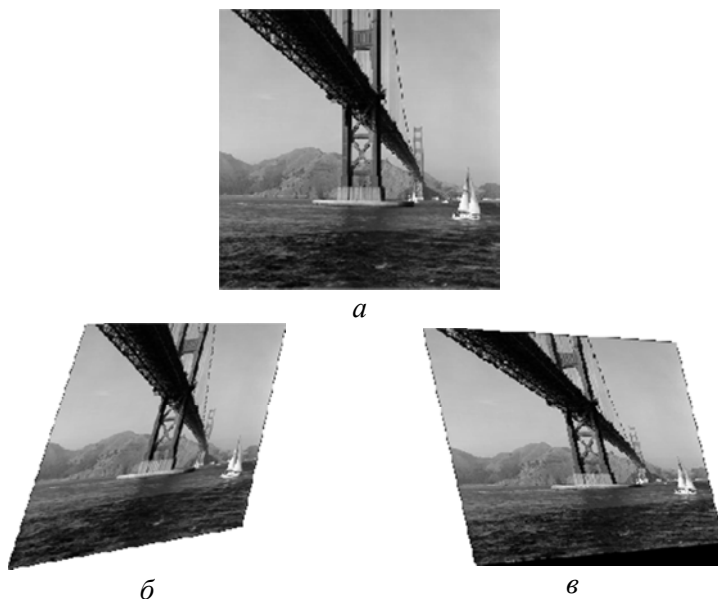


Рис. 8.3. Изображения: *a* – исходное изображение; *б* и *в* – два варианта его проективного преобразования

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

$$p = (\bar{X}/W, \bar{Y}/W, \bar{Z}/W)^t \quad \text{и} \quad x = (\bar{x}/w, \bar{y}/w)^t. \quad (8.22)$$

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 направления (луча проектирования) в плоскость снимка.

При рассмотрении геометрических преобразований использована идеальная модель камеры. В действительности формирование изображений сопровождается различными нелинейными искажениями.

ями (типа оптической дисторсии линзы). Приведение снимков к реальному представлению можно выполнить нелинейной функцией геометрического преобразования. Для этого используют полиномиальную аппроксимирующую функцию преобразования:

$$\begin{aligned} x' &= a_0 + a_1x + a_2y + a_3x^2 + a_4xy + a_5y^2 + \dots + a_p y^m; \\ y' &= b_0 + b_1x + b_2y + b_3x^2 + b_4xy + b_5y^2 + \dots + b_p y^m, \end{aligned} \quad (8.23)$$

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

Полиномиальная аппроксимирующая функция преобразования для снимка на рис. 8.4 принимает вид

$$\begin{aligned} x' &= 0,1x + 0,9y + 0,002xy, \\ y' &= 0,2x + 1,1y + 0,0022xy. \end{aligned} \quad (8.24)$$



Рис. 8.4. Билинейное геометрическое преобразование снимка с параметрами преобразования (8.2)

8.3. Уточнение координат и яркости пикселей после преобразования

После поворота или увеличения изображения часто возникает задача уточнения новых координат пикселей (на рис. 8.5 показаны серым) в целочисленной системе координат. Она решается путем округления координат пикселей до целых значений и двумерной интерполяции

значений яркости в неопределенных пикселях. Наиболее простой является интерполяция методом ближайшего соседа, где значения координат округляются до ближайших точек дискретного растра.

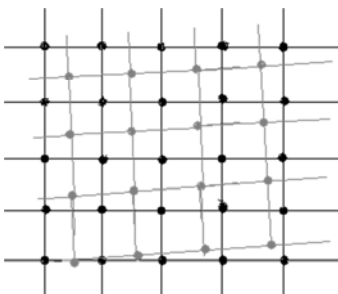


Рис. 8.5. Несовпадение координат пикселей (выделены серым фоном) дискретной целочисленной системой координат

Алгоритм ближайшего соседа присваивает неопределенному пикселю яркость ближайшего после геометрического преобразования с неокругленными координатами.

Методы увеличения изображений помимо «растяжения» пиксельной сетки требуют интерполяционного вычисления яркости дополнительных пикселей. Эти методы можно разделить на две группы: с произвольным и двукратным увеличением (метод zoom).

Отметим, что при уменьшении изображения его качество практически не меняется при использовании разных методов интерполяции, поэтому можно использовать простые методы.

8.4. Геометрическая коррекция изображений

8.4.1. Построение мозаики из изображений

Как можно использовать геометрические преобразования? Рассмотрим задачу построения мозаики из двух изображений. Мозаика или сшивка – это объединение двух или более изображений в единое. Пусть два изображения получены путем сканирования по частям одного большого рисунка (рис. 8.6). Обязательное условие: на обоих изображениях частично присутствуют одни и те же объекты, т. е. изображения имеют области перекрытия.



Рис. 8.6. Пример сканирования документа большого размера

Системы координат двух изображений будут различаться из-за разного ракурса съемки, вращения камеры и движения самого фотографируемого объекта. Для изображений, представленных на рис. 8.6, освещение, расстояние до камеры, оптика одинаковые, поэтому основной задачей обработки таких изображений является приведение изображений в общую систему координат. Если изображения регистрировались в разное время, необходимо учесть изменения яркости, возникающие из-за изменения освещения, сезонных и суточных изменений, использования другой камеры.

Рассмотрим задачу нахождения такого пространственного преобразования, которое позволяет преобразовать пиксели обоих изображений, представленных на рис. 8.7, в единую систему координат таким образом, чтобы точки, соответствующие одинаковым объектам на двух изображениях, совпали. Для упрощения в качестве общей системы координат можно использовать систему левого изображения, тогда требуется найти преобразование координат всех пикселей второго изображения (x, y) в общую систему координат (x', y') . Для упрощения задачи считаем, что в процессе регистрации не произошло искривление прямых линий, а лишь аффинные трансформации (поворот, сдвиг). Аффинные преобразования являются подмножеством полиномиальных преобразований первого порядка и описываются двумя уравнениями:

$$x' = a_0 + a_1x + a_2y; \tag{8.25}$$

$$y' = b_0 + b_1x + b_2y.$$

На обоих изображениях можно интерактивно выбрать или найти с помощью специальных методов пиксели, соответствующие одинаковым объектам (рис. 8.7).



Рис. 8.7. Соответствующие точки

Обозначим через (x_i, y_i) координаты таких точек на правом изображении в системе координат этого изображения, (x'_i, y'_i) – координаты в системе координат левого изображения. Эти координаты известны (точки заданы или вычислены), коэффициенты a_0 – a_2 , b_0 – b_2 неизвестны. Для решения задачи вычисления коэффициентов a_0 – a_2 , b_0 – b_2 аффинного преобразования минимальное количество общих точек $n = 3$. Если задать больше точек, то необходимо минимизировать ошибки их задания, которые неизбежны в силу неточного задания и округления координат до целых значений.

Итак, имеем 6 неизвестных, для их поиска требуются три пары точек 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) . Подставив координаты (x, y) в уравнение (8.25), получим три пары уравнений с неизвестными a_i и b_i :

$$\begin{aligned}
 x'_1 &= a_0 + a_1 x_1 + a_2 y_1, & y'_1 &= b_0 + b_1 x_1 + b_2 y_1, \\
 x'_2 &= a_0 + a_1 x_2 + a_2 y_2, & y'_2 &= b_0 + b_1 x_2 + b_2 y_2, \\
 x'_3 &= a_0 + a_1 x_3 + a_2 y_3, & y'_3 &= b_0 + b_1 x_3 + b_2 y_3.
 \end{aligned}
 \tag{8.26}$$

В матричной форме это выглядит следующим образом:

$$\begin{aligned}
 \begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \end{bmatrix} &= \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}; \\
 \begin{bmatrix} y'_1 \\ y'_2 \\ y'_3 \end{bmatrix} &= \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix}.
 \end{aligned}
 \tag{8.27}$$

Для вычисления коэффициентов a_i и b_i каждая часть матричного уравнения должна быть умножена на обратную матрицу с коэффициентами, например для a_i :

$$\begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}^{-1} \begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}^{-1} \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}.
 \tag{8.28}$$

В матричной форме коэффициенты вычисляются по формулам

$$\begin{aligned}
 \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} &= \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}^{-1} \begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \end{bmatrix}; \\
 \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix} &= \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}^{-1} \begin{bmatrix} y'_1 \\ y'_2 \\ y'_3 \end{bmatrix}.
 \end{aligned}
 \tag{8.29}$$

Полученный набор коэффициентов подставляется в исходные уравнения (8.25), и новые координаты вычисляются для каждого пиксела второго изображения. Осталось определить, какую часть второго изображения необходимо «пришить» к первому.

К полученному изображению можно «пришивать» справа и снизу другие изображения. Результирующее изображение представлено на рис. 8.8.



Рис. 8.8. Пример мозаики из двух изображений

При фотографировании больших объектов можно делать фотографии по частям, а затем выполнить шивку полученных изображений. Пример обработки панорамных фотографий приведен на рис. 8.9. Таким же образом была создана панорама г. Праги из 2600 снимков. В результате получено изображение размером $260\,000 \times 130\,000$ пикселей. Если его распечатать с разрешением 150DPI, получится фотография длиной почти 40 м.



Рис. 8.9. Пример сшивки из двух фотографий одного объекта

8.4.2. Коррекция проективных искажений

Проективное преобразование описывается уравнениями:

$$\begin{aligned}
 u_1 &= \frac{a_{01}x_1 + a_{10}x_2 + a_{00}}{c_{01}x_1 + c_{10}x_2 + 1}; \\
 u_2 &= \frac{b_{01}x_1 + b_{10}x_2 + b_{00}}{c_{01}x_1 + c_{10}x_2 + 1},
 \end{aligned}
 \tag{8.30}$$

где x_1 и x_2 – точки идеального изображения,

u_1 и u_2 – точки проективно искаженного изображения.

При коррекции перспективных искажений исходное изображение одно. На корректируемом изображении интерактивно задаются концы вертикальных и горизонтальных отрезков; посредством решения системы линейных уравнений вычисляются неизвестные параметры b_{ij} и c_{ij} . Исходное изображение трансформируется так, чтобы отмеченные отрезки стали вертикальными и горизонтальными.

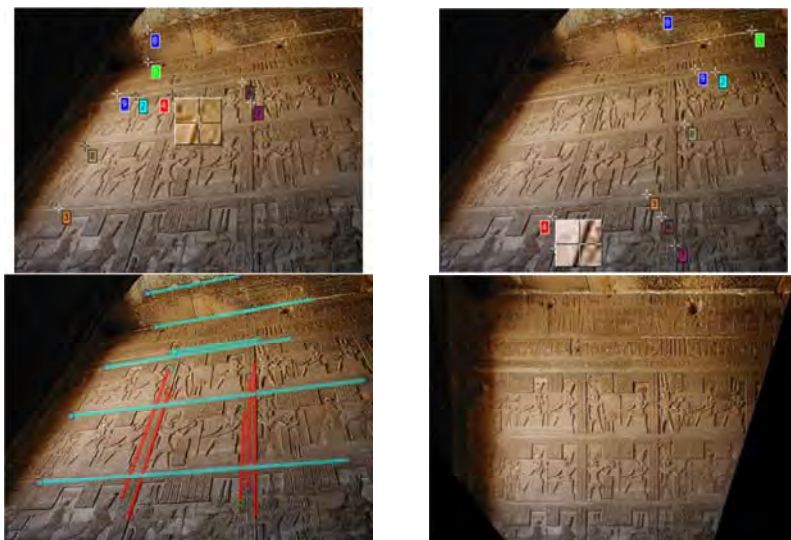


Рис. 8.10. Пример перспективной коррекции изображения

8.5. Нелинейная геометрическая коррекция

8.5.1. Общий подход к вычислению параметров преобразования

Не всегда задачу можно решить с помощью линейных преобразований. Нелинейные преобразования (второго и более порядка) помогают корректировать нелинейные искажения изображения, возникшие в процессе его регистрации (рис. 8.11).



Рис. 8.11. Примеры некоторых нелинейных преобразований

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

Способ решения задачи трансформации аналогичен предыдущему и сводится к нахождению коэффициентов системы уравнений. В случае полиномиального преобразования 2-й степени система уравнений выглядит следующим образом:

$$\begin{aligned} x' &= a_0 + a_1x + a_2y + a_3x^2 + a_4xy + a_5y^2; \\ y' &= b_0 + b_1x + b_2y + b_3x^2 + b_4xy + b_5y^2, \end{aligned} \quad (8.31)$$

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 минимально необходимо знать координаты 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 (8.31), перегруппируем и вычислим 12 неизвестных a_i и b_i :

$$\begin{aligned} x_1' &= a_0 + a_1x_1 + a_2y_1 + a_3x_1^2 + a_4x_1y_1 + a_5y_1^2; \\ &\dots \\ x_6' &= a_0 + a_1x_6 + a_2y_6 + a_3x_6^2 + a_4x_6y_6 + a_5y_6^2; \end{aligned} \quad (8.32)$$

$$\begin{aligned}
 y'_1 &= b_0 + b_1 x_1 + b_2 y_1 + b_3 x_1^2 + b_4 x_1 y_1 + b_5 y_1^2; \\
 &\dots \\
 y'_6 &= b_0 + b_1 x_6 + b_2 y_6 + b_3 x_6^2 + b_4 x_6 y_6 + b_5 y_6^2.
 \end{aligned}
 \tag{8.33}$$

Запишем эти уравнения для поиска a_i в матричной форме:

$$\begin{bmatrix} x'_1 \\ \dots \\ \dots \\ \dots \\ \dots \\ x'_6 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 & x_1^2 & x_1 y_1 & y_1^2 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & x_6 & y_6 & x_6^2 & x_6 y_6 & y_6^2 \end{bmatrix} \begin{bmatrix} a_1 \\ \dots \\ \dots \\ \dots \\ \dots \\ a_6 \end{bmatrix}.
 \tag{8.34}$$

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

$$\begin{bmatrix} a_1 \\ \dots \\ \dots \\ \dots \\ \dots \\ a_6 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 & x_1^2 & x_1 y_1 & y_1^2 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & x_6 & y_6 & x_6^2 & x_6 y_6 & y_6^2 \end{bmatrix}^{-1} \begin{bmatrix} x'_1 \\ \dots \\ \dots \\ \dots \\ \dots \\ x'_6 \end{bmatrix};
 \tag{8.35}$$

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

Полученный набор коэффициентов подставляется в исходные уравнения (8.31) для пересчета координат всех пикселей. Сначала нужно определить размер нового изображения, найдя минимум и максимум координат углов исходных изображений в единой системе координат. Как правило, пиксели левого изображения переносятся с теми же координатами, а правого – с учетом уравнений (8.31), т. е. новые координаты вычисляются для каждого пиксела второго изображения.

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

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

$$\frac{((t+1)(t+2))}{2}. \quad (8.36)$$

8.5.2. Коррекция дисторсии

При фотографировании на изображении может возникнуть оптическое искривление, которое называется дисторсия.

Дисторсия – это аберрация оптических систем, при которой линейное увеличение изменяется по полю зрения. Световые лучи проходят через центр линзы и сходятся в точке, расположенной дальше от линзы, чем лучи, которые проходят через ее края (рис. 8.12).

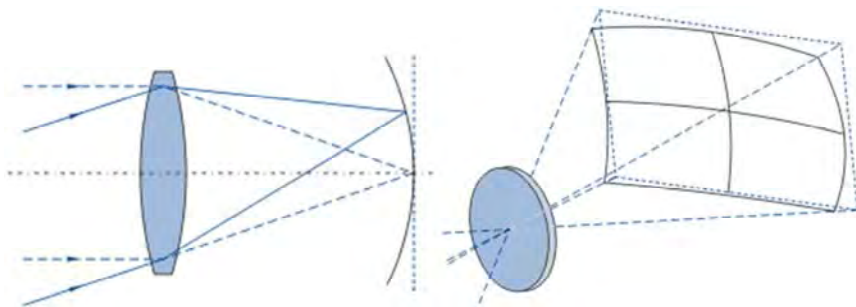


Рис. 8.12. Порождение дисторсии оптическими искажениями

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. 8.13). Это наиболее распространенные варианты дисторсии оптических систем фотоаппаратов с осевой симметрией.

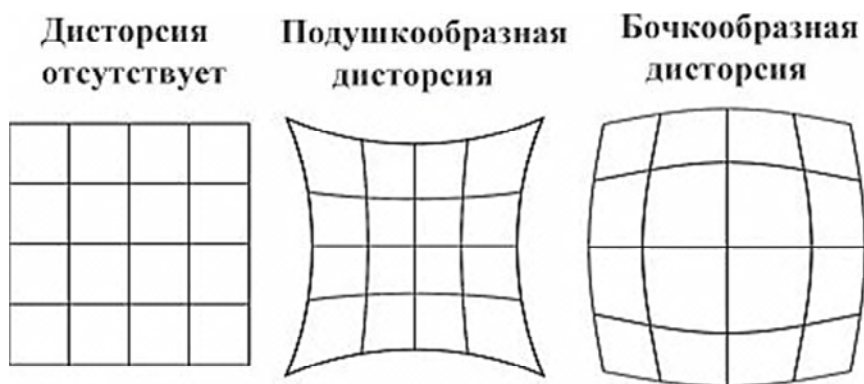


Рис. 8.13. Основные варианты дисторсии фотообъективов

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 программно можно исправить методом, описанным ниже.

Будем считать, что центр системы координат находится в центре изображения. Пусть $\vec{r} = (y, z)$ – вектор, задающий две координаты в плоскости, расположенной перпендикулярно оптической оси (координата x вдоль оси сюда не входит).

Для идеального изображения все лучи, вышедшие из точки с координатами \vec{r} и прошедшие через оптическую систему, попадут в точку изображения с координатами \vec{R} , которые определяются по формуле

$$\vec{R} = b_0 \vec{r}, \quad (8.37)$$

где b_0 – коэффициент линейного увеличения.

Если присутствует дисторсия третьего порядка (для осесимметричных оптических систем бывает только нечетных порядков: третьего, пятого, седьмого и т. д.), тогда лучи из \vec{r} попадут в

$$\vec{R} = b_0 \vec{r} + F_3 r^2 \vec{r}, \quad (8.38)$$

где r – длина \vec{r} ,

F_3 – дисторсия третьего порядка, которая обычно вносит наибольший вклад в искажение формы;

если F_3 имеет тот же знак, то и b_0 , возникает «подушка», в противном случае – «бочка».

Для вычисления параметров корректирующего преобразования используют, например, изображение регулярной сетки и ее искривленные изображения (рис. 8.14). На них выбирают пары соответствующих точек, по координатам которых вычисляются неизвестные коэффициенты b_0 и F_3 .

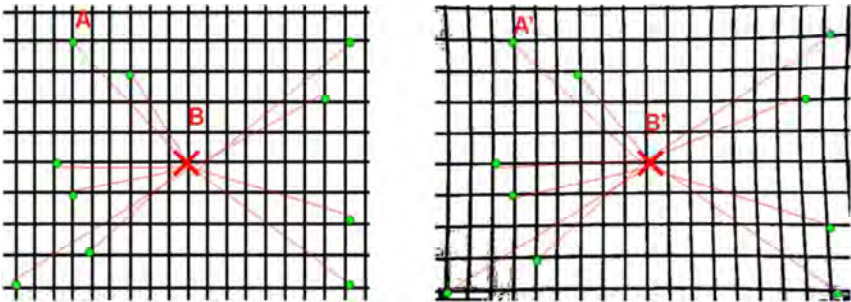


Рис. 8.14. Изображение сетки без искажений (слева) и ее фотография мобильным телефоном Nokia 6103 (справа) с соответствующими точками

Этим же преобразованием изменен рис. 8.15.



Рис. 8.15. Примеры двух видов дисторсии и исправленное изображение

Глава 9. ПРЕОБРАЗОВАНИЯ ИЗОБРАЖЕНИЙ ТОЧЕЧНОГО ТИПА

9.1. Логические операции над изображениями

Яркость бинарных изображений задается 0 и 1. Эти изображения можно интерпретировать как множество черных пикселей на белом фоне. К ним можно применять операции теории множеств AND, OR, NOT, XOR (рис. 9.1). Пусть L_1 и L_2 – значения яркости двух бинарных изображений равных размеров. Операции применяются к пикселям с одинаковыми координатами.

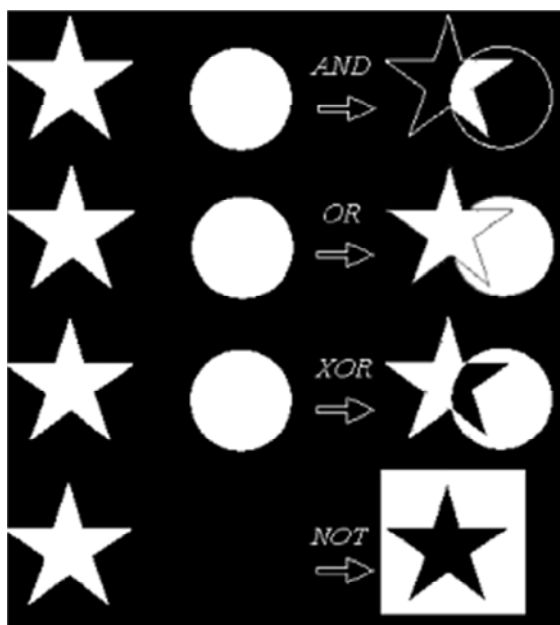


Рис. 9.1. Примеры логических операций над бинарными изображениями

Табл. 9.1 показывает, как изменяются исходные яркости при применении этих логических операций.

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

9.2. Линейные преобразования яркости полутоновых изображений

Яркостными преобразованиями изображения называются преобразования двумерных функций яркости, описываемые формулой

$$I'(i, j) = f(I(i, j))$$

Преобразования яркости изображения 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 исходного 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.

9.2.1. Препарирование изображений

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. 9.2, а) или оставить неизменными (рис. 9.2, б). Перемещая, выделенный интервал по яркостной шкале, и изменяя его ширину, можно детально исследовать содержание картины.

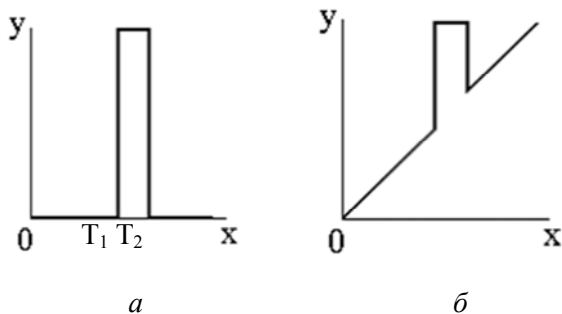


Рис. 9.2. Варианты графического задания функции яркостного среза

Преобразование выполняет яркостные срезы, выделяя на изображении области определенной яркости. При этом остальные участки яркости либо показаны черным (яркость 0, вариант *a*), либо линейно контрастируются. Перемещая выделенный интервал по шкале яркости и изменяя его ширину, можно детально исследовать содержание картины. Такие преобразования используются, если интересные интервалы яркости можно определить (вычислить или задать интерактивно).

Аналитически функция среза для (*a*) задается следующим образом:

$$y = \begin{cases} 255, & \text{если } T_1 < x < T_2; \\ 0, & \text{иначе.} \end{cases} \quad (9.1)$$

Пилообразное контрастирование – это линейное контрастирование, применяемое к отдельному диапазону (или диапазонам) яркости (рис. 9.3). Оно также позволяет повысить детальность изображения в выбранном интервале яркости.

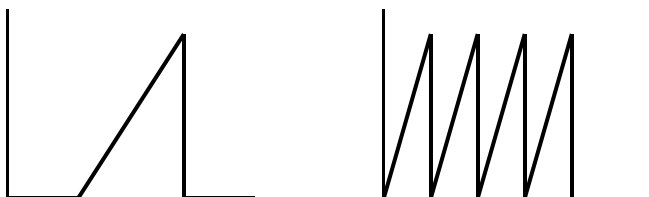


Рис. 9.3. Пилообразные варианты яркостного среза

9.2.2. Бинаризация изображений

Простейшим методом препарирования изображений является **бинаризация**. Она преобразует полутоновое изображение в бинарное (черно-белое). Преобразование имеет единственный параметр – порог, относительно которого яркость пикселей меняется на черную или белую. Функция бинаризации с глобальным (т. е. единым для всех пикселей) порогом представлена на рис. 9.4. Это преобразование используют при выделении темных объектов на светлом фоне или наоборот (рис. 9.5).

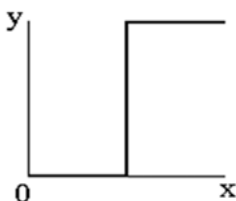


Рис. 9.4. Функция бинаризации с глобальным порогом



Рис. 9.5. Пример бинаризации изображения

Опишем три варианта вычисления порога.

Наиболее простой способ вычисления порога заключается в нахождении минимального g_{\min} и максимального значения g_{\max} яркости изображения и нахождении среднего значения между ними:

$$t = (g_{\max} - g_{\min}) / 2. \quad (9.2)$$

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

$$\sigma_{\omega}^2(t) = \omega_1(t)\sigma_1^2(t) + \omega_2(t)\sigma_2^2(t), \quad (9.3)$$

где веса ω_i – вероятности двух классов, разделенных порогом t ;
 σ_i^2 – дисперсия этих классов.

Минимизация дисперсии *внутри* класса равносильна максимизации дисперсии *между* классами, которая выражается вероятностью ω_i среднего арифметического i -го класса μ_i :

$$\sigma_b^2(t) = \sigma^2 - \sigma_{\omega}^2(t) = \omega_1(t)\omega_2(t)[\mu_1(t) - \mu_2(t)]^2, \quad (9.4)$$

где σ^2 – дисперсия всего изображения.

Алгоритм вычисления порога по методу Отсу

Шаг 1. Вычислить гистограмму яркостей и вероятность для каждого уровня интенсивности

$$p_i = n_i / N,$$

где N – число пикселей в изображении,

n_i – число пикселей 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 $\mu_i(0)$.

$$\omega_0(k) = \sum_{i=1}^k p_i; \quad \omega_1(k) = \sum_{i=k+1}^L p_i = 1 - \omega_0(k); \quad (9.5)$$

$$\mu_0(k) = \sum_{i=1}^k \frac{ip_i}{\omega_0}; \quad \mu_1(k) = \sum_{i=k+1}^L \frac{ip_i}{\omega_1}.$$

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

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

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

Алгоритм вычисления порога на основе градиента яркости изображения состоит из следующих шагов:

Шаг 1. Вычислить модуль градиента яркости для каждого пикселя изображения f :

$$G(m, n) = \max \{ |G_m(m, n)|, |G_n(m, n)| \}, \quad (9.6)$$

где

$$\begin{aligned} G_m(m, n) &= f(m+1, n) - f(m-1, n); \\ G_n(m, n) &= f(m, n+1) - f(m, n-1). \end{aligned} \quad (9.7)$$

Шаг 2. Вычислить значение порога t :

$$t = \frac{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) G(m, n)}{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} G(m, n)}. \quad (9.8)$$

9.2.3. Линейное растяжение гистограммы изображений

Задача линейного растяжения гистограммы связана с улучшением согласования динамического диапазона изображения и экрана, на котором выполняется визуализация. Если для цифрового представления каждого пикселя изображения отводится 1 байт (8 бит) запоминающего устройства, то входное или выходное изображения могут принимать одно из 256 значений. Чаще всего для отображения используется диапазон 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, изображение выглядит серым, малоконтрастным.

При линейном растяжении гистограммы изображений используется преобразование яркости типа

$$y = a \cdot x + b, \quad (9.9)$$

где a и b определяются желаемыми значениями минимальной и максимальной яркости результирующего изображения, обычно 0 и 255.

С учетом этого преобразование яркости принимает вид

$$y = \frac{x - x_{\min}}{x_{\max} - x_{\min}} (y_{\max} - y_{\min}) + y_{\min}. \quad (9.10)$$

Функция и пример **линейного растяжения гистограммы** изображения представлены на рис. 9.6 и 9.7 соответственно

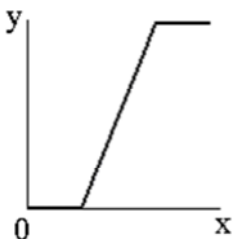


Рис. 9.6. Функция линейного контрастирования изображения



Рис. 9.7. Пример линейного растяжения гистограммы светлого изображения с диапазоном яркостей: $a - x_{\min} = 180$, $b - x_{\max} = 240$

Частным случаем линейного изменения гистограммы является преобразование изображения в **негативное представление**:

$$y = 255 - x. \quad (9.11)$$

Контрастирование можно выполнить **кусочно-линейно**. Пусть заданы $x_1 < x_2 < \dots < x_n$ – границы диапазонов изменения яркости (рис. 9.8).

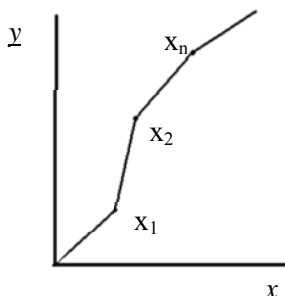


Рис. 9.8. Графическое представление функции кусочно-линейного контрастирования изображения

Тогда кусочно-линейная функция изменения яркости имеет вид

$$y = \begin{cases} a_0x + b_0, & x < x_1; \\ a_1x + b_1, & x_1 < x < x_2; \\ \dots & \\ a_nx + b_n, & x_n < x. \end{cases} \quad (9.12)$$

9.3. Арифметические операции над изображениями

Арифметические преобразования изображений относятся к точечному типу.

Умножение изображения на константу

$$y(i, j) = ax(i, j), \quad (9.13)$$

где $a > 0$ – константа.

Если $y(i, j) > 255$, заменить на $y = 255$.

Изменение яркости на константу c

$$z(i, j) = x(i, j) + c. \quad (9.14)$$

Сложение (вычитание) равных изображений (рис. 9.9):

$$z(i, j) = ax(i, j) \pm (1-a)y(i, j), \quad (9.15)$$

где x, y, z – цифровые изображения одного размера;

$0 < a < 1$ – константа.

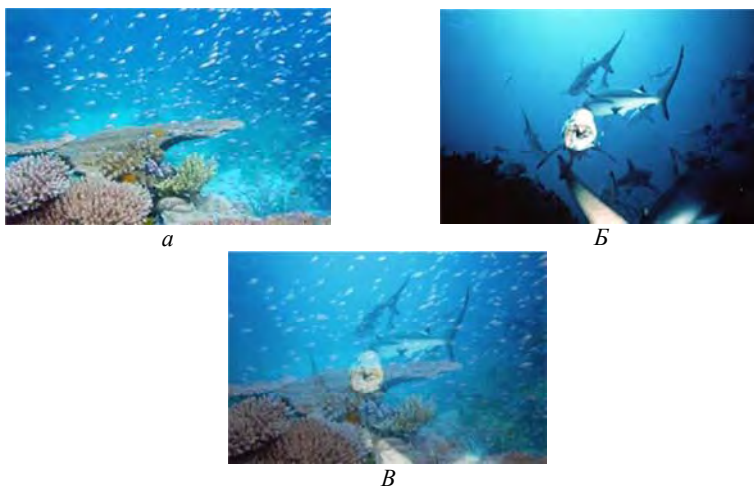


Рис. 9.9. Пример сложения изображений: a, \bar{b} – исходные изображения; \bar{b} – результат сложения двух исходных изображений

Пространственное маскирование изображения (рис. 9.10)

$$y(i, j) = a(i, j)x(i, j), \quad (9.16)$$

где $a(i, j)$ – матрица из нулей и единиц такого же размера, как изображение x .

Операция представляет собой поэлементное произведение двух матриц.



Рис. 9.10. Пример маскирования изображений

9.4. Нелинейная коррекция яркости изображений

9.4.1. Гамма-коррекция

Гамма-коррекция – это операция коррекции яркости изображения (I), используемая для демонстрации изображений на мониторах с нелинейной яркостной характеристикой. Обычно используется степенная функция в виде:

$$I = cI^\gamma. \quad (9.17)$$

Исторически введение гамма-коррекции было обусловлено тем, что у электронно-лучевой трубки зависимость между количеством испускаемых фотонов и напряжением на катоде близка к степенной формуле. В результате это вошло в стандарт и для жидкокристаллических мониторов, где зависимость между напряжением и яркостью имеет более сложный характер. Стандартное значение параметра «гамма» для видеоизображений NTSC равно 2,2. Для большинства мониторов значение гаммы составляет от 1,8 до 2,4.

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

$$I = 255(I/255)^\gamma. \quad (9.18)$$

Если изображение выглядит темным или блеклым, гамма-коррекция используется для исправления этих недостатков.

9.4.2. Логарифмическое преобразование яркости

Аналогично выполняется логарифмическое изменение яркости (c – задаваемая константа):

$$I = c \log(I + 1). \quad (9.19)$$

9.4.3. Соляризация изображений

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

$$y = kx(x_{\max} - x), \quad (9.20)$$

где k – константа, управляющая динамическим диапазоном преобразованного изображения;

x_{\max} – максимальное значение яркости исходного изображения (обычно 255).

Функция преобразования является квадратичной параболой, ее график при $k = 1$ приведен на рис. 9.11.

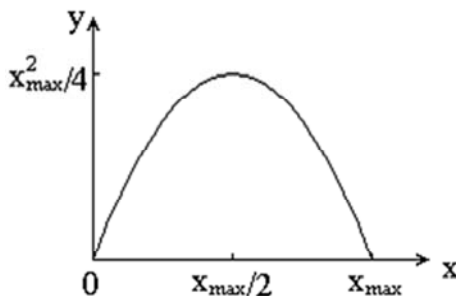


Рис. 9.11. Функции изменения яркости типа «соляризация»

Смысл преобразования состоит в том, что участки исходного изображения, имеющие яркость, близкую к белому цвету, после обработки принимают значения ближе к черному. При этом темные участки остаются темными. Значения, близкие к белому, приобретают участки, изначально имевшие средние уровни яркости (рис. 9.12).



Рис. 9.12. Пример применения соляризации изображения

На рис. 9.12, *а* показано исходное изображение и результат его соляризации, рис. 9.12, *б*. Преобразование повышает четкость темных и светлых областей: улучшено изображение глаз, повышен контраст на переходе «лицо-волосы».

9.4.4. Приведение среднего к заданному значению

Рассмотрим алгоритм коррекции яркости путем приведения среднего значения яркости к заданному значению (рис. 9.13).

Пусть L_{ij} – яркость пикселя с координатами ij .

Шаг 1. Задать требуемое значение средней яркости K .

Шаг 2. Вычислить минимальное L_{\min} , максимальное L_{\max} и среднееарифметическое значение яркости исходного изображения (A_{ν}):

$$A_{\nu} = \text{mean}(L) = \sum_{i,j} L_{ij}. \quad (9.21)$$

Шаг 3. Определить параметры преобразования значений яркости так, чтобы значение средней яркости стало равным K . Для этого, изменив значения яркости всех пикселей изображения

$$K = \sum_{i,j} F(L_{ij}), \quad (9.22)$$

например, нелинейно:

$$F(L_{ij}) = \left(\frac{L_{ij} - L_{\min}}{L_{\max} - L_{\min}} \right)^\alpha, \quad (9.23)$$

получим

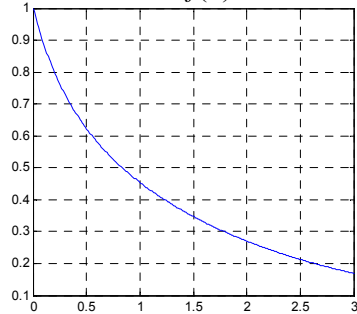
$$K = \sum_{i,j} F(L_{ij}) = \sum_{i,j} \left(\frac{L_{ij} - L_{\min}}{L_{\max} - L_{\min}} \right)^\alpha. \quad (9.24)$$

Исходное изображение



Средняя яркость 119

$K = f(\alpha)$



Средняя яркость 177



$K = 77$ при $\alpha = 1,85$

Рис. 9.13. Преобразование к заданному значению средней яркости

Подобрать значение α такое, чтобы среднеарифметическое значение яркости пикселей изображения стало равно заданному K .

Шаг 4. Выполнить преобразование яркости всех пикселей исходного изображения по формуле (9.23).

9.4.5. Табличный метод изменения яркости

В преобразованиях точечного типа яркость пикселя меняется по некоторой формуле для всех пикселей большого изображения. Пусть размер изображения равен $N \times N$, где $N \gg 256$. При нелинейных изменениях объем вычислений может оказаться значительным. То есть если в функции изменения яркости $f(x)$ используются умножение, деление, расчеты тригонометрических выражений, степенных и других функций, потребуется N^2 раз вычислить значения яркости пикселей.

От этих недостатков избавлен табличный метод замены яркости (LookUp Table или LUT). Его суть состоит в том, что предварительно, путем вычисления функции изменения яркости $f(x)$, создается таблица новых значений $y = f(x)$ (вида табл. 9.2). Затем яркость меняется в соответствии с этой таблицей, при этом x – исходная яркость, y – новое значение яркости.

Таблица 9.2

Табличное представление 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, ..., 198, 199, 201, 203, ..., 254, 255, 255, 255, ..., 12, 8, 4, 0.

Изменения яркости вычисляются по формуле $y = f(x)$ 256 раз, а если яркость вычислять для каждого пикселя, потребуется N^2 вычислений, где N может равняться нескольким тысячам. Поэтому табличное изменение яркости является очень быстрым.

9.4.6. Интерактивное изменение функции яркости изображения

Рассмотрим преобразование яркости, которое реализовано в пакете «Фотошоп» (функция «кривые»). Изначально график изменения яркости описывается тождественной функцией $y = x$, где x – яркость исходного, y – результирующего изображения (рис. 9.14). Значения x и y изменяются в диапазоне от 0 до 255.

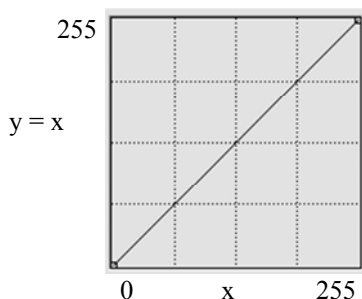


Рис. 9.14. Исходный график функции изменения яркости

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 $x_{11} \neq y_{11}$.

Кривая, задающая функцию изменения яркости, проходит через точки P_0, P_1, P_n (рис. 9.15), где $n = 2, P_0 = (0,0), P_1 = (x_{11}, y_{11}), P_n = (255, 255)$. При $n = 2$ это кривая второго порядка.

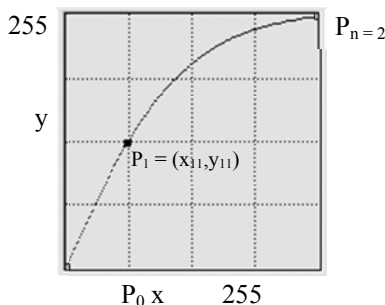


Рис. 9.15. Параболическая функция изменения яркости

Можно изменить форму графика, задав еще одну точку $P_2 = (x_{22}, y_{22})$ (рис. 9.16).

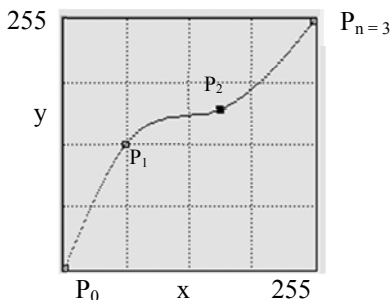


Рис. 9.16. Функция изменения яркости третьего порядка

Кривая должна проходить через конечные $P_0 = (0, 0)$, $P_n = (255, 255)$ и заданные точки P_1, P_2 . Как вычислить параметры функции, заданной этой кривой?

Кривая описывается полиномом степени, на 1 большей числа заданных (промежуточных) точек, в данном примере степени $n = 3$:

$$a_0x^n + a_1x^{n-1} + \dots + a_{n-1}x + a_n = y. \quad (9.25)$$

Из того что $P_0 = (0,0)$, уравнение (9.3) превращается в $a_n = 0$. Остальные a_i неизвестны, их количество равно n . Чтобы найти a_i , в уравнение (9.25) нужно подставить координаты точек $P_i = (x_i, y_i)$. Получим систему из n линейных уравнений с неизвестными a_i :

$$\begin{aligned} a_0x_1^n + a_1x_1^{n-1} + \dots + a_{n-1}x_1 &= y_1; \\ a_0x_2^n + a_1x_2^{n-1} + \dots + a_{n-1}x_2 &= y_2; \\ \dots & \\ a_0x_{n-1}^n + a_1x_{n-1}^{n-1} + \dots + a_{n-1}x_{n-1} &= y_{n-1}; \\ a_0255^n + a_1255^{n-1} + \dots + a_{n-1}255 &= 255. \end{aligned} \quad (9.26)$$

В матричном виде эта система записывается как

$$XA = Y, \quad (9.27)$$

где X – квадратная матрица констант;

$A = (a_1, a_2, \dots, a_{n-1})$ – вектор-столбец неизвестных;

$Y = (y_1, y_2, \dots, y_{n-1})$ – вектор-столбец констант:

$$X = \begin{bmatrix} x_1^n & x_1^{n-1} & \dots & x_1 \\ x_2^n & x_2^{n-1} & \dots & x_2 \\ & \dots & & \\ x_{n-1}^n & x_{n-1}^{n-1} & \dots & x_{n-1} \\ 255^n & 255^{n-1} & \dots & 255 \end{bmatrix}. \quad (9.28)$$

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} :

$$A = X^{-1}Y. \quad (9.29)$$

Вычислив коэффициенты a_i , подставляем их в уравнение (2.28) и строим таблицу LUT для всех $0 \leq x \leq 255$. Получим новые значения яркости y . Если $y < 0$, коэффициенты заменяем на 0. Если $y > 255$, коэффициенты заменяем на 255. LUT готова. Согласно полученной таблице заменяем все значения яркости в исходном изображении.

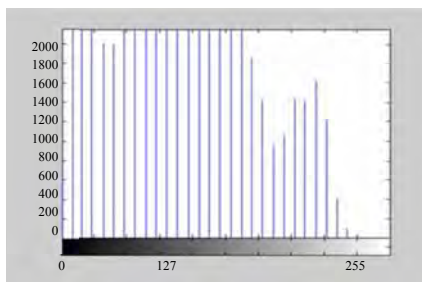
9.5. Преобразования яркости на базе гистограммы изображения

Гистограмма – это график 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 пикселей 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 пикселей распределена неравномерно (рис. 9.17). В результате часто на темных участках изображений детали оказываются плохо различимыми.



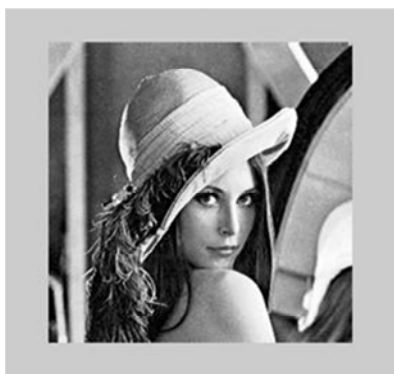
a



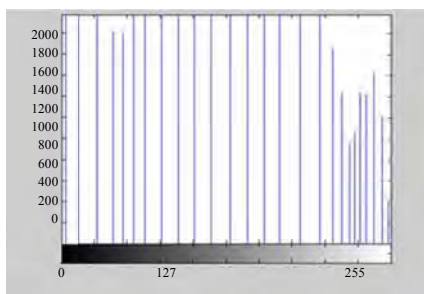
б

Рис. 9.17. Пример изображения (*a*) и его гистограммы (*б*)

Разработан ряд методов улучшения таких изображений на базе изменения гистограммы. Наиболее простым методом является **эквализация (выравнивание) гистограммы**. Это преобразование стремится изменить распределение яркостей изображения сделав график гистограммы максимально близким к горизонтальной линии (рис. 9.18).



a



б

Рис. 9.18. Пример выравнивания гистограммы изображения, представленного на рис. 9.17: *a* – преобразованное изображение; *б* – его гистограмма

Алгоритм эквализации (выравнивания) гистограммы состоит из следующих шагов.

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

Шаг 2. Массив H нормализовать так, чтобы сумма всех элементов равнялась 255.

Шаг 3. Вычислить кумулятивную гистограмму

$$\text{sum}(i) = \sum_0^i H(j). \quad (9.30)$$

Шаг 4. Вычислить новые значения яркости по формуле

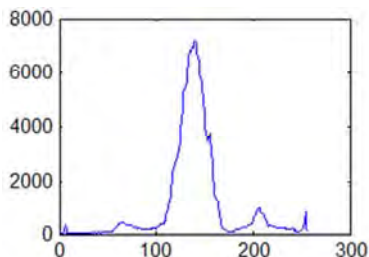
$$g(x, y) = \text{sum}(f(x, y)). \quad (9.31)$$

На рис. 9.18 показан результат выравнивания гистограммы изображения. Если изображение имеет много темных участков и малое число участков с высокой яркостью или наоборот, то выравнивание гистограммы корректирует яркость изображения, ровняя интегральные площади участков с различными яркостями.

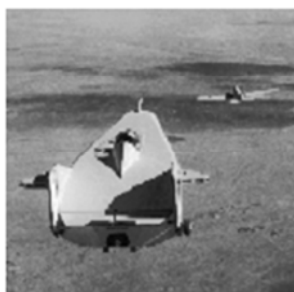
Преобразования гистограммы могут быть и нелинейными, например, приближенные к кривой любой формы (рис. 9.19), полученные по алгоритму, который будет описан ниже.



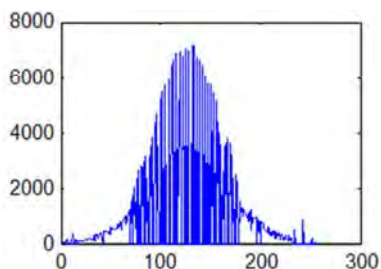
Исходное изображение



Гистограмма исходного изображения



Результирующее изображение



Гистограмма преобразованного изображения

Рис. 9.19. Приведение гистограммы к распределению яркостей по нормальному закону (по функции Гаусса)

Алгоритм нелинейного преобразования гистограммы

Шаг 1. Вычислить гистограмму h_x исходного изображения и его кумулятивную гистограмму H_x , после чего функция выравнивания этой гистограммы примет вид

$$H_x[j] = \sum_{i=0}^j h_x[i]. \quad (9.32)$$

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

$$H_z[j] = \sum_{i=0}^j h_z[i]. \quad (9.33)$$

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 $H_x[i] - H_z[j] > H_x[i] - H_z[j - 1]$, то уменьшить $j = j - 1$; иначе $LUT[i] = j$.

9.6. 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. применять глобальную обработку. Однако во многих случаях яркость, контраст, освещение, фон неоднородны. Поэтому глобальные методы обработки не дают желаемый результат. В таких случаях используются **адаптивные**, т. е. подстраиваемые под локальные характеристики 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 – плохая стыковка объектов на границах фрагментов, скачки яркости.

Чтобы снизить нежелательные эффекты 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.

Рассмотрим особенности адаптивных алгоритмов применительно к преобразованиям точечного типа.

9.6.1. Адаптивное повышение контраста

Операция аналогична линейному контрастированию, но коэффициенты меняются в зависимости от значений пикселей, попавших в скользящее окно:

$$g(n1, n2) = a(n1, n2) \cdot f(n1, n2) + \mu - \mu_f(n1, n2) \cdot a(n1, n2), \quad (9.34)$$

где

$$a(n1, n2) = \frac{\sigma}{\sigma_f(n1, n2)}; \quad (9.35)$$

$\mu_f(n1, n2)$, $\sigma_f(n1, n2)$ – локальное среднее и дисперсия,

μ и σ – глобальное среднее и дисперсия.

Если в окне изменение яркости мало, $\sigma_f(n1, n2)$ тоже мало, поэтому значения яркости больше растягиваются внутри окна. По этой же причине растягивается яркость в темных участках изображения (рис. 9.20).

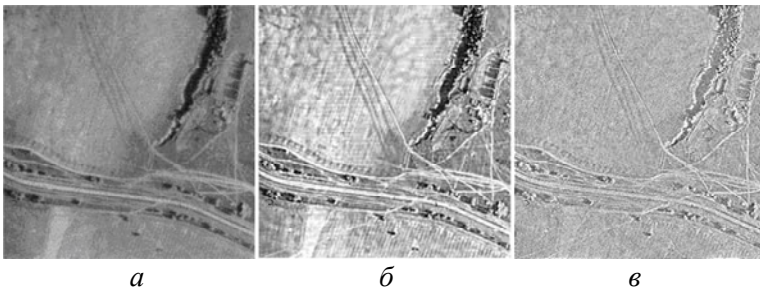


Рис. 9.20. Исходное изображение (а), локальное усиление контраста исходного изображения (б), глобальное усиление контраста исходного изображения (в)

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

$$L_{\text{out}}(i, j) = k \frac{\bar{L}}{\sigma(i, j)} [L_{\text{in}}(i, j) - \bar{L}_{\text{local}}(i, j)] + \bar{L}_{\text{local}}, \quad (9.36)$$

где \bar{L} – среднее значение яркости всего изображения L_{in} ;

$\sigma(i, j)$ – среднеквадратическое отклонение яркости в локальной окрестности изображения в пикселе с координатами (i, j) ;

$\bar{L}_{\text{local}}(i, j)$ – среднее значение яркости в окрестности с центром в пикселе (i, j) ;

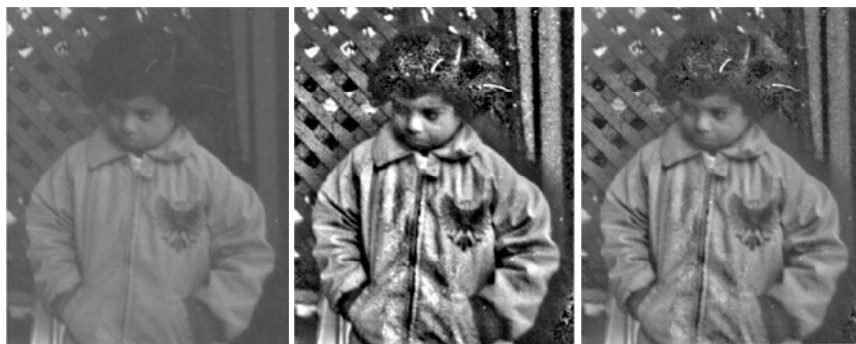
k – некоторая константа:

$$0 < k < 1.$$

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

На рис. 9.21 представлены результаты адаптивного повышения контраста по формуле (9.36). 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.



Исходное
изображение

Результирующее
изображение
при $k = 0,7$, $m = 35$

Результирующее
изображение
при $k = 0,3$, $m = 35$

Рис. 9.21. Результаты обработки исходного изображения, полученные с использованием формулы (9.4); m – размер локального окна

9.6.2. Адаптивное преобразование гистограммы

Аналогично адаптивному повышению контраста можно выполнить адаптивное преобразование гистограммы. Оно выполняется так же, как стандартное (глобальное) преобразование гистограммы, но применяется локально, т. е. к фрагментам изображения. Например, выравнивание гистограммы можно выполнить в скользящем окне (рис. 9.22).

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 пикселей, образующих левый столбец текущего фрагмента, и добавлением пикселей, образующих правый столбец соседнего фрагмента.

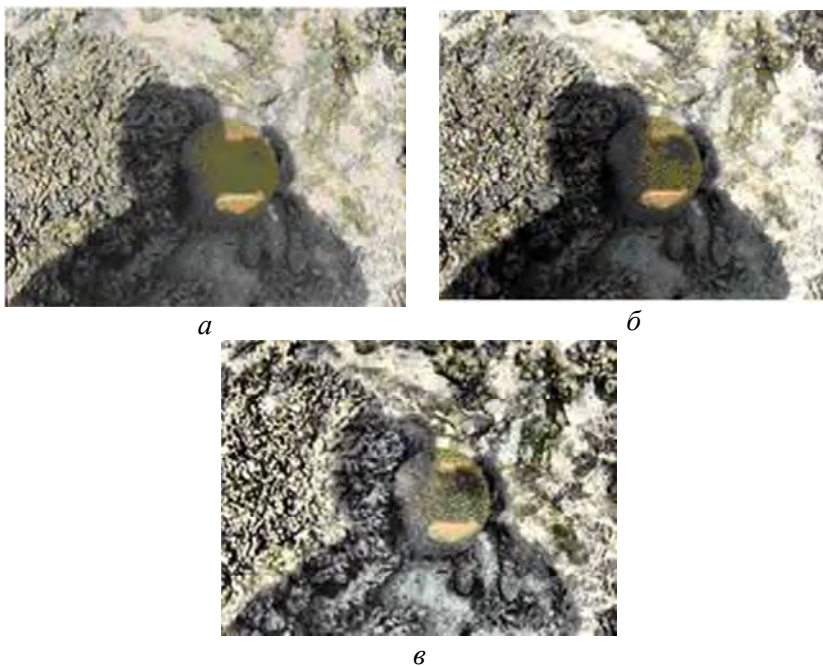


Рис. 9.22. Исходное изображение (*a*); после глобального выравнивания гистограммы (*б*); после адаптивного выравнивания гистограммы компоненты *L* в пространстве Lab фрагментами 8×8 (*в*)

Глава 10. ПРЕОБРАЗОВАНИЯ ЛОКАЛЬНОГО ТИПА

Цифровые изображения, сформированные различными оптико-электронными системами и зарегистрированные разными приемниками, искажаются действием помех различного характера. Искажения изображения вносятся всеми компонентами изображающего прибора, начиная с осветительной системы (например, неравномерность освещенности предмета). Искажения, которые вносит оптическая система, называются *абберации*. Искажения, которые вносят электронные приемники излучения, например ПЗС-матрицы, называются *электронным шумом*. Все помехи, вносимые в цифровое изображение, принято называть *шумом*. Шум затрудняет и визуальный анализ изображений человеком, и их автоматическую обработку.

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

10.1. 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.

10.1.1. Импульсный шум

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 точеч-

ным шумом. Выбросы шума особенно заметны на очень темных или очень светлых участках изображений.

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

$$x_{i,j} = \begin{cases} d & \text{с вероятностью } p; \\ s_{i,j} & \text{с вероятностью } (1 - p), \end{cases} \quad (10.1)$$

где $s_{i,j}$ – значения яркости исходного изображения.

Если новое значение $d = 0$, вносятся черные значения яркости, это шум типа «перец». Если $d = 255$, вносятся белые значения яркости, это 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.

10.1.2. Аддитивный шум

Индивидуальный шум описывается соотношением

$$g(x, y) = f(x, y) + \eta(x, y), \quad (10.2)$$

где $g(x, y)$ – зашумленное изображение;

$f(x, y)$ – исходное изображение;

$\eta(x, y)$ – аддитивный и не зависящий от сигнала шум с гауссовым (или другим) распределением функции плотности вероятности.

Гауссов шум (также называемый нормальный шум) возникает на изображении в результате возникновения 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 она предполагает независимость и нулевое математическое ожидание помех.

Функция плотности распределения вероятностей гауссовой случайной величины z (рис. 10.1) задается следующим выражением:

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\mu)^2/2\sigma^2}, \quad (10.3)$$

где z определяет уровень яркости из всех возможных значений для изображения (например, для восьмибитового изображения $0 \leq z \leq 255$);

μ – среднее значение (математическое ожидание) случайной величины z ;

σ – среднеквадратическое отклонение.

Величина, равная квадрату среднеквадратического отклонения, называется дисперсией σ^2 и определяет мощность вносимого шума.

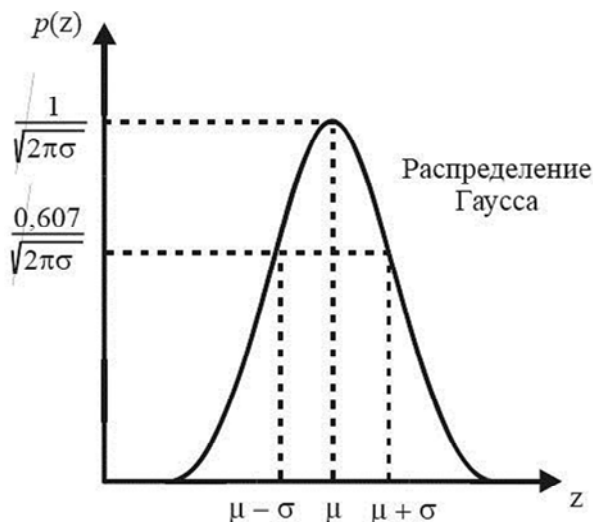


Рис. 10.1. Функция плотности распределения вероятностей гауссовой случайной величины z , имеющая форму колокола

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 $[(\mu - 2\sigma), (\mu + 2\sigma)]$.

10.1.3. Мультипликативный шум

Мультипликативный шум описывается соотношением

$$g(x, y) = f(x, y) \eta(x, y), \quad (10.4)$$

т. е. шумовая составляющая умножает принятый сигнал. Примерами такого шума являются:

- зернистость фотопленки;
- изображения, полученные с помощью ультразвука;
- радиолокационные изображения.

10.1.4. Шум квантования

Шум квантования зависит от сигнала и характеризуется выбранным шагом квантования значений сигнала. Например, в результате бинаризации яркость изображения описывается двумя значениями: черный и белый.

Такой шум может производить нежелательные артефакты (например, ложные контуры вокруг объектов) или устранять низкоконтрастные детали на изображении. Он не устраняется.

10.1.5. Спекл-шум

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 он обычно является мультипликативным. Спекл-шум возникает при получении цифровых изображений с помощью ультразвуковых медицинских сканеров, оптических сканеров и радаров. В результате изображение выглядит «зернистым».

10.2. Фильтрация изображений

Ослабление помех достигается фильтрацией изображения. Изображение описывается двумерной функцией, значения которой из-

меняются медленнее, чем двумерной функции, описывающей шум. Многообразие методов фильтрации изображений связано с разнообразием математических моделей сигналов, помех и критериев оптимальности фильтрации. Фильтрация изображений производится в пространственной или частотной областях. Для фильтрации в частотной области изображение должно быть преобразовано в частотное представление с помощью, например, преобразования Фурье, которое относится к преобразованиям глобального типа.

Методы фильтрации можно разделить на два класса: *линейные* и *нелинейные*.

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

При оценке полезного сигнала в каждом пикселе рассматривают окрестность этого пикселя и значения яркости в этой окрестности.

Преобразования, которые для вычисления значения яркости одного пикселя учитывают значения пикселей в некоторой окрестности называются *локальными*. Окрестность называют также окном. Форма и размеры окна могут быть произвольными, положение центра окна также может быть любым.

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

При линейной фильтрации изображений форма и размер используемой окрестности определяют маску фильтра. *Маска* – это матрица весовых коэффициентов, определенных для пикселей, попавших в окрестности анализируемого пикселя. На практике используют в основном маски симметричной формы, центр которой совпадает с центром окрестности, т. е. маски. Распространенным видом окрестности является квадрат размером 3×3 , 5×5 и т. п.

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

$$g(x, y) = \sum_s \sum_t w(s, t) f(x + s, y + t), \quad (10.5)$$

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. 10.2).

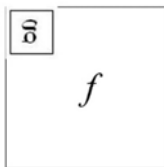


Рис. 10.2. Операция свертки изображения f с помощью ядра g

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

$$(f * g)[m, n] = \sum_{k, l} f[m - k, n - l]g[k, l]. \quad (10.6)$$

Отметим, что, если ядро несимметрично, его надо преобразовать симметрично относительно его же центра.

Таким образом, идея фильтрации основывается на интегральном использовании яркости пикселей, попавших в заданное окно, с учетом коэффициентов ядра фильтра (рис. 10.3). В этом состоит основное отличие фильтрации от рассмотренных выше операций обработки изображений точечного типа: фильтрация является процедурой обработки изображений локального типа.

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 несколько способов учесть это обстоятельство:

- 1) простейший способ состоит в ограничении перемещения центра окна по изображению, т. е. не выходить за края изображения;
- 2) расширить изображение путем добавления строк и столбцов с нулевыми значениями;
- 3) расширить изображение симметрично границе, повторив строки и столбцы изображения.

После обработки добавленные строки и столбцы удаляются. Это позволяет получить обработанное изображение размером с исходное. Следует помнить, что значения пикселей, использовавшихся

для расширения изображения, оказывают влияние на результат фильтрации пикселей внутри граничной полосы, ширина которой зависит от размеров маски.

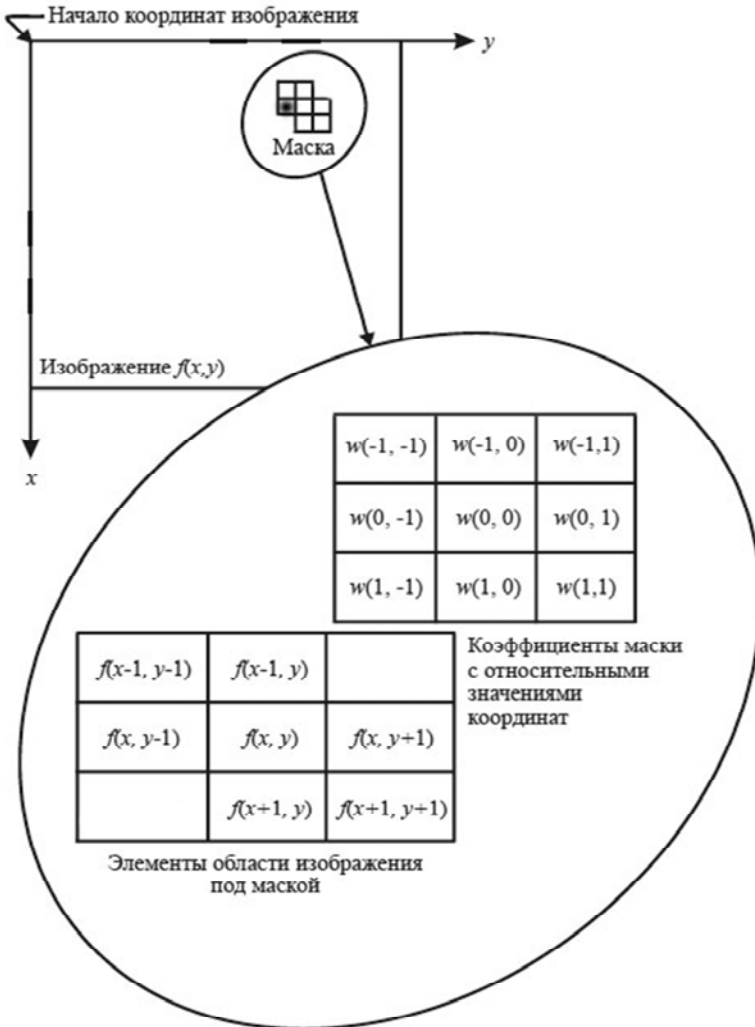


Рис. 10.3. Изображение, окно фильтра, наложенное на пиксель, показанный темным цветом, коэффициенты ядра фильтра (причем $w(1, -1) = w(-1, 1) = 0$), значения яркости пикселей, попавших в окно

10.3. Низкочастотные фильтры

Низкочастотные пространственные фильтры оставляют низкочастотные компоненты изображения (т. е. области с плавными изменениями яркости) нетронутыми и ослабляют высокочастотные компоненты. Такие фильтры используются для понижения шума, содержащегося в изображении, удаления высокочастотных компонент из изображения с тем, чтобы можно было тщательнее исследовать содержание низкочастотных компонент. Результат низкочастотной фильтрации – размытие изображения. На рис. 10.4 показаны примеры ядер низкочастотных фильтров.

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

Рис. 10.4. Примеры ядер низкочастотных фильтров

Отличительные признаки низкочастотных фильтров:

- а) неотрицательные коэффициенты ядра;
- б) сумма всех коэффициентов равна единице.

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

Большую группу низкочастотных фильтров составляют усредняющие фильтры. В усредняющих фильтрах различен способ нахождения среднего значения яркости в окрестности. Различают арифметические, геометрические, гармонические и контргармонические усредняющие фильтры.

10.3.1. Арифметический усредняющий фильтр

Арифметический усредняющий, или **box-фильтр**, усредняет значение яркости пикселя по окрестности путем использования маски с одинаковыми коэффициентами. Например, для маски размером 3×3 коэффициенты равны $1/9$ при 5×5 – $1/25$. Такая нормировка необходима для того, чтобы привести значение результата фильтрации к диапазону яркости исходного изображения. Графиче-

ское представление двумерной функции, описывающей маску этого фильтра, похоже на коробку или параллелепипед (рис. 10.5), отсюда и англоязычное название – box-фильтр:

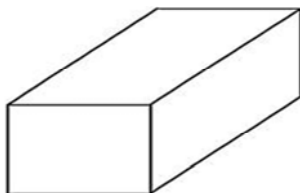


Рис. 10.5. Графическое представление функции усредняющего фильтра (box-фильтр) и его функция

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

В данной формуле

$f(x, y)$ – 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 $1/(MN)$. Из формулы следует, что арифметический метод усреднения – частный случай алгоритма линейной свертки и может применяться для устранения высокочастотных помех и сглаживания изображений. Алгоритм арифметического усреднения эффективен лишь для слабо зашумленных изображений.

10.3.2. Геометрический усредняющий фильтр

Геометрическое усреднение может быть записано в виде следующей формулы:

$$f(x, y) = \left[\prod_{i=0}^M \prod_{j=0}^N g(i, j) \right]^{\frac{1}{MN}}. \quad (10.7)$$

При таком усреднении происходит сглаживание изображения, аналогичное арифметическому усреднению, наблюдается ухудшение резкости, свойственное всем фильтрам из этого класса, но отдельные объекты исходного изображения искажаются меньше.

Этот фильтр так же, как усредняющий арифметический, может использоваться для подавления высокочастотного аддитивного шума, имея при этом лучшие статистические характеристики, но новые значения яркости дольше вычисляются.

10.3.3. Гармонический усредняющий фильтр

Фильтр на базе **гармонического усреднения** основывается на выражении

$$f(x, y) = \frac{MN}{\sum_{i=0}^M \sum_{j=0}^N \frac{1}{g(i, j)}}. \quad (10.8)$$

Этот фильтр хорошо работает с импульсными шумами типа «соль», т. е. белыми точками, и не работает с шумами типа «перец».

10.3.4. Контргармонический усредняющий фильтр

Контргармонический фильтр представляет собой обобщение арифметического и гармонического усреднения. Этот фильтр меняет яркость пикселя согласно функции

$$f(x, y) = \frac{\sum_{i=0}^M \sum_{j=0}^N g(i, j)^{Q+1}}{\sum_{i=0}^M \sum_{j=0}^N g(i, j)^Q}, \quad (10.9)$$

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 (т. е. 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.

10.3.5. Фильтр Гаусса

При задании фильтров можно использовать маски с разными весовыми коэффициентами (см. рис. 10.3). Логично предположить, что пиксели, расположенные ближе к анализируемому пикселю, оказывают большее влияние на яркость, вычисляемую в процессе фильтрации. К такому типу фильтров относятся фильтры, значения весов которых описывается колоколообразной **функцией Гаусса** (рис. 10.6):

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad (10.10)$$

где параметры μ – координата центральной точки, σ – вещественная константа, определяющая ширину колокола. Также на рис. 10.6 представлены двумерные варианты задания функции Гаусса.

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

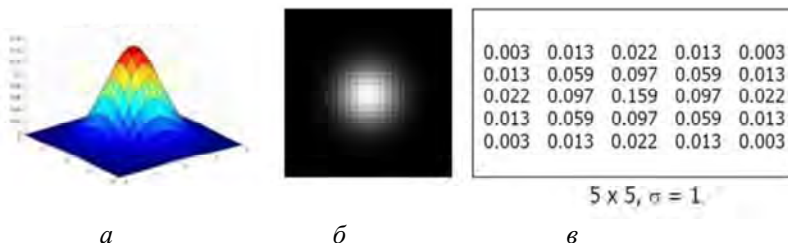


Рис. 10.6. Разные представления функции Гаусса: аналитическое, a – в виде поверхности; b – в виде полутонного изображения; c – в виде маски с коэффициентами

Чем больше σ , 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 3σ .

Свойства фильтра Гаусса

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

$$G_{\sigma}(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) = \left(\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right)\right) \cdot \left(\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{y^2}{2\sigma^2}\right)\right), \quad (10.11)$$

а это позволяет снизить количество вычислений с $4r^2$ до $4r$ на каждый пиксель.

2. Свертка два раза с ядром фильтром радиуса σ дает тот же результат, что с фильтром радиуса $\sigma\sqrt{2}$.

10.4. 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. Таким образом, после обработки высокочастотным фильтром изображение становится более резким.

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

Края на изображении можно разделить на два основных типа: в виде ската и в виде крыши (рис. 10.7).

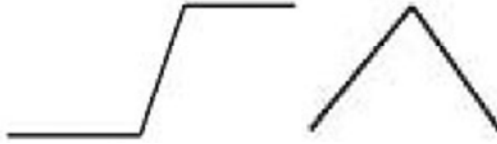


Рис. 10.7. Основные типы краев

Как правило, резкое изменение яркости можно определить, анализируя первую производную функции яркости (рис 10.8). Производная характеризует величину градиента. Граница объекта на изображении обычно перпендикулярна к вектору градиента.

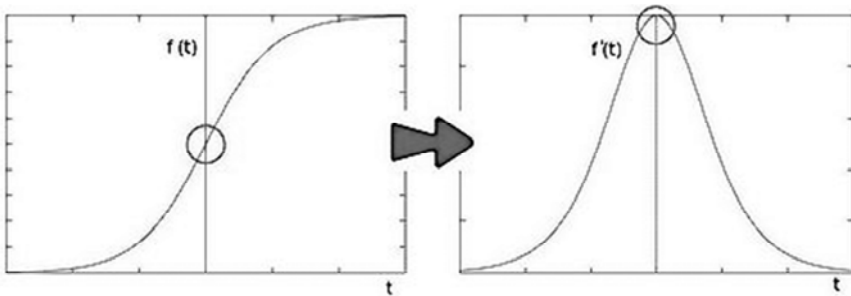


Рис. 10.8. Функция яркости и ее первая производная (максимум производной соответствует краю)

Для вычисления производной применяют высокочастотные фильтры, которые аппроксимируют вычисление производных по направлению. При этом Δx равно 1 или 2. В ядрах высокочастотных фильтров часть коэффициентов имеет отрицательные значения, но сумма всех коэффициентов равна нулю, что означает вычисление разности значений яркости соседних пикселей. Для ускорения вычислений все коэффициенты, как правило, целочисленные.

10.4.1. Фильтр Робертса

Метод Робертса является самым простым, быстрым и весьма эффективным. Он работает с маской 2×2 , представленной на рис. 10.9.

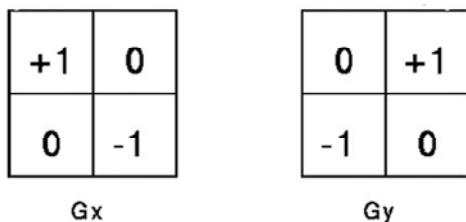


Рис. 10.9. Ядра фильтра Робертса для вычисления производных по X и по Y

В результате свертки изображения с этими ядрами получаем оценку градиента (т. е. перепада яркости) по направлениям G_x , G_y . Суммарное значение градиента всех детекторов края вычисляется по формуле

$$G = \sqrt{G_x^2 + G_y^2}$$

или упрощенно (10.12)

$$G = |G_x| + |G_y|.$$

10.4.2. Фильтр Превитта

Метод Превитта использует две ортогональные матрицы (ядра Превитта) размерностью 3×3 для приближенного вычисления производных по X и по Y :

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix};$$

(10.13)

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

10.4.3. Фильтр Собела

Фильтр Собела использует разные веса в ядрах. Классическим вариантом фильтра Собела являются следующие ядра:

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix};$$

(10.14)

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

10.4.4. Фильтр Лапласа

Ранее описанные фильтры использовали аппроксимацию первой производной по оси X и по оси Y . В фильтре Лапласа используется аппроксимация вторых производных по этим осям (рис. 10.10). 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:

$$L(f(x, y)) = \frac{d^2 f}{dx^2} + \frac{d^2 f}{dy^2}. \quad (10.15)$$

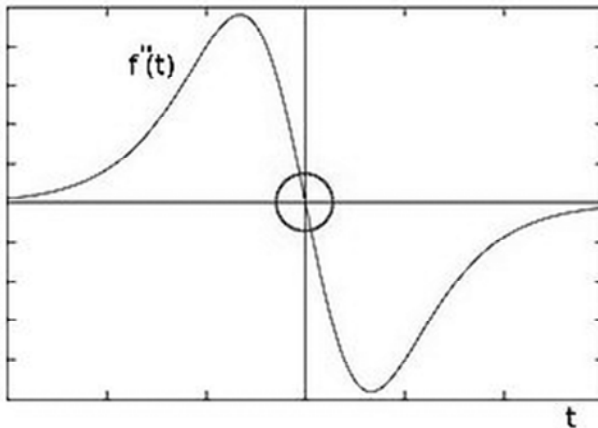


Рис. 10.10. Вторая производная функции яркости (меняет знак, т. е. проходит через ноль в точке соответствующей краю)

Для дискретных функций вторые производные могут быть аппроксимированы следующим образом:

$$L(f(x, y)) = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y). \quad (10.16)$$

Выражение (10.16) позволяет реализовать фильтр Лапласа посредством свертки некоторого ядра $w(s, t)$ (10.17) с изображением $f(x, y)$.

$$w(s, t) = \begin{vmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{vmatrix} \quad (10.17)$$

После свертки близкие к нулю значения соответствуют краевым пикселям. Ядро фильтра Лапласа может быть описано разными способами, например:

$$\begin{vmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{vmatrix}; \quad (10.18)$$

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

10.4.5. Выделение краев методом Кэнни

Цель алгоритма Кэнни – более точное выделение не краев, а границ на изображении (рис. 10.11). Алгоритм основан на использовании двух порогов при анализе матрицы градиентов. Пороги выделяют сильные и слабые границы. Причем слабые границы отмечаются, только если они соединены с сильными границами.

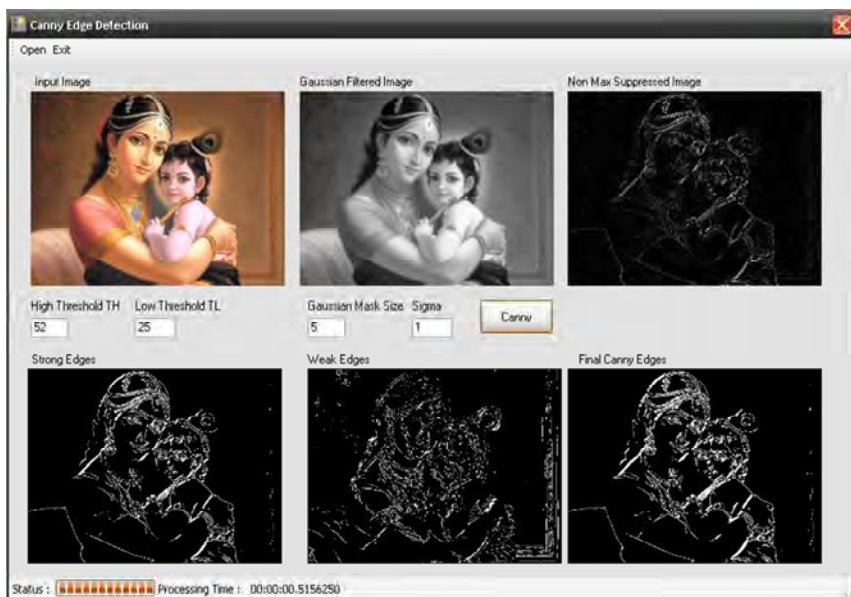


Рис. 10.11. Пример последовательной реализации алгоритма выделения границ методом Кэнни

Алгоритм Кэнни состоит из следующих шагов.

Шаг 1. Сгладить изображение.

Выполнить сглаживание изображения фильтром Гаусса. Радиус и сигма задаются.

Шаг 2. Вычислить градиенты всех пикселей.

Применить фильтр Собела для вычисления значений градиента во всех пикселях изображения. Направление градиента округляется до 0° , 45° , 90° и 135° .

Шаг 3. Подавить не-максимумы модуля градиента.

Краям соответствуют максимальные значения градиентов. Отобрать их, сравнивая значения модуля градиента, ортогонально вычисленным направлениям градиента. Если направление градиента в пикселе равно 0° , анализируется модуль градиента ортогонально расположенных соседних пикселей (вертикальные соседи); пиксель может быть граничным, если его градиент больше, чем у соседей. Аналогично анализируются значения модулей градиента для трех других направления градиента.

После подавления не-максимумов края становятся более точными, тонкими и связными. Простой выбор локальных максимумов выделит мало краевых пикселей.

Шаг 4. Выполнить двойную пороговую фильтрацию.

Для ранее отобранных пикселей применить порог, чтобы уточнить, являются ли они граничными. Чем ниже порог, тем больше граничных пикселей будет отобрано и тем более зашумленным будет результат. Наоборот, высокий порог может отбросить слабые края и получить границу с множеством разрывов.

Для выделения границ используются два значения порога: если значение градиента в пикселе выше верхнего порога, он принимает максимальное значение (граница в нем считается достоверной), если меньше нижнего порога, пиксель не считается граничным. Пиксели со значениями градиента, попадающими в диапазон между порогами, считаются неоднозначными, им присваивают фиксированное среднее значение, а их статус уточняется на следующем шаге.

Значения порогов задаются.

Шаг 5. Уточнить границы путем трассировки области неоднозначности.

Итоговые границы определяются путем подавления всех краевых пикселей, не связанных с ранее определенными достоверными граничными пикселями. Выделяются группы пикселей, получивших на предыдущем этапе среднее значение. Если они соединены с одной из достоверных границ, они относятся к границе, а в противном случае подавляются. Пиксель добавляется к группе, если он соседствует с ней по одному из восьми направлений.

10.5. Нелинейная фильтрация

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

10.5.1. Медианная фильтрация

Если на изображении имеются помехи в виде белых или черных точек, это шум импульсного типа. Линейная фильтрация не устраняет их полностью. Для удаления шума такого типа используются нелинейные фильтры, например медианные.

При медианной фильтрации также используется понятие окрестности и ее центра. Окрестность может иметь произвольную форму и размер, центр может располагаться произвольно относительно окрестности. При совмещении центра окрестности с анализируемым пикселем окрестность служит окном, сквозь которое «видим» и отбираем ряд соседних пикселей, их значения яркости сортируются по возрастанию, значение среднего в ряду (т. е. медианного) элемента после сортировки и будет результатом фильтрации в данном окне. Затем окно смещается, а процедура повторяется для всех пикселей 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 3×3 (рис. 10.12) попали девять пикселей. После сортировки их значений в результате медианной фильтрации центральный пиксель принимает значение 6.



Рис. 10.12. Идея медианной фильтрации в окне 3×3

Поскольку окно может иметь произвольную форму, для его описания удобно использовать прямоугольную маску, элементы которой принимают значения 0 и 1. В сортировке участвуют только пиксели, которые соответствуют ненулевым элементам маски.

Существуют алгоритмы взвешенной медианной фильтрации. В маске, соответствующей окну фильтра, используются, кроме 0 и

1, целые числа 2, 3 и др. Они указывают, сколько раз повторить яркость соответствующего пикселя перед сортировкой. Как правило, большие веса располагаются ближе к центру окна для усиления влияния яркости центральных пикселей на результат. Число значений яркости, участвующих в сортировке, равно сумме весов маски.

Свойства медианного фильтра:

несепарабельный;

нелинейный;

на полутоновых изображениях фильтр не вносит значения яркости, отсутствующие в исходном изображении;

качественно удаляет шумы импульсного типа.

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

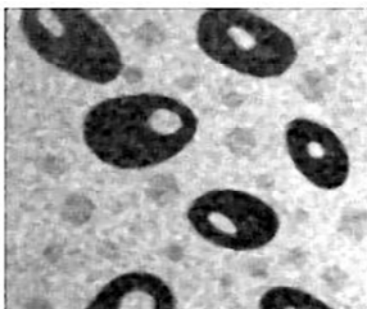
На рис. 10.13 приведен пример медианной фильтрации полутонового зашумленного гауссовским аддитивным шумом, с разными размерами окна фильтрации. Медианный фильтр хорошо справляется со слабой и средней степенью зашумления, однако для подавления более интенсивных шумов необходимо использовать медианный фильтр с большими размерами окна фильтрации.

На рис. 10.13 видно, 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 два раза применить фильтр с окном 3×3 , чем один раз с окном 5×5 (рис. 10.14).

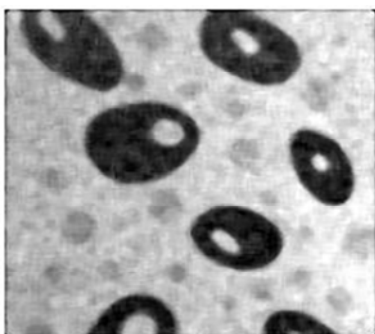
Результат медианной фильтрации нечувствителен к плотности распределения шума. Это связано с тем, что медианный фильтр с площадью окна $2M + 1$ эффективно подавляет шумовые пиксели в окне суммарной площадью менее M . В то же время при фильтрации контрастных крупноразмерных объектов медианный фильтр не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



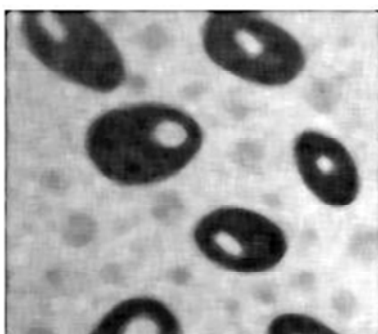
Исходное изображение
(сильная степень зашумления)



Результат медианной
фильтрации в окне 5×5



Результат медианной
фильтрации в окне 7×7



Результат медианной
фильтрации в окне 9×9

Рис. 10.13. Примеры фильтрации с разным размером окна



Рис. 10.14. Пример трехкратного применения медианного фильтра.
Шум занимает 15 % площади изображения

10.5.2. Адаптивная медианная фильтрация

Рассмотрим вариант адаптивного медианного фильтра. Он использует прямоугольное окно размером $S \times S$. Адаптация состоит в увеличении размера окна S в процессе фильтрации в зависимости от его локальных статистик.

Введем обозначения:

Z_{\min} – минимальное значение в окне;

Z_{\max} – максимальное значение в окне;

Z_{med} – медианное значение 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. Организовать цикл по строкам и столбцам.

Шаг 2. Задать исходные размеры окна фильтрации. Для пикселя (i, j) в заданном окне вычислить Z_{\min} , Z_{\max} , Z_{med} , $A_1 = Z_{\text{med}} - Z_{\min}$, $A_2 = Z_{\text{med}} - Z_{\max}$.

Если $A_1 > 0$ и $A_2 < 0$, перейти на *шаг 2*.

В противном случае следует увеличить размер окна.

Если текущий размер окна $S \leq S_{\max}$, повторить *шаг 1*.

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

Шаг 3. 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_{med} .

Шаг 4. Изменить координаты (i, j) . Если не вышли за пределы изображения, установить исходные размеры окна фильтрации, перейти на *шаг 2*.

В противном случае фильтрация окончена.

Цель *шага 2* – определить, является ли выход медианного фильтра Z_{med} импульсным шумом (черным или белым). Если условие $Z_{\min} < Z_{\text{med}} < Z_{\max}$ соблюдается, то Z_{med} не может быть импульсом. В этом случае осуществляется переход на *шаг 3* и проверяется, является ли импульсом пиксель $Z_{i,j}$ в центре окна?

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_{med} . В этом случае

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. 10.15).

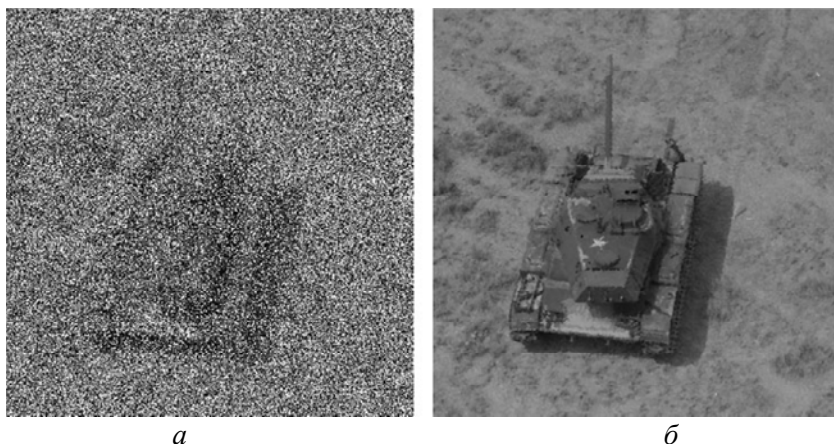


Рис. 10.15. Пример фильтрации адаптивным медианным фильтром:
а – слева изображение с импульсным шумом; *б* – результат
фильтрации описанным алгоритмом

Если условие $B_1 > 0$ и $B_2 < 0$ неверно, тогда $Z_{i,j} = Z_{\min}$ или $Z_{i,j} = Z_{\max}$. И в том и другом случае значение пикселя является критическим и алгоритм выводит медиану Z_{med} , которая, как следует из *шага 2*, не является импульсным шумом.

Предположим, что на *шаге 2* обнаружен импульс. Тогда увеличивается размер окна и повторяется *шаг 2*. Повтор шага происходит до тех пор, пока алгоритм не находит среднее значение, которое не является импульсом (и выполняет переход к *шагу 3*), 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 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 медианного фильтра аналогична алгоритму стандартного.

Таким образом, рассмотренная модификация медианного фильтра преследует следующие цели:

оптимальное удаление импульсного шума;

сглаживание других типов шумов;

уменьшение искажений в виде потери мелких деталей.

Основной недостаток адаптивных методов обработки – увеличение объема вычислений. Для уменьшения вычислений используют рекурсивные процедуры коррекции статистических при нахождении характеристик соседнего фрагмента характеристик текущего 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 пикселей, образующих левый столбец текущего фрагмента, и добавлением пикселей, образующих правый столбец соседнего фрагмента.

10.5.3. 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.

10.6. Внесение эффектов в изображение

Одним из способов обработки изображений является их изменение путем внесения различных эффектов.

10.6.1. 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 пикселем и соседями используются отрицательные весовые коэффициенты.

Пример маски для увеличения резкости изображения:

$$G = \begin{vmatrix} -0,1 & -0,1 & -0,1 \\ -0,1 & 1,8 & -0,1 \\ -0,1 & -0,1 & -0,1 \end{vmatrix}. \quad (10.19)$$

Сумма коэффициентов маски равна +1, но может и немного отличаться.

10.6.2. Создание эффекта тиснения

Эффект тиснения можно придать изображению с помощью фильтров следующего типа:

$$T_4 = \begin{vmatrix} 0 & 1 & 2 \\ 0 & 0 & 0 \\ 0 & -1 & -2 \end{vmatrix};$$

$$T_5 = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{vmatrix}; \quad (10.20)$$

$$T_6 = \begin{vmatrix} 2 & -1 & 0 \\ -1 & 0 & -1 \\ 0 & -1 & 2 \end{vmatrix}.$$

Коэффициенты ядра симметричны относительно центра или оси, проходящей через центр, а их сумма равна нулю.

При изменении знаков в ядрах фильтра на противоположные изменяется направление подсветки.

10.6.3. Создание эффекта акварелизации

В результате применения эффекта акварелизации изображение будет выглядеть так, будто оно нарисовано акварелью.

На первом этапе применения сглаживается яркость редактируемого изображения. Можно использовать свертку, например, с ядром

$$M = \frac{1}{16} \begin{vmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{vmatrix}. \quad (10.21)$$

На следующем этапе для завершения создания эффекта акварели повышается резкость переходов полученного изображения, например сверткой с ядром (рис. 10.16):

$$R = \begin{vmatrix} -0,5 & -0,5 & -0,5 \\ -0,5 & 5 & -0,5 \\ -0,5 & -0,5 & -0,5 \end{vmatrix}. \quad (10.22)$$



Рис. 10.16. К исходному изображению (а) применили медианный фильтр в окне 5×5 и свертку с ядром R , результирующим изображение (б)

10.7. «Продвинутые» алгоритмы

Вышеописанные алгоритмы не учитывают локальные особенности изображений, поэтому, если изображение содержит и очень тем-

ные, и очень светлые участки, эти алгоритмы не могут обеспечить качественное улучшение изображения. Для таких изображений применяют так называемые «продвинутые» алгоритмы, например алгоритм Retinex и его модификации.

Алгоритм SSR (Single-Scale Retinex) выравнивает освещенность изображения, сохраняя локальный контраст в плохо и ярко освещенных областях (рис. 10.17). Коррекция выполняется по формуле

$$R(x, y, \sigma) = \log[I(x, y)] - \log[I(x, y) * G(x, y, \sigma)], \quad (10.23)$$

где σ – коэффициент размытия;

* – оператор свертки;

G – Гауссиан.



Рис. 10.17. Пример применения алгоритма Single-Scale Retinex при $\sigma = 18$ (а – исходное изображение)

После применения формулы (10.23) большинство значений R находится в диапазоне $[-1; 1]$, для визуализации изображения значения R необходимо нормализовать по формуле

$$I = \text{Int}(255 * R) + 127. \quad (10.24)$$

Этот алгоритм можно применить несколько раз с разными параметрами фильтра Гаусса и вычислить взвешенную сумму яркостей каждого пикселя. Получим алгоритм MSR (Multi-Scale Retinex), (рис. 10.18):

$$\begin{aligned} MSR &= w_1 SSR_1 + w_2 SSR_2 + \dots + w_n SSR_n; \\ w_1 + w_2 + \dots + w_n &= 1. \end{aligned} \quad (10.25)$$

Обычно n невелико: 3 или 5.



Рис. 10.18. Пример применения алгоритма Multi-Scale Retinex для $\sigma = 7, 480, 4800$ (*a*), плюс линейное растяжение контраста (*б*)

10.8. Сравнение фильтров

Выбор конкретного фильтра зависит от задачи и условий ее реализации, например скорости вычислений.

Арифметический, усредняющий фильтр, помимо подавления шума, искажает резкие границы и размывает мелкие детали изображения; он самый быстрый.

Фильтр Гаусса меньше размывает мелкие детали и лучше убирает шум, достаточно быстр, если коэффициенты вычисляются один раз.

Медианный фильтр (median filter) резких границ не искажает, убирает мелкие детали, изображение становится более однородным. Если маска велика, сортировка замедляет вычисления.

Адаптивные фильтры меньше искажают детали и зависят от большего числа параметров, требующих дополнительных вычислений.

«Продвинутые» фильтры обеспечивают лучшее сохранение деталей на изображении, меньше размытие. Они часто сложны в реализации и работают медленнее.

Глава 11. ОПЕРАЦИИ МАТЕМАТИЧЕСКОЙ МОРФОЛОГИИ

Одним из сравнительно новых направлений в обработке изображений является математическая морфология. Это теория обработки и анализа геометрических форм, основанная на теории множеств и топологии. Напомним основные понятия из теории множеств.

Если $X = \{x\}$ и $Y = \{y\}$ – множества, то объединением множеств X и Y называется множество $Z = \{z : z \in X \text{ или } z \in Y\} = X \cup Y$ (т. е. 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

$$Z = X^C = \{z : z \in X\}$$

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 такое множество $Z = \emptyset$. Справедливы следующие соотношения:

$$\begin{aligned}(X \cup Y)^C &= X^C \cap Y^C; \\ (X \cap Y)^C &= X^C \cup Y^C; \\ X \setminus Y &= X \cap Y^C;\end{aligned}\tag{11.1}$$

Бинарные изображения легко описать в виде множеств. Если яркость пикселя равна 1 (белый цвет), он принадлежит некоторому множеству, если 0 (черный) – нет.

Бинарное изображение можно рассматривать как индикаторную функцию набора множеств, т. е. черных пятен, например объединения множеств $X = X_1 \cup X_2 \cup X_3$ на рис. 11.1.

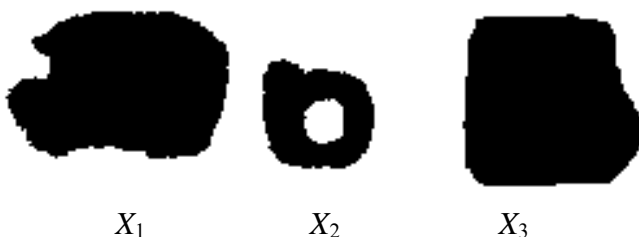


Рис. 11.1. Бинарное изображение

К такому описанию бинарных изображений применим аппарат теории множеств.

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 пикселей.

Одним из основных понятий математической морфологии является понятие структурного элемента. *Структурный элемент V* – это множество точек, для которых определен его центр (не обязательно являющийся центром этого множества). Важно, что структурный элемент определяет некоторую геометрическую форму, с помощью которой выполняются морфологические операции. Элемент 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 для обработки бинарных изображений.

11.1. Математическая морфология бинарных изображений

11.1.1. Эрозия

Пусть задан структурный элемент V . *Эрозией* множества X называется множество Y , состоящее из тех элементов исходного множества X , для которых выполняется условие $V_x \subset X$ (рис. 11.2):

$$Y = X \ominus B = \{x : B_x \subset X\}. \quad (11.2)$$

Так, если $x \in X$, а $b \in B$, то в множество Y включаются такие элементы, для которых выполняется условие

$$x+b \in X. \quad (11.3)$$

Эрозию можно трактовать следующим образом: центр структурного элемента помещают во все точки $x \in X$, если элемент полностью принадлежит X , тогда точка $x \in Y$. Очевидно, после этой операции множество X уменьшается.

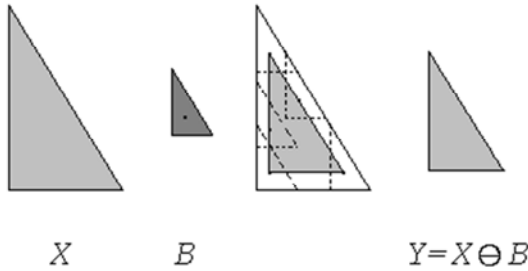


Рис. 11.2. Операция эрозии

С другой стороны, если точка b пробегает все возможные положения в множестве B , условие (11.3) выполняется тогда и только тогда, когда x принадлежит смещенному множеству X_{-b} (рис. 11.3).

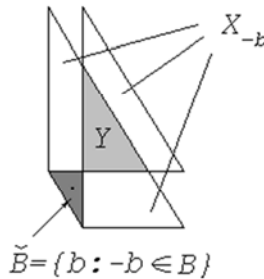


Рис. 11.3. Эрозия как пересечение смещенных множеств

Поэтому операцию эрозии можно записать

$$Y = X \ominus B = \bigcap_{b \in B} X_{-b} = \bigcap_{-b \in B} X_b = \bigcap_{b \in \bar{B}} X_b, \quad (11.4)$$

где $\bar{B} = \{b : -b \in B\}$ – множество, симметричное элементу B относительно его центра.

11.1.2. Дилатация

Операцией, двойственной к эрозии, является дилатация (или расширение), которая определяется следующим образом (рис. 11.4):

$$Y = X \oplus B = \left\{ x : B_x \cap X \neq \emptyset \right\}. \quad (11.5)$$

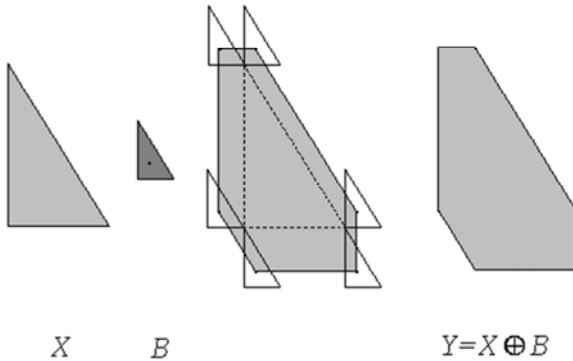


Рис. 11.4. Операция дилатации

Другое представление дилатации имеет вид (рис. 11.4):

$$Y = X \oplus B = \bigcup_{b \in B} X_b. \quad (11.6)$$

Дилатацию можно осуществить следующим образом: центр структурного элемента помещаем во все точки $x \in X$ и добавляем

к X все точки смещенного структурного элемента. Объединение полученных точек определяет множество Y . Очевидно, после этой операции множество X увеличивается.

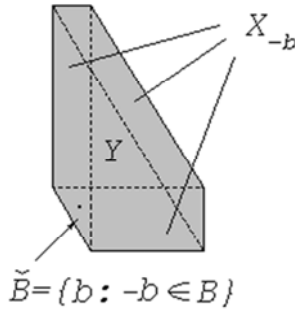


Рис. 11.5. Дилатация как объединение смешанных множеств

Если рассматривать множество X как объект, а его дополнение X^C как фон на изображении, то дилатацию объекта можно интерпретировать как эрозию фона:

$$X \oplus B = (X^C \ominus B)^C. \quad (11.7)$$

Доказательство тривиально:

$$(X^C \ominus B)^C = \left(\bigcap_{b \in B} X_b^c \right)^c = \bigcup_{b \in \bar{B}} X_b = X \oplus B. \quad (11.8)$$

11.1.3. Свойства дилатации и эрозии

Эти две операции являются базовыми в математической морфологии. Приведем их основные свойства.

1. Дистрибутивность. Дилатация дистрибутивна относительно объединения множеств:

$$\left(\bigcup_{i=1}^I X_i \right) \oplus B = \bigcup_{i=1}^I (X_i \oplus B), \quad (11.9)$$

а эрозия – относительно пересечения множеств:

$$\left(\bigcup_{i=1}^I X_i\right)\ominus B = \bigcup_{i=1}^I (X_i\ominus B). \quad (11.10)$$

Свойство дистрибутивности с учетом соотношения (11.9) позволяет выполнять операции над X по фрагментам, комбинируя затем результаты посредством объединения или пересечения.

2. Ассоциативность. Дилатация ассоциативна:

$$(X \oplus A) \oplus B = X \oplus (A \oplus B), \quad (11.11)$$

а эрозия нет, но верно

$$(X \ominus A) \ominus B = X \ominus (A \oplus B). \quad (11.12)$$

Эти свойства позволяют упрощать выполнение операций посредством разложения сложных структурных элементов в композицию более простых (рис. 11.6). 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 на рис. 11.6, можно заменить четырьмя последовательными операциями эрозии со структурными элементами A, B, C, D , показанными на данном рисунке.

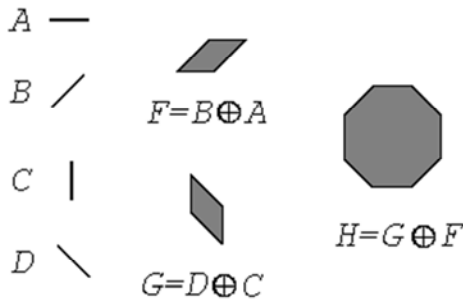


Рис. 11.6. Декомпозиция структурных элементов

3. Инвариантность к изменению масштаба

$$\lambda X \oplus \lambda B = \lambda(X \oplus B);$$

$$\lambda X \ominus \lambda B = \lambda(X \ominus B).$$

В этих соотношениях через λB , λA обозначены множества, промасштабированные относительно своих центров (рис. 11.7)

$$B' = \lambda B = \{b' = \lambda b : b \in B\};$$

$$X' = \lambda X = \{x' = \lambda x : x \in X\}.$$

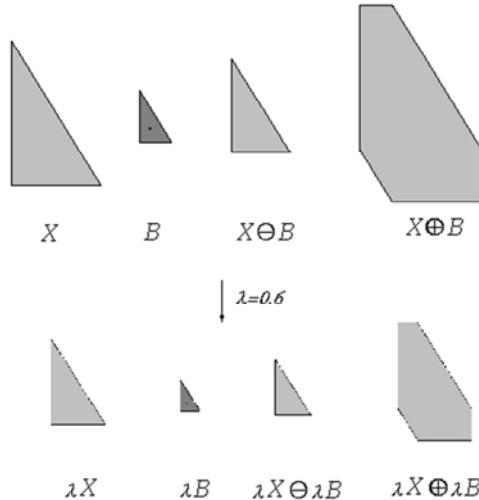


Рис. 11.7. Инвариантность эрозии и дилатации к масштабу

Если центр структурного элемента B находится внутри него, тогда для любых множеств A и C , если $A \subseteq C$, справедливо:

$$A \ominus B \subseteq C \ominus B;$$

$$A \oplus B \subseteq C \oplus B.$$

11.1.4. Применение эрозии и дилатации

Эрозия и дилатация – операции, предназначенные в первую очередь для выделения на бинарном изображении объектов (связных множеств точек) с определенной формой путем использования структурных элементов с разной формой и размерами. 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, смещение между которыми определяется вектором h , эрозия позволит выделить объекты, имеющие соседей в направлении и на расстоянии, заданными этим вектором (рис. 11.8). Под объектами подразумеваются односвязные множества пикселей.

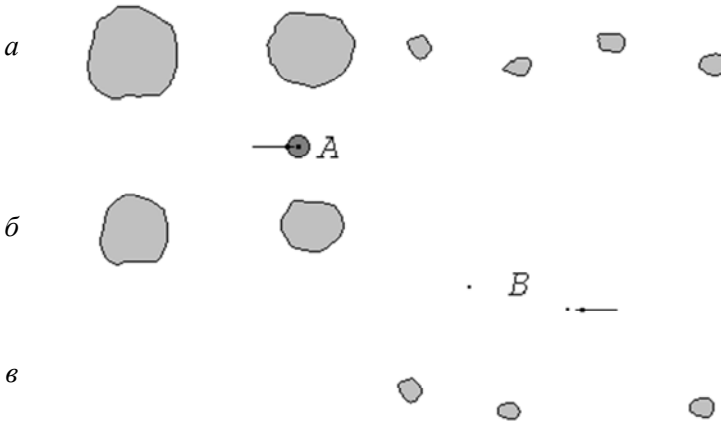


Рис. 11.8. Множество пикселей: a – исходное множество X ; b – $X \oplus A$; $в$ – $X \oplus B$. Стрелками отмечены начала структурных элементов

Одно из простейших применений дилатации – это устранение разрывов линий путем их перекрытия. На рис. 11.9 показано изображение с разорванными символами. Пусть известно, что максимальная длина разрывов составляет два пикселя. На рис. 11.9, b показан структурный элемент, позволяющий устранить разрывы. Результат дилатации исходного изображения по такому примитиву показан на рис. 11.9, $в$. 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. Напротив, при низкочастотной фильтрации из исходного бинарного изображения вначале получается полутоновое, которое затем требуется преобразовать обратно в бинарное.

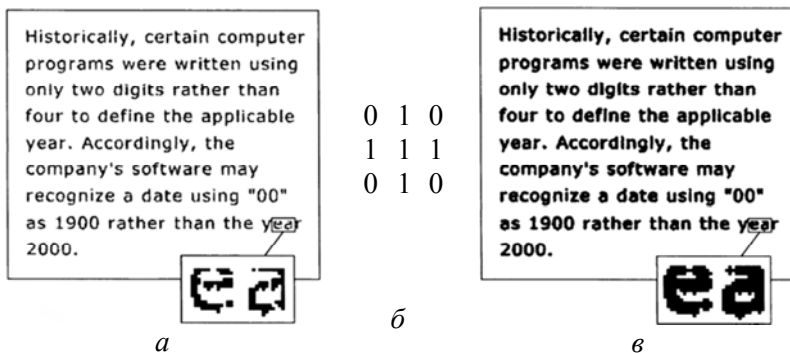


Рис. 11.9. Пример изображения текста с недостаточным разрешением и разным размером букв (а); структурный элемент (б); (в) дилатация изображения (а) с элементом (б)

Рассмотрим способ морфологического выделения границ (рис. 11.10). Определим морфологическую границу множества X (т. е. всех темных пятен на изображении) как разность дилатации и эрозии этого множества с круглым структурным элементом B :

$$\delta X = X - (X \ominus B). \quad (11.16)$$



Рис. 11.10. Пример морфологического выделения границы

11.1.5. 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

$$A \circ B = (A \ominus B) \oplus B. \quad (11.17)$$

Операция **замыкания** множества X посредством структурного элемента B определяется применением эрозии и дилатации в обратном порядке:

$$A \bullet B = (A \oplus B) \ominus B. \quad (11.18)$$

Рассмотрим основные свойства этих операций. Они двойственны друг другу:

$$A \bullet B = (A^c \circ B^s)^c. \quad (11.19)$$

Применительно к бинарным изображениям это соотношение 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 этих операций, их идемпотентность:

$$(A \circ B) \circ B = A \circ B \quad \text{и} \quad (A \bullet B) \bullet B = A \bullet B. \quad (11.20)$$

Оно означает неизменность результата при повторении операций отмыкания и замыкания.

Если центр структурного элемента B находится внутри него, тогда для любого множества A справедливо

$$A \ominus B \subseteq A \circ B \subseteq A \subseteq A \bullet B \subseteq A \oplus B. \quad (11.21)$$

11.1.6. Применение операций отмыкания и замыкания

Отмыкание и замыкание используются для фильтрации бинарных изображений, сглаживания формы объектов путем удаления мелких и узких частей (операция отмыкания); удаления мелких полостей и узких «каналов» (операция замыкания). 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 :

$$gX_m = X \circ B - X \bullet B. \quad (11.22)$$

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 отсекал 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:

$$hX = (X \circ B_1) \bullet B_1 - (((X \circ B_1) \bullet B_1) \circ B_2) \bullet B_2. \quad (11.23)$$

Комбинируя более сложные последовательности базовых операций математической морфологии, можно сконструировать другие операции, ориентированные на решение конкретных задач.

11.1.7. Операция утоньшения

Цель *утоньшения* – получить связанное множество пикселей толщиной не более одного. Эта операция выполняется при обработке изображений, содержащих тонкие линии (чертежи, текст, отпечатки пальцев и т. п.).

Операция утоньшения определяется следующим образом:

$$Y = XOT = X \setminus (X \otimes T), \quad (11.24)$$

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 $X \circ T = X$ при любом T_1 . Чтобы избежать этого при выполнении операции утоньшения, начало структурного элемента не должно принадлежать T_1 .

Модифицированный алгоритм Зонга – Суня

Исходное изображение бинарное, значения яркости 0 и 1. Будем использовать следующую нумерацию соседей центрального пиксела P_1 и обозначения:

$$\begin{bmatrix} P_9 & P_2 & P_3 \\ P_8 & P_1 & P_4 \\ P_7 & P_6 & P_5 \end{bmatrix}.$$

Обозначим $A(P_1)$ – число переходов яркости с 0 на 1 в последовательности $P_9, P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9$, а также

$$B(P_1) = \sum_{i=2}^9 P_i,$$

\bar{P}_i – отрицание P_i .

Шаг 1. В цикле по i и j для каждого пиксела $P(i, j)$ изображения вычислить $P_1 = P(i, j)$.

Если $2 \leq B(P_1) \leq 6$

$A(P_1) = 1$ и

а) одно из условий верно:

1) $P_2 \times P_4 \times P_6 = 0$ при нечетной итерации;

2) $P_2 \times P_4 \times P_8 = 0$ при четной итерации;

б) одно из условий верно:

1) $P_4 \times P_6 \times P_8 = 0$ при нечетной итерации,

2) $P_2 \times P_6 \times P_8 = 0$ при четной итерации,

тогда удалить пиксель $P(i, j)$, т. е. значение его яркости заменить на фоновое.

В противном случае, если $A(P_1) = 2$:

а) одно из условий верно

1) $P_4 \times P_6 = 1$ и $P_9 = 0$ при нечетной итерации,

2) $P_2 \times P_8 = 1$ и $P_5 = 0$ при четной итерации,

б) одно из условий верно

1) $P_4 \times P_2 = 1$ и $\bar{P}_3 \times \bar{P}_7 \times \bar{P}_8 = 1$ при нечетной итерации,

2) $P_6 \times P_8 = 1$ и $\bar{P}_3 \times \bar{P}_4 \times \bar{P}_7 = 1$ при четной итерации,

тогда удалить пиксель $P(i, j)$, т. е. значение его яркости заменить на фоновое.

Шаг 2. Если на предыдущем шаге были выполнены изменения, повторить *шаг 1* (иначе – закончить работу).

Пример выполнения операции утоньшения представлен на рис. 11.11.



Рис. 11.11. Пример выполнения операции утоньшения:
а – исходное бинарное изображение; *б* – результат утоньшения

Скелетон множества точек, заданного на дискретной решетке, только приблизительно 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 структурных элементов в последовательности (топологические свойства скелетона, такие как количество связанных компонент, точек разветвления, ветвей, концевых точек и тому подобное при этом сохраняются).

11.2. 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

$$(f \oplus B)(x) = \max\{f(x - z), z \in B\};$$
$$(f \ominus B)(x) = \max\{f(x + z), z \in B\}.$$
(11.25)

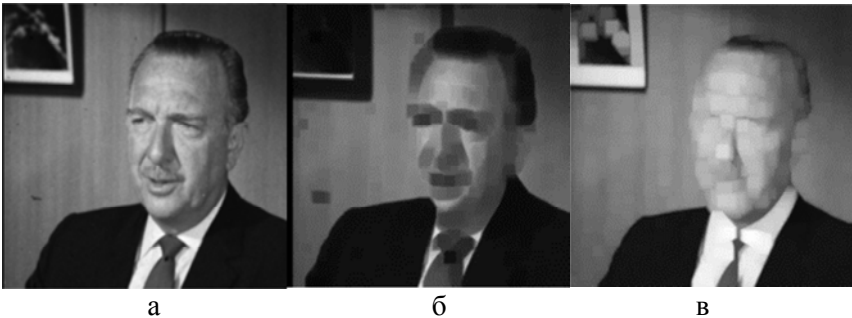


Рис. 11.12. Примеры эрозии (б) и дилатации (в) плоским структурным элементом квадратной формы

Эрозию полутонового изображения плоским структурным элементом можно представить наглядно: представим, что функция яркости задает некую поверхность; двигаем плоский структурный элемент внутри поверхности по ее краю, так чтобы элемент полностью поместился внутри. Очерченная элементом новая поверхность будет результатом.

Дилатацию полутонового изображения плоским структурным элементом можно трактовать следующим образом. Функция яркости задает некую поверхность. При дилатации (если структурный

элемент несимметричен относительно своего центра) преобразуем его, вычислив новый, симметричный относительно центра элемент. Двигаем структурный элемент по поверхности, расширяя ее в стороны в соответствии с формой элемента.

Рассмотрим пример выполнения этих операций. Пусть плоский структурный элемент несимметричен и задан единицами $[0 \ 1 \ 1]$, центр элемента – его геометрический центр, а строка яркостей изображения имеет значения

0 0 8 8 9 0 0.

Применяя дилатацию, находим максимум в окрестности каждого пикселя, с которым совмещаем центр элемента, окрестность отмечена единицами $[1 \ 1 \ 0]$. В результате яркости 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 примут значения

0 8 8 0 0 0.

Глава 12. МЕТОДЫ ОБРАБОТКИ ЦВЕТНЫХ ИЗОБРАЖЕНИЙ

Методы обработки цветных изображений делятся на два основных типа:

- 1) цветовые компоненты обрабатываются по отдельности;
- 2) цвет рассматривается как единый вектор в трехмерном пространстве.

В методах первого типа фактически выполняется обработка R , G , B -цветов как отдельных полутоновых изображений. Однако в результате 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.

12.1. Изменение цветового баланса

Цель методов изменения цветового баланса – компенсация неверного цветового восприятия камеры и/или цветного освещения сцены (рис. 12.1).

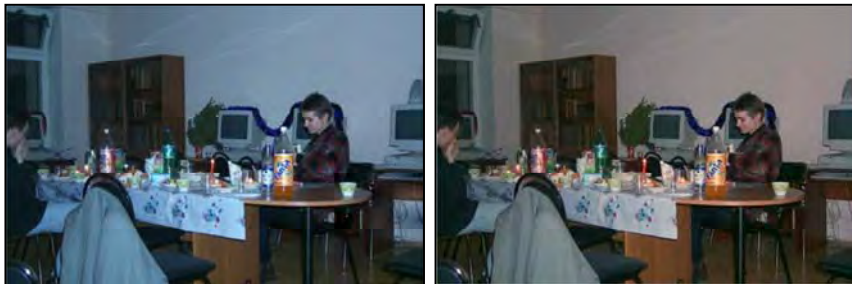


Рис. 12.1. Коррекция изображения с опорным цветом

Предполагается, что пользователь интерактивно укажет нужный опорный цвет, например белый цвет – цвет облаков на фотографии. Либо источникам «правильных» цветов может быть фотография этой же сцены с хорошо переданными цветами.

Преобразование по каждому из каналов цвета осуществляется по формулам

$$\begin{aligned}R &= R \frac{R_{\text{real}}}{R_{\text{orig}}}; \\G &= G \frac{G_{\text{real}}}{G_{\text{orig}}}; \\B &= B \frac{B_{\text{real}}}{B_{\text{orig}}},\end{aligned}\tag{12.1}$$

где индекс «orig» означает цвет пикселя на исходном изображении, «real» – желаемый цвет.

12.2. Идеальный отражатель

Предполагается, что наиболее яркие области изображения относятся к бликам на поверхностях, модель отражения которых такова, что цвет блика равен цвету освещения (дихроматическая модель).

Алгоритм коррекции

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

$$\begin{aligned}R &= R \frac{255}{R_{\text{max}}}; \\G &= G \frac{255}{G_{\text{max}}}; \\B &= B \frac{255}{B_{\text{max}}}.\end{aligned}\tag{12.2}$$

12.3. Преобразование «Серый мир»

В основе этого преобразования лежит предположение, что сумма всех цветов на изображении естественной сцены дает серый цвет. Пример коррекции изображения представлен на рис. 12.2.



Рис. 12.2. Пример коррекции цвета преобразованием «Серый мир»

Алгоритм преобразования «Серый мир»

Шаг 1. Вычислить средние значения всех цветов по формулам:

$$\begin{aligned}\bar{R} &= \frac{1}{N} \sum R(x, y); \\ \bar{G} &= \frac{1}{N} \sum G(x, y); \\ \bar{B} &= \frac{1}{N} \sum B(x, y); \\ Avg &= \frac{\bar{R} + \bar{G} + \bar{B}}{3}.\end{aligned}\tag{12.3}$$

Шаг 2. Скорректировать значения цветов согласно коэффициентам

$$\begin{aligned}R' &= R \frac{Avg}{R}; \\G' &= G \frac{Avg}{G}; \\B' &= B \frac{Avg}{B},\end{aligned}\tag{12.4}$$

где R' , G' , B' – новые значения цвета по каждому из каналов.

12.4. Линейное растяжение контраста цветного изображения

Идея метода – линейно растянуть интенсивности по каждому цветовому каналу на весь динамический диапазон. Этот метод используется в пакете «Фотошоп» в функции «autolevels».

Алгоритм линейного растяжения контраста

Шаг 1. Найти минимум и максимум по каждому из каналов (R_{\min} , R_{\max} , G_{\min} , G_{\max} , B_{\min} , B_{\max}).

Шаг 2. Преобразовать оттенки цвета каждого пикселя:

$$\begin{aligned}(R - R_{\min}) &\frac{(255 - 0)}{(R_{\max} - R_{\min})}; \\(G - G_{\min}) &\frac{(255 - 0)}{(G_{\max} - G_{\min})}; \\(B - B_{\min}) &\frac{(255 - 0)}{(B_{\max} - B_{\min})}.\end{aligned}\tag{12.5}$$

12.5. Автоматическое вычисление порога бинаризации цветного изображения

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. Однако в этом методе используется константа, которая должна подбираться интерактивно.

Метод Эйквеля часто применяют для обработки четких и контрастных изображений. Согласно данному методу изображение обрабатывается с помощью двух концентрических окон. Но при обработке тонких пересекающихся линий могут возникать разрывы, поэтому данный метод рекомендуется применять исключительно для толстых линий и крупных объектов.

Метод Яновица и Брукштейна лучше всего обрабатывает изображения, которым свойственна яркостная зональная неравномерность, при которой 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 Отсу, который использует гистограмму распределения значений яркости растрового изображения. Этот метод дает хороший результат разделения объектов и однородного фона на полутоновых изображениях. К недостаткам метода Отсу можно отнести размытие и потерю тонких линий, «слипание» объектов в местах их пересечений, а также долгий процесс вычисления значения порога T .

Для полной автоматизации и адаптации к разным условиям получения изображений при вычислении порога T рекомендуется метод, который называется алгоритм бинаризации k -средних.

Пусть задана некоторая начальная величина порога T , она может быть любой, тогда

$$\forall x \in 1 \dots M-2 \quad \forall y \in 1 \dots N-2 \quad \exists B(x, y) = \begin{cases} 1, & \nabla F \geq T; \\ 0, & \nabla F < T, \end{cases} \quad (12.6)$$

где M, N – соответствующие значения ширины и высоты изображения;

$B(x, y)$ – бинарное изображение (функция, принимающая значения 0 или 1);

∇F – градиент изображения.

$$\begin{aligned} \forall B(x, y) = 1, \quad H_i &= R(x, y) + G(x, y) + B(x, y), \quad i \in 1 \dots \alpha; \\ \forall B(x, y) \neq 1, \quad L_j &= R(x, y) + G(x, y) + B(x, y), \quad j \in 1 \dots \beta, \end{aligned} \quad (12.7)$$

где $R(x, y), G(x, y), B(x, y)$ – соответствующие значения интенсивности красной, зеленой и голубой компоненты на исходном мультиспектральном изображении;

α, β – соответствующие значения количества точек перепада и всех оставшихся.

$$T = \frac{1}{2} \left(\frac{\sum_{i=1}^{\alpha} H_i + \sum_{j=1}^{\beta} L_j}{(M-2) \cdot (N-2)} \right) \quad (12.8)$$

Из (12.1)–(12.3) видно, что вначале проводится бинаризация изображения с использованием любой оценки порога T . В результате образуются две группы пикселей: H – состоящая из пикселей со значением ∇F , большим либо равным T , и L – состоящая из пикселей со значением $\nabla F < T$. Далее находится половина среднего значения яркости пикселей полноцветного изображения по областям H и L соответственно. Полученный результат и будет являться требуемым глобальным порогом T для всего изображения, при использовании которого и происходит бинаризация по необходимому для задач обработки изображений в масштабе реального времени способу.

В оригинальном методе бинаризации k -средних наличие двух неизвестных факторов (начальная оценка значения порога, для вычисления которой предлагается ряд методов, и значение наперед заданного параметра T_0 , которое служит ориентиром для количества итераций данного алгоритма) не позволяет с уверенностью применять его на практике, учитывая возникающую неопределенность со 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 коэффициент, учитывающий восприятие последних в видимом диапазоне длин волн. В противном случае приведенный метод не будет обеспечивать нужное качество бинаризации изображения.

На рис. 12.3 представлен пример выполнения бинаризации изображения с применением оператора Собела и автоматическим вычислением порога бинаризации T .



Рис. 12.3. Пример выполнения бинаризации:
a – исходное мультиспектральное изображение;
б – бинарное изображение, полученное
из *a* вышеописанным методом

12.6. Повышение четкости цветного портрета

Повысить четкость цветного портрета можно разными способами. Один из них – преобразовать RGB-изображение в пространство Lab. Это 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 – в полном разделении цвета и контраста. В моделях RGB и CMYK все каналы 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, 50%-ми 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 не пурпурный, не 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 всегда является промежуточным этапом коррекции. Файлы переводятся 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.

Опишем алгоритм повышения четкости портрета.

Шаг 1. Открыть изображение. Преобразовать из RGB в Lab.

Шаг 2. К полутоновому изображению канала L применить фильтр типа нерезкое маскирование, получим L_{new} .

Шаг 3. Преобразовать изображение $L_{\text{new}}ab$ в изображение $R_{\text{new}}G_{\text{new}}B_{\text{new}}$.

На рис. 12.4 приведен пример применения описанного алгоритма.

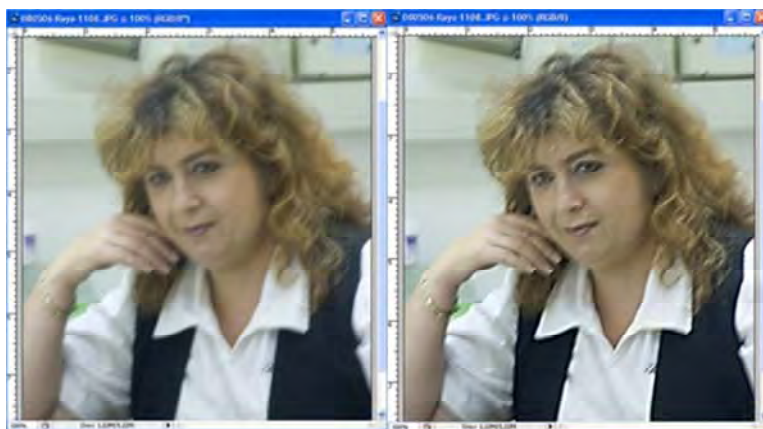


Рис. 12.4. Пример повышения четкости в пространстве Lab

Параметры максимального увеличения четкости применяются:

1) если портрет не в фокусе, поэтому необходимо максимально увеличить его четкость, чтобы попытаться вернуть объекту снимка первоначальную резкость;

2) на снимке много объектов с ярко выраженными границами (например, архитектурные сооружения, монеты, автомобили, какие-либо механизмы и т. п.).

Для увеличения резкости в пакете «Фотошоп» используется фильтр Unsharp Mask с параметрами: Amount (величина) – 65 %, Radius (радиус) – 4 и Threshold (порог) – 3. Данные параметры дают максимальное увеличение четкости для сильно расфокусированных изображений, причем фильтр Unsharp Mask с одними и теми же параметрами применяется дважды.

Описанный алгоритм часто применяется как к снимкам, сделанным цифровой камерой в неблагоприятных условиях, так и к отсканированным. При применении этого алгоритма для увеличения четкости портрета, сделанного крупным планом, на лице часто появляются отчетливо видимые поры, морщины и другие недостатки кожи. Если увеличить четкость всех компонент RGB-изображения или четкость только канала L, используя Unsharp Mask, то текстура кожи будет выглядеть грубее, чем на исходном изображении.

В этом случае можно выполнить обработку в RGB-пространстве, применив к каналу Red (Красный) фильтр Unsharp Mask, т. е. изменить четкость только одного канала Red, а остальные использовать без изменений. Красный канал, как правило, содержит наименьшее количество мелких деталей изображения, поэтому применение фильтра Unsharp Mask к одному этому каналу позволяет повысить четкость только нужных областей лица (глаз, губ и т. д.); при этом четкость текстуры кожи остается практически неизменной, гладкой (рис. 12.5). Если вместо Red использовать канал Blue, то увеличится интенсивность текстуры кожи.



Рис. 12.5. Пример повышения четкости изменением только канала R

Незначительное увеличение четкости снимков 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 Unsharp Mask (Amount – 150 %, Radius – 1 и Threshold – 10).

Для увеличения четкости web-изображений используют значения Amount – 400 %, Radius – 0,3 и Threshold – 0. Изображение получается мягким и немного размытым. Перед размещением его в Web разрешение высококачественного снимка уменьшается с 300 до 72 dpi. Если интенсивность эффекта слишком велика, значение параметра Amount можно уменьшить до 200 %. Значение параметра Amount (400 %) используется для сильно расфокусированных снимков.

Глава 13. СЕГМЕНТАЦИЯ ИЗОБРАЖЕНИЙ

Часто изображение представляется для наблюдателя в виде совокупности однородных участков. Количество различных типов (или классов) участков обычно невелико, все изображение может быть разделено на непересекающиеся области, каждая из которых заполнена изображением одного из типов. При анализе таких изображений целью является выделение таких 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 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 , – таких, что

$$\text{а) } \bigcup_{i=1}^n R_i = R;$$

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 $i \neq j$.

Условие а) означает, что должна быть сегментирована вся область изображения. Условие б) требует, чтобы пиксели одной области были связаны. Условие в) – чтобы области не пересекались; условие г) требует однородности пикселей одного класса. Последнее условие требует различимости разных классов.

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 может служить цвет, который дополняет совокупность характеристик, применяемых при обработке черно-белых изображений.

Единого или универсального подхода к решению задачи сегментации не существует. Существует множество алгоритмов сегментации. Каждый из них предназначен для работы с определенным классом изображений и для решения определенных задач. Основная идея большинства алгоритмов заключается в нахождении соответствий или различий в характеристиках пикселей. Характеристиками могут быть их взаимное расположение, яркость, цвет и текстура.

Основные методы сегментации:

на основе пороговой обработки;

наращивание областей;

разделение и слияние областей;

сегментация на базе морфологического водораздела.

13.1. Сегментация изображений по яркости на основе пороговой обработки

13.1.1. Вычисление порогов по гистограмме

Пороговая обработка, вероятно, самый простой метод сегментации. Метод ориентирован 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 каждой точки исходного изображения $x(i_1, i_2)$ в выходное $y(i_1, i_2)$ выполняется по правилу

$$y(i_1, i_2) = \begin{cases} y_0 & \text{при } x(i_1, i_2) \leq T; \\ y_1 & \text{при } x(i_1, i_2) > T, \end{cases} \quad (13.1)$$

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 карты – 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 должно быть задано семейство порогов, отделяющих яркости различных классов друг от друга.

Центральным вопросом пороговой сегментации является определение порогов, которое должно выполняться автоматически. Предложено огромное число методов вычисления порогов, в основном они базируются на анализе гистограммы исходного изображения.

Пусть $w(x)$, $0 \leq x \leq 255$, – гистограмма полутонового изображения. Диапазон яркостей имеет диапазон от 0 до 255. Идея гистограммного метода определения порога основывается на предположении о том, что распределения вероятностей для каждого класса унимодальны (т. е. содержат по одному доминирующему пику), а граничных пикселей, разделяющих участки разных классов на изображении, гораздо меньше, чем пикселей, 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 входящих в них точек граничным областям. Пороги сегментации находятся при этом по положению впадин. Рис. 13.1 иллюстрирует сказанное выше применительно к сегментации на два класса.

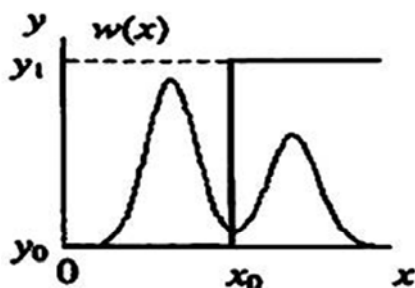


Рис. 13.1. Определение порога по гистограмме при сегментации по яркости на два класса

На практике воспользоваться такими простыми соображениями для выбора порога удастся крайне редко, так как реальные гистограммы обычно имеют множество мелких пиков и сильно изрезаны. В таком случае применяется сглаживание гистограммы как одномерного массива данных любым низкочастотным фильтром, описываемым ниже. Вторым препятствием является то, что границы между однородными участками на изображении бывают размыты, вслед-

ствии 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. 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.

13.1.2. 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 , $j = 1, \dots, k$ (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 m_j по всем кластерам.

Шаг 4. Повторять шаги 2 и 3, пока центры кластеров не перестанут изменяться.

13.1.3. Сегментация по принципу Вебера

Опишем алгоритм сегментации полутоновых изображений, основанный на принципе Вебера. Принцип Вебера подразумевает, что различия уровней серого $W(I)$, которые могут быть замечены человеческим глазом, – это нелинейные функции градаций серого I . простейшей функцией Вебера является

$$W(I) = \begin{cases} 20 - \frac{12I}{88}, & 0 \leq I \leq 88; \\ 0.002(I - 88)^2 + 8, & 88 \leq I < 138; \\ \frac{7(I - 138)}{255 - 138} + 13, & 138 \leq I \leq 255. \end{cases} \quad (13.2)$$

В соответствии с принципом 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.

По принципу Вебера, которому удовлетворяют характеристики, различимые человеческим глазом, **алгоритм нелинейного преобразования уровней серого** для полутоновых изображений будет выглядеть следующим образом.

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 пикселей, 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 $W(I(n))$.

Этот способ сегментации можно выполнить, построив LUT, вычисляя новые яркости с помощью функции Вебера, заменив исходные значения яркости найденными номерами классов одновременно.

При сегментации цветных изображений можно использовать статистические характеристики в окрестности пикселя: например, отклонение каждой из цветовых компонент от среднего значения по окрестности, дисперсию каждой цветовой компоненты в окрестности пикселя и т. д.

Существуют известные рекурсивные алгоритмы для простого случая разбиения по яркости – так называемый алгоритм наращивания областей (Region Growing).

13.1.4. Алгоритм сегментации Вежневца

Вежневц разработал следующий **итеративный алгоритм сегментации**.

Шаг 1. Начать обход изображения из левого верхнего угла. Объявить левый верхний пиксель изображения новым классом C_1 .

Для пикселей 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} . Принадлежность к классу можно отметить номером 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 классов больше порога, то завести новый класс; если отклонение больше лишь для одного класса, то добавить пиксель к тому классу, отклонение от которого меньше порога.

Если отклонение допустимо для обоих классов, возможны два варианта:

1) $L(g(C_i) - g(C_j)) < \delta$ – объединить эти два класса (если это не один и тот же класс) и к объединенному классу добавить текущий пиксель;

2) $L(g(C_i) - g(C_j)) > \delta$ – 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).

Отметим, что алгоритм использует 4-связность пикселей, но возможна реализация и для 8-связных областей. Результат сегментации представлен на рис. 13.2.

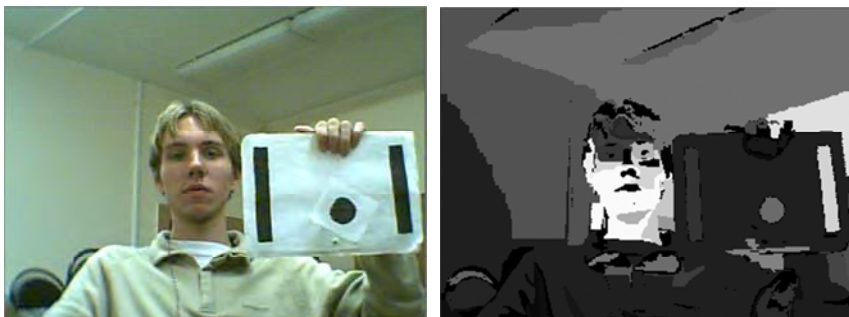


Рис. 13.2. Сегментация изображения с $\delta = 50$

13.1.5. 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 пикселей вычислительно эффективен. В литературе описано много подходов к сегментации цветных изображений с целью выделения на фотографиях людей таких областей, которые соответствуют цвету кожи. Далее в найденных областях выполняется детекция лица человека и его распознавание.

Задача

Выполнить сегментацию изображения по цвету кожи. Решение 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 подходов к решению этой задачи. Их суть заключается в описании некоторой модели распределения цвета кожи, с помощью которой и оценивается принадлежность цвета коже. Разработка модели цвета кожи разбивается на три этапа:

накопление тренировочных данных с использованием изображений, на которых указывают область «кожи» и «не-кожи». По этим данным накапливается статистика оттенков кожи (иногда гистограмма оттенков, к коже не относящихся, т. е. «не-кожи»);

обработка полученной статистики и выбор параметров модели цвета кожи для последующего использования. Выбор критериев оценки принадлежности пикселей к области кожи;

обработка изображений с использованием полученных критериев.

Простейшими (но не всегда дающими корректные результаты) вариантами 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 кожи, если выполнены следующие условия:

$$R > 95 \text{ и } G > 40 \text{ и } B < 20 \text{ и}$$

$$\max\{R, G, B\} - \min\{R, G, B\} > 15 \text{ и } |R - G| > 15 \text{ и} \quad (13.3)$$

$$R > G \text{ и } R > B$$

или

$$R > 220 \text{ и } G > 210 \text{ и } B > 170 \text{ и}$$

$$|R - G| \leq 15 \text{ и } R > B \text{ и } G > B. \quad (13.4)$$

Другой вариант:

$$\left(\frac{r}{g} > 1,185 \right) \text{ и } \left(\frac{rb}{(r+g+b)^2} > 0,107 \right) \text{ и } \left(\frac{rg}{(r+g+b)^2} > 0,112 \right), \quad (13.5)$$

где

$$r = R / (R + G + B);$$

$$g = G / (R + G + B);$$

$$b = B / (R + G + B).$$

После такой сегментации необходимо выполнить морфологическую или медианную фильтрацию. Пример сегментации представлен на рис. 13.3.



Рис. 13.3. Пример сегментации по цвету кожи

Преимущества таких подходов: простота реализации, не требуется обучение. Недостатки: цветовое пространство и набор правил необходимо подбирать эмпирически; часто случается неточная сегментация кожи; трудно подобрать единые критерии к людям разных рас.

13.2. Статистический подход к текстурной сегментации

В цифровой обработке изображений для описания текстуры применяются два основных подхода: статистический и структурный.

Статистические методы позволяют охарактеризовать текстуру области как гладкую, грубую, зернистую.

Структурные методы определяют и описывают взаимное расположение простейших повторяющихся элементов изображения: например, отрезки параллельных линий, проходящих с постоянным шагом, шахматная доска. Рассмотрим статистический подход к сегментации изображения на примере изображения, содержащего два типа областей, представленных разными текстурами (рис. 13.4). Результат сегментации – разбиение изображения на два типа областей: водную поверхность и сушу. Это невозможно сделать методами бинаризации, но только путем анализа параметров текстуры в окрестности каждого пикселя.

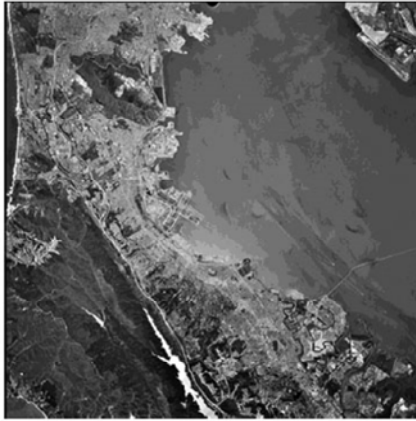


Рис. 13.4. Пример изображения с областями текстуры разных типов, соответствующих суше и воде

Алгоритм сегментации посредством текстурного анализа

Шаг 1. Считать изображение.

Шаг 2. Определить параметры текстуры на изображении. Будем считать, что яркости 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_i :

$$\mu_n(z) = \sum_{i=0}^{L-1} (z_i - m)^n p(z_i), \quad (13.6)$$

где m – среднее значение z (средняя яркость изображения):

$$m = \sum_{i=0}^{L-1} z_i p(z_i). \quad (13.7)$$

Из формулы (13.6) 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:

$$R = 1 - \frac{1}{1 + \sigma^2(z)} \quad (13.8)$$

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

$$s = \sigma(z) \quad (13.9)$$

также часто используется в качестве характеристики текстуры. Третий момент является характеристикой симметрии гистограммы:

$$\mu_3(z) = \sum_{i=0}^{L-1} (z_i - m)^3 p(z_i). \quad (13.10)$$

Для оценки текстурных особенностей используется функция энтропии, определяющая разброс по яркости соседних пикселей:

$$e = - \sum_{i=0}^{L-1} p(z) \log_2 p(z_i), \quad (13.11)$$

где e – значение энтропии в текущей точке;

$p(z_i)$ – вероятность текущей яркости в окрестности точки;

L – количество уровней яркости

Для описания текстуры также используется мера однородности, оценивающая равномерность гистограммы:

$$U = \sum_{i=0}^{L-1} p^2(z_i). \quad (13.12)$$

В табл. 13.1 приведены значения описанных выше характеристик для текстур трех видов: гладкой, грубой и периодичной.

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. На изображении имеются текстуры двух типов: крупной и более мелкой, более зернистой. Вторая соответствует водной зоне. Для отделения одной области от другой создать маску, которая удаляет маленькие объекты. Для этого использовать функцию определения связного множества пикселей 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, меньшей заданной величины S , удалить.

Результат текстурной фильтрации на базе вычисления энтропии по формуле (13.11) в окне 9×9 представлен на рис. 13.5.

Маска водной поверхности после удаления объектов малой площади и результат сегментации суши представлен на рис. 13.6.

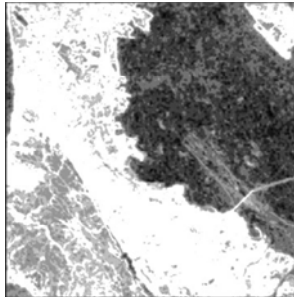


Рис. 13.5. Результат текстурной фильтрации на базе вычисления энтропии по формуле (13.6) в окне 9×9 . Области двух текстур показаны темными (вода) и светлыми (суша) оттенками

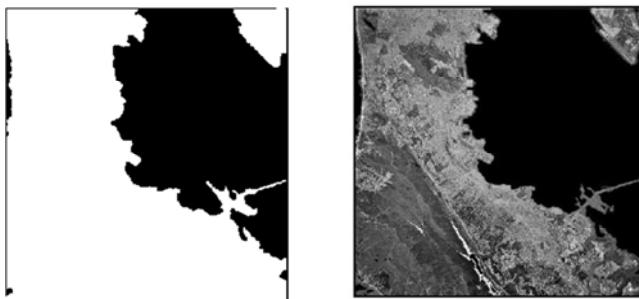


Рис. 13.6. Маска водной поверхности после удаления объектов малой площади и результат сегментации суши

Для сегментации двух указанных типов текстур можно использовать разные текстурные признаки: например, пары признаков, вычисляемых по формулам (13.2)–(13.4), (13.3)–(13.5), (13.2)–(13.6), (13.4)–(13.7), (13.6)–(13.7), (13.5)–(13.4), (13.2)–(13.3), (13.3)–(13.7), (13.4)–(13.6), и др.

13.3. Сегментация методом морфологического водораздела

В этом методе полутоновое изображение рассматривается как цифровая модель местности, где значения яркости представляют собой значения высот 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 по высотам уровня воды. Места объединения бассейнов отмечаются как линии водораздела. В итоге вся местность может быть затоплена. Результат сегментации зависит от «прекращения дождя». Если процесс остановить рано, изображение будет сегментировано на мелкие области, если поздно – на очень крупные.

В такой интерпретации все пиксели делятся на три типа:

- а) локальные минимумы;
- б) находящиеся на склоне, т. е. с которых вода скатывается в один и тот же локальный минимум;
- в) локальные максимумы, т. е. с которых вода скатывается более чем в один минимум.

При сегментации с помощью этого метода на изображении нужно определить «водосборные бассейны» и «линии водораздела» пу-

тем обработки локальных областей в зависимости от их яркостных характеристик.

При реализации метода водораздела выполняются следующие основные шаги.

1. Вычисляется функция сегментации. Она касается изображений, где объекты размещены в темных областях и являются трудно-различимыми.

2. Вычисляются маркеры переднего плана изображений. Они вычисляются на основании анализа связности пикселей каждого объекта.

3. Вычисляются фоновые маркеры. Они представляют собой пиксели, которые не являются частями объектов.

4. На основании значений расположения маркеров фона и маркеров переднего плана модифицируется функция сегментации.

Одним из важнейших этапов 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 часто применяется не к самому изображению, а к его градиентному представлению.

Выводы по разделу II

Общей теории обработки изображений не существует. Какой метод использовать при решении конкретной практической задачи, зависит от знаний и опыта исполнителя. Если задача многоцветная, т. е. регулярно повторяется последовательность действий, меняются только исходные данные, наработываются определенные алгоритмы обработки и параметры. Примером может служить ретушь фотографий для документов – пользовательский вариант обработки изображений. Для него достаточно освоить один – два пакета обработки и следить за обновлениями, осваивая новые возможности.

Если исходные изображения относятся к разным классам, классы периодически меняются, – это исследовательский тип обработки изображений. В этом случае разносторонние знания методов обработки подскажут ее правильную технологию.

В обоих случаях знание основных методов, их суть и параметры ускоряют исполнителю путь к получению желаемого результата.

Раздел III. ПРИМЕРЫ ПРИМЕНЕНИЯ ЦИФРОВОЙ ОБРАБОТКИ ИЗОБРАЖЕНИЙ

Глава 14. РЕШЕНИЕ ПРИКЛАДНЫХ ЗАДАЧ

Как правило, при решении реальных задач обработки изображений недостаточно применения одного метода. Требуется последовательное, а нередко итерационное, применение нескольких методов. Возникает проблема выбора методов и их конкретных параметров. Рассмотрим примеры решения реальных задач.

14.1. Повышение контраста рентгеновских изображений

Задача

Рентгеновские изображения часто имеют слабый контраст, плохую проработку деталей и не всегда можно сделать повторный снимок. Для улучшения таких снимков возникает необходимость цифровой обработки. Например, исходное изображение (рис. 14.1) низкоконтрастное, что затрудняет анализ мелких деталей.

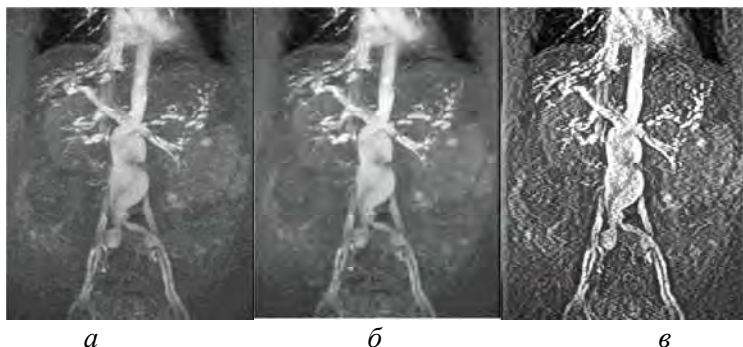


Рис. 14.1. Повышение контраста рентгеновского изображения

Решение

Для улучшения снимка можно сначала применить линейное растяжение гистограммы исходного изображения на максимальный диапазон $[0, \dots, 255]$ по формуле (9.12) (рис. 14.1, б).

Затем выполнить локальное контрастирование изображения.

Шаг 1. Выполнить усредняющую фильтрацию изображения L с ядром фильтра 3×3 . Обозначим 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 $L(i, j) > L_{yc}(i, j)$,

$$L_{рез}(i, j) = L_{yc}(i, j) \cdot (1 + C(i, j)) / (1 - C(i, j)),$$

иначе

$$L_{рез}(i, j) = L_{yc}(i, j) \cdot (1 - C(i, j)) / (1 + C(i, j)).$$

Результирующее изображение $L_{рез}$ визуально контрастнее исходного рентгеновского изображения (рис. 14.1, в).

14.2. Морфологическая фильтрация бинарных изображений

Морфологические операции можно использовать для построения фильтров, по принципу работы похожих на пространственные фильтры. На рис. 14.2 показано бинарное изображение фрагмента отпечатка пальца, искаженное шумом, в виде светлых точек на темном фоне и темных – на светлых полосах, составляющих отпечаток. Задача состоит в устранении шума при минимальном искажении формы отпечатка. Для решения этой задачи можно применить морфологический фильтр, вначале выполняющий операцию размыкания, затем – замыкания. На рис. 14.2, б представлен примитив, остальные части рисунка иллюстрируют последовательность шагов операции фильтрации. Шум в фоновой области изображения полностью устраняется на этапе эрозии (рис. 14.2, в), входящем в состав операции размыкания, поскольку в данном примере физические размеры всех составляющих шума меньше размеров примитива. Однако шумовые компоненты в виде темных пятен на отпечатке увеличились в размерах. Такое расширение нейтрализуется путем дилатации, что приводит к результату, показанному на рис. 14.2, г. Видно, что шумовые составляющие на полосах отпечатка пальца

уменьшились в размерах или полностью исчезли. Две описанные выше операции вместе (эрозия и дилатация) составляют операцию размыкания исходного изображения по примитиву, представленному на рис. 14.2, б. Из рис. 14.2, г видно, что результирующий эффект от размыкания состоит в устранении 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 к результату размыкания применить операцию дилатации, как показано на рис. 14.2, д. В большинстве указанных промежутков целостность полос восстановилась, однако полосы стали шире. Такое увеличение можно скомпенсировать эрозией.

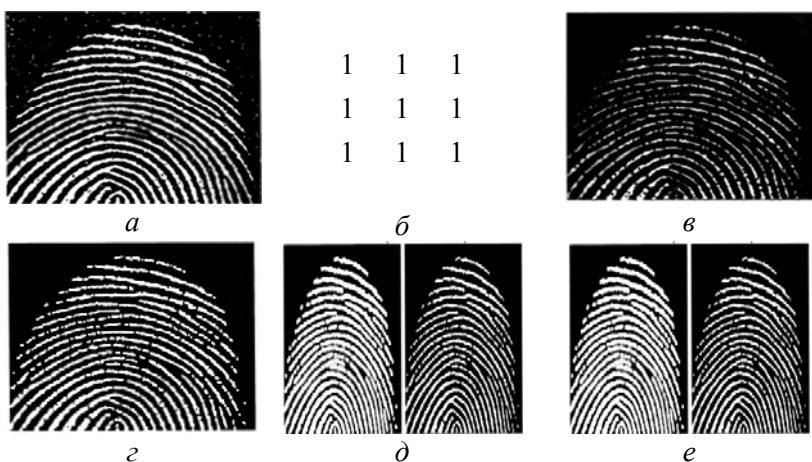


Рис. 14.2. Повышение качества отпечатка пальца (а) путем морфологической фильтрации: б – примитив; в – изображение после применения операции эрозии; г – результат размыкания исходного изображения; д – дилатация результата размыкания; е – замыкание результата размыкания

Результат, представленный на рис. 14.2, е, получен путем применения операции замыкания к ранее построенному размыканию исходного изображения, представленному на рис. 14.2, г. В результате отсутствуют шумовые точки, хотя некоторые из полос отпечатка пальца восстановлены не полностью, содержат разрывы. Этого можно было ожидать, поскольку в описываемую процедуру не было заложено никаких условий для сохранения связности.

14.3. Коррекция фона при неравномерном освещении сцены при анализе изображений

Задача

Дано изображение сцены с неравномерным освещением (рис. 14.3), автоматический анализ объектов, представленных на изображении, затруднен. Требуется выполнить коррекцию фона, сделав его однородным.

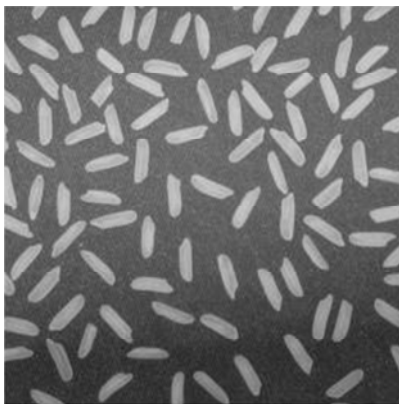


Рис. 14.3. Изображение с неравномерным фоном при неравномерном освещении сцены при регистрации

Алгоритм возможного решения

Шаг 1. Загрузить изображение.

Шаг 2. Применить морфологическую операцию отмыкания для оценки фона (рис. 14.4).

Шаг 3. Попиксельно вычесть яркость фона из исходного изображения. Выполнить линейное растяжение контраста полученного изображения (рис. 14.5).

Шаг 4. Выполнить бинаризацию изображения методом Отсу (см. рис. 14.5).

Шаг 5. Выполнить выделение объектов как связанных компонентов на бинарном изображении и отобразить их (рис. 14.6).

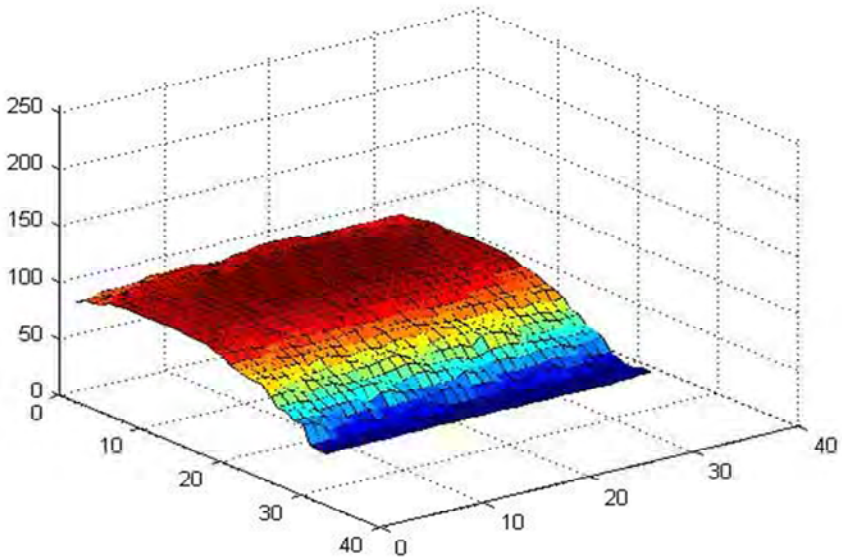


Рис. 14.4. Яркость фона предыдущего изображения, полученная после операции отмыкания с плоским структурным элементом радиуса 15

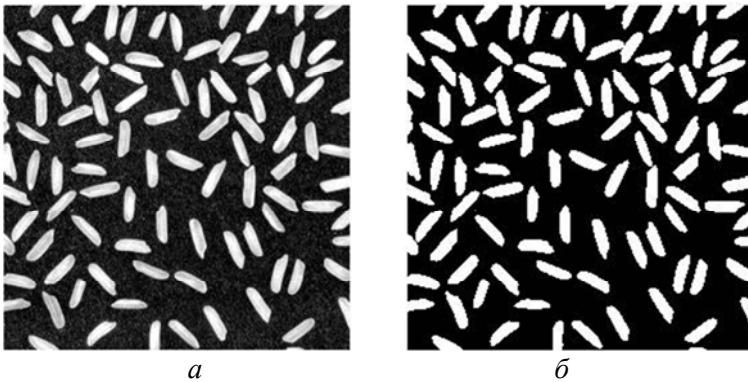


Рис. 14.5. Изображение после выполнения шага 3 (а) и шага 4 (б)



Рис. 14.6. Выделенные объекты

Шаг 6. Вычислить характеристики выделенных объектов.

14.4. Повышение разрешения мультиспектрального изображения

Задача

Дано: мультиспектральные спутниковые изображения в диапазонах B, G, R, NIR (ближний инфракрасный) и Pan (панхроматический), имеющий разрешение выше предыдущих в два или четыре раза. Требуется повысить разрешение мультиспектральных изображений до уровня панхроматического и подготовить их к отображению в виде цветного изображения. Такая процедура называется слиянием с панхроматическим каналом (Pansharpening) и объединяет данные панхроматического изображения местности 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. 14.7). Такая процедура используется при отображении местности в Google.maps. Отметим, что масштабирование исходных R, G, B изображений даст размытое изображение.



Рис. 14.7. Процедура Pansharpening снимков спутника «Иконос»

Решение

Кратко опишем идеи, использованные в простых методах решения данной задачи.

Самым простым является *алгоритм Бровей*.

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:

$$R_{\text{out}}(i, j) = R(i, j)\text{Pan}(i, j) / (B(i, j) + G(i, j) + R(i, j));$$

$$G_{\text{out}}(i, j) = G(i, j)\text{Pan}(i, j) / (B(i, j) + G(i, j) + R(i, j));$$

$$B_{\text{out}}(i, j) = B(i, j)\text{Pan}(i, j) / (B(i, j) + G(i, j) + R(i, j)).$$

Шаг 3. Полученные спектральные изображения R_{out} , G_{out} , B_{out} объединить в виде цветного изображения RGB.

Этот алгоритм быстр, но искажает цветовые характеристики.

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 пиксела каждого канала приобретают вид:

$$DNF(i, j) = (\text{Pan}(i, j) - NIR_w NIR(i, j)) / (R_w R(i, j) + G_w G(i, j) + B_w B(i, j)),$$

$$R_{\text{out}}(i, j) = R(i, j) DNF(i, j);$$

$$G_{\text{out}}(i, j) = G(i, j) DNF(i, j);$$

$$B_{\text{out}}(i, j) = B(i, j) DNF(i, j);$$

$$I_{\text{out}}(i, j) = I(i, j) DNF(i, j),$$

где Pan – панхроматический;

R – красный;

G – зеленый;

B – синий;

NIR – ближний инфракрасный каналы;

NIR_w, R_w, G_w, B_w – константы для разных каналов.

Еще один вариант **алгоритма операции** Pansharpening – преобразование из RGB в другое цветовое пространство, например в IHS. В нем корректируется яркостная компонента.

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.

Шаг 3. Построить гистограммы изображений Pan и компоненты I . Преобразовать форму гистограммы I в форму гистограммы Pan (см. п. 9.5) и получить новое представление яркостной компоненты I_{out} .

Шаг 4. В IHS заменить яркостную компоненту на I_{new} и выполнить преобразование из представления $I_{\text{out}} HS$ в $R_{\text{out}} G_{\text{out}} B_{\text{out}}$.

Последний шаг всех алгоритмов: выполнить переквантование значений яркости полученного цветного изображения в 8-битовое представление (см. п. 2.2.2).

14.5. Детекция дорожных знаков

Задача

Автоматически найти на видео предупреждающие дорожные знаки (рис. 14.8); скорость обработки кадров (не менее 10 кадров/с).



Рис. 14.8. Предупреждающие дорожные знаки

Для этого необходимо решить следующие задачи:
преобразование цветового пространства;
удаление шумов;
выделение областей интереса;
распознавание типа знака.

Шаг 1. Преобразование цветового пространства.

Все предупреждающие дорожные знаки имеют общий признак – контур красного цвета (см. рис. 14.8). Соответственно первоначальный поиск объектов интереса заключается в определении областей красного цвета.

Входное изображение цветное, представлено в цветовом пространстве RGB. Для детекции знаков по цвету целесообразнее использовать другое цветовое пространство, например YC_bC_r . Для конвертации из RGB в YC_bC_r используется уравнение

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 65,481 & 128,553 & 24,966 \\ -37,797 & -74,203 & 112 \\ 112 & -93,786 & -18,214 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \quad (14.1)$$

Для сегментации изображения по красному цвету можно использовать C_r -составляющую, но она содержит шумы.

Шаг 2. Удалить шумы на изображении. Для этого требуется выполнить фильтрацию изображения. Фильтрацию можно выполнять

разными способами: а) на полутоновом представлении C_b или б) на бинарном.

Для получения бинарного изображения выполняется операция бинаризации изображения C_r (рис. 14.9).



Рис. 14.9. Бинарное представление C_r -составляющей реального изображения, содержащего знак

Однако для удаления мелких объектов (рис. 14.10, а) в обоих случаях требуется использовать морфологические фильтры «размыкание» и «замыкание».



Рис. 14.10. Изображение 14.9 после медианной фильтрации: б – с морфологической фильтрацией, а – без

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 доски). Вторым признаком класса рассматриваемых дорожных знаков, который можно использовать, – форма правильного треугольника.

Для определения формы необходимо провести анализ 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 выбираются несколько граничных точек, которые аппроксимируются тремя прямыми (рис. 14.11).

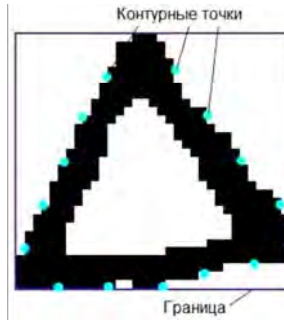


Рис. 14.11. Изображение

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

$$A = y_2 - y_1;$$

$$B = x_1 - x_2 \tag{14.2}$$

$$\cos \varphi = \frac{|A_1 A_2 + B_1 B_2|}{\sqrt{A_1^2 + B_1^2} \sqrt{A_2^2 + B_2^2}}.$$

Объект считается распознанным как знак, если

$$60^\circ - \Delta\varphi \leq \varphi \leq 60^\circ + \Delta\varphi,$$

где φ – угол между прямыми,

$\Delta\varphi$ – допустимое отклонение угла.

Далее выполняется распознавание конкретного знака путем анализа его внутреннего содержания.

14.6. 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 состоит в использовании спектральных индексов на основе соотношения яркости изображений, зарегистрированных в синем (380–500 нм) и красном (600–760 нм) диапазонах электромагнитного спектра. Идея состоит в детекции процесса флуоресценции поверхности пятен нефти и увеличенного отражения в красном диапазоне находящихся под ними слоев нефти.

Индекс флуоресценции FI вычисляется для всех пикселей мультиспектральных изображений синего и красного диапазонов спектра по формуле (это формула вычисления контраста по двум изображениям)

$$FI = (B - R) / (B + R). \quad (14.3)$$

Нефтяные пятна на воде меняют оптические свойства в зависимости от угла падения солнечных лучей и геометрии регистрации спутниковыми датчиками. Нефтяные пятна практически не видны на безбликовых изображениях, но могут быть заметны в виде довольно темных пятен (из-за отрицательного контраста) при низких или умеренных солнечных бликах либо при довольно ярких пятнах (из-за положительного контраста) при сильных солнечных бликах.

14.7. 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.

Для получения цифрового изображения поперечной балки под транспортировочной линией были помещены осветители, а над линией была расположена видеокамера. На изображении темные пиксели в области тени от балки имеют значения 1 и обозначают сталь, а пиксели в области светлых просверленных отверстий имеют значение 0. Количество отверстий вычисляется как четверть разности количества внешних и внутренних углов. На рис. 14.12 показаны три светлых отверстия (значения 0) на темном фоне (значения 1). Внешний угол (external corner) – это окрестность 2×2 соседних пикселей, содержащая три единицы, а внутренний угол (internal corner) – окрестность 2×2 соседних пикселей, содержащая три нулевых значения. На рис. 14.12 показан пример обработки изображения из 7 строк и 16 столбцов. Ниже также схематично описан алгоритм обработки.

$$1. \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

$$2. \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$$



4.

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

5.

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

Рис. 14.12. Подсчет отверстий на бинарном изображении:

1. *a* – шаблоны окрестностей 2×2 пикселей, представляющие «внешние углы»;
2. *b* – шаблоны окрестностей 2×2 пикселей, представляющие «внутренние углы»;
3. *v* – три светлых отверстия на темном фоне;
4. *z* – входное бинарное изображение с размерами 7 строк и 16 столбцов;
5. *d* – символами «e» и «i» отмечены окрестности, соответствующие внешним и внутренним углам

Схема алгоритма

Входные данные – бинарное изображение, выходные данные – количество найденных отверстий.

Обозначения:

M – бинарное изображение из R строк и C столбцов;

значения «1» представляют материал, не пропускающий свет;

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

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

E – счетчик внешних углов (три единицы и один нуль в окрестности 2×2);

I – счетчик внутренних углов (три нуля и одна единица в окрестности 2×2).

```
integer procedure Count_Holes (M)
```

```
{
```

```
  обработать все изображение по две строки на каждом шаге;
```

```
  подсчитать количество внешних углов  $E$ ;
```

```
  подсчитать количество внутренних углов  $I$ ;
```

```
  вернуть в качестве результата ( $\text{number\_of\_holes} = (E - I) / 4$ );
```

```
}
```

14.8. Улучшение изображений в системах видеонаблюдения

Изображения, которые регистрируются системой видеонаблюдения, не всегда качественны, поскольку системы регистрируют изображения в различных погодных условиях, при разном освещении. Рассмотрим, как в системах видеонаблюдения можно улучшить низкокачественные изображения.

14.8.1. Усиление контрастности

Низкий контраст является одним из наиболее распространенных дефектов фотографических и телевизионных изображений. Причины могут быть самые различные: 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. 14.13).

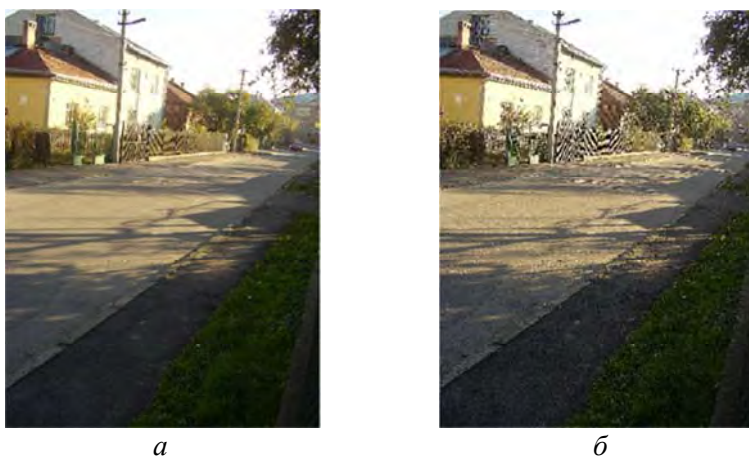


Рис. 14.13. Исходное (*a*) и обработанное (*б*) изображения

14.8.2. Коррекция динамического диапазона яркости

При недостаточном освещении регистрируемые изображения имеют узкий динамический диапазон значений. Поэтому коррекция диапазона в системах видеонаблюдения очень актуальна.

Независимо выполним нелинейное растяжение гистограммы для трех цветовых компонент по формуле

$$K_{\text{stretch}} = ((K - K_{\min}) / (K_{\max} - K_{\min}))^a. \quad (14.4)$$

Затем выполним медианную фильтрацию. Фильтрация необходима, поскольку после контрастирования темных изображений они имеют зернистый вид. Результат фильтрации и коррекции динамического диапазона представлен на рис. 14.14–14.15.

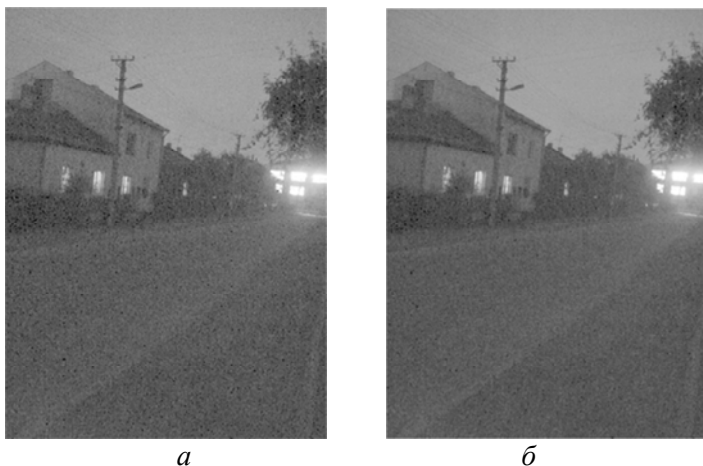


Рис. 14.14. Исходное изображение после применения операции: *a* – контрастирования; *б* – фильтрации



Рис. 14.15. Исходное изображение (*a*) и результат контрастирования (*б*)

Глава 15. ПРОГРАММНЫЕ СРЕДСТВА ОБРАБОТКИ ЦИФРОВЫХ ИЗОБРАЖЕНИЙ

В данной главе не рассматриваются пакеты обработки изображений типа Adobe Photoshop «Фотошоп». Назначение этого пакета – интерактивное редактирование изображений. Его коды закрыты, и вся обработка изображений выполняется нажатием комбинации клавиш и мыши без описания алгоритмов преобразования и обработки изображений.

15.1. Писать код или взять из готовой библиотеки?

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 вспомогательной инфраструктуре. Однако библиотеки обычно пишутся для универсального применения, решения множества различных задач. Часто их код довольно громоздок для понимания и не всегда оптимален для решения конкретной задачи. Если разработчик реализует собственное решение, то напишет его оптимальным (согласно опыту) способом и будет конкретизировать его именно под свою задачу.

При использовании библиотеки имеется решение, оттестированное множеством пользователей. Если же все-таки будет найдена ошибка, то в интернете можно найти советы людей, сталкивавшихся с ней. Однако, если с этой ошибкой никто не сталкивался или о ней никто не пишет, можно потратить много времени, чтобы разобраться, почему что-то не работает, ведь код библиотеки может быть довольно громоздким и не всегда доступным.

Если разработчик писал весь код сам, то он хорошо в нем ориентируется и найдет ошибку просто. Но ошибок, скорее всего, будет значительно больше, так как тестировал этот код только разработчик. Если он пишет код с нуля, то есть вероятность, что из вида будет упущен какой-нибудь редкий случай, который в библиотеке со временем же выявили и закодировали.

Если работает в команду разработчика приходит новый работник, есть вероятность, что он уже работал с методами, реализованными в библиотеке (либо его берут с соответствующим опытом). Если же разработчик использует свое собственное решение, то новому сотруднику придется разбираться в деталях его работы. С другой стороны, написать самому код, который уже реализован в библиотеках, весьма непросто (особенно для начинающих). Для этого нужен высококвалифицированный специалист, иначе все остальные преимущества собственного решения сведутся на нет.

Резюмируем достоинства и недостатки каждого подхода.

Достоинства:

- быстрая разработка проекта;
- код оттестирован многими пользователями;
- новичкам можно объяснить, как работает программа;
- решает проблемы, которые могли не предвидеть, реализуя функцию библиотеки самостоятельно;
- программист концентрируется на основной задаче, а не на вспомогательных;
- разработка собственной библиотеки стоит большего количества человека-часов, чем готовая.

Недостатки:

- обычно программные компоненты продаются с лицензиями на каждый компьютер и стоимость всех лицензий может превысить стоимость собственной разработки;
- очень много классов, сложная общая архитектура, тяжело разбираться в возникающих ошибках;
- программный продукт может не содержать специфичную функцию, которая нужна для вашей задачи;
- нет возможности решить данную задачу оптимальным способом, требуется загрузка множества ненужных для данного проекта функций;
- могут возникнуть проблемы с обновлением библиотеки в будущем;
- чаще всего для проекта требуется несколько библиотек, из каждой используется только часть функции;
- к чужой библиотеке тоже надо писать код, разбираться в документации.

Достоинства при использовании собственной разработки:

члены команды знают, как работает программа;

проще искать причины ошибок, так как написанный код минимален;

максимально производительное решение;

возможность развития библиотеки в нужном направлении.

Недостатки:

требуется высококвалифицированный разработчик для написания хорошей библиотеки;

длительность разработки;

содержится большое количество ошибок, особенно на стадии внедрения;

новички не знают, как работает программа.

Не стоит писать программы:

если разработчик не обладает необходимой квалификацией (как программист или конкретно в этой области) для качественной реализации этой библиотеки;

речь идет об одной из стандартных функций, которые используются в библиотеках как базовые;

существующая библиотека написана просто и ясно;

нет уверенности, что данное решение будет значительно проще.

Если задача, решаемая библиотекой, действительно сложна, то реализация будет лучше и проще и на ее написание и отладку уйдет меньше времени и сил, чем на обнаружение и исправление ошибок в чужой библиотеке. Кроме того, нередко эту работу удастся переложить на автора библиотеки.

Можно сделать следующий вывод: если писать код, то очень много сил и времени уйдет на поиск решения задач, которые уже давно решены, причем оптимальным образом.

Создавать свою библиотеку нужно:

если то, что нужно, быстрее сделать, чем найти готовое;

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.

При самостоятельном программировании ключевой фактор – длительная жизнь проекта. С запуском проекта он не заканчивается, а только начинается, т. е. 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: возможна высокая стоимость проекта. Этот риск компенсируется тем, что такой проект, будучи успешным, способен кормить предприятие многие годы.

Использование и анализ готовых решений позволяет начинающим быстрее повышать квалификацию и создавать несложные проекты. Рассмотрим два варианта библиотек, реализующих функции обработки изображений и имеющих открытые коды, MATLAB и OpenCV.

15.2. Система MATLAB

MATLAB – это сокращение от английского «Matrix Laboratory», матричная лаборатория.

MATLAB – это пакет прикладных программ для решения задач технических вычислений, множество специализированных библиотек и одноименный язык программирования. Язык MATLAB – интерпретатор. MATLAB работает на большинстве современных операционных систем.

Использование MATLAB ускоряет исследовательскую работу в несколько раз, позволяет сосредоточиться именно на проверке алгоритма, а не на его реализации. В MATLAB ошибки реализации отслеживаются легче.

В MATLAB можно создавать специальные библиотеки функций (toolbox), расширяющие его функциональность. Например, библиотеки Digital Signal Processing Toolbox, Image Processing Toolbox, Wavelet Toolbox, Filter Design Toolbox используются для цифровой обработки сигналов и изображений.

Будучи тесно связанным со средой разработки приложений MATLAB, пакет Image Processing Toolbox освобождает разработчика от поиска решений, кодирования и отладки алгоритмов, позволяя

сосредоточить усилия на решении основной практической задачи. MATLAB и пакет Image Processing максимально приспособлены для обучения, развития новых идей и их проверки.

Библиотека Image Processing Toolbox представляет собой набор основных функций обработки изображений:

- чтение и запись различных форматов изображений, видео и последовательностей 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;

 - 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 виде m-файлов. В них оптимально реализованы наиболее используемые алгоритмы обработки изображений. Коды этих функций открыты.

Отличительной чертой MATLAB является ориентация на матричные операции. Даже константа трактуется как матрица размером 1×1 . Такая ориентация очень удобна для обработки изображений.

Преимущества системы MATLAB:

- большое количество встроенных операций по работе с матрицами. Очень упрощает как различные задачи, связанные с решением систем линейных алгебраических уравнений и т. п., так и работу с изображениями;

- богатый набор постоянно развиваемых библиотек различных математических методов (численные методы, дифференциальные уравнения, обработка сигналов, вычислительная геометрия и т. д.);

- в Интернете доступно множество библиотек и функций для MATLAB;

 - многие функции описаны в книгах;

большие возможности по быстрой и простой (для разработчика) визуализации различных данных (двумерных и трехмерных графиков функций, множеств и т. д.);

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;

возможность использовать МАТЛАБ в С++ программах и наоборот (в МАТЛАБ можно использовать библиотеки, написанные на С++). Таким образом, удастся тестировать алгоритмы, написанные на С++, с помощью среды МАТЛАБ, и использовать в своих программах средства МАТЛАБ (например, визуализацию данных) через МАТЛАБ. Например, добавлять в свою программу вывод 3D-графи-ком (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. 15.1):

```
A = imread ('cameraman.tif'); % прочитать изображение в A;  
B = A/2; % уменьшить яркость всех пикселей в два раза;  
imshow (B); % отобразить результат обработки.
```

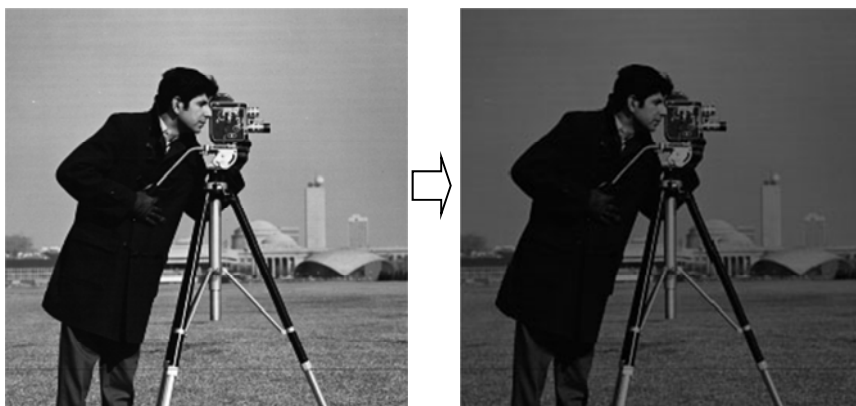


Рис. 15.1. Результат обработки изображения системой МАТЛАБ (код описан выше)

15.3. Открытая библиотека OpenCv

OpenCV (англ. *Open Source Computer Vision Library* – 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 C и C++, 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.

OpenCV содержит более 500 функций для различных областей приложения: контроль качества на производстве, системы безопасности, обработку изображений, пользовательские интерфейсы, стереоскопическое зрение и роботостроение, получение видео с камер, распознавание образов, включая обнаружение лиц, отслеживание движений в двух и трех измерениях, 3D-реконструкцию стереоскопического зрения и др. OpenCV также включает в себя библиотеку обучения машин, ориентированную на задачи статистического распознавания образов и кластеризации.

OpenCV используется во многих приложениях и научно-исследовательских работах. Основные модули OpenCV:

`opencv_core` – основная функциональность, включающая базовые структуры, вычисления (математические функции, генераторы случайных чисел) и линейную алгебру, DFT, DCT, ввод/вывод для XML и YAWL и т. д.;

`opencv_imgproc` – обработка изображений (фильтрация, геометрические преобразования, преобразование цветовых пространств и т. д.);

`opencv_highgui` – простой UI, ввод/вывод изображений и видео;

`opencv_ml` – статистические модели машинного обучения (SVM, деревья решений, обучение со стимулированием и т. д.);

`opencv_features2d` – распознавание и описание плоских примитивов (SURF, FAST и др.);

`opencv_video` – анализ движения и отслеживание объектов (оптический поток, шаблоны движения, устранение фона);

`opencv_objdetect` – обнаружение объектов на изображении (нахождение лиц алгоритмом Виолы – Джонса, распознавание людей HOG и т. д.);

opencv_calib3d – калибровка камеры, поиск стерео-соответствия и элементы обработки трехмерных данных;

opencv_flann – библиотека быстрого поиска ближайших соседей (FLANN 1.5) и обертки OpenCV;

opencv_gpu – ускорение некоторых функций OpenCV за счет CUDA, создан при поддержке NVidia.

Документацию по библиотеке можно посмотреть на сайте <http://docs.opencv.org/index.html>.

Выводы по разделу III

Решение конкретной прикладной задачи по обработке изображений зависит от опыта разработчика. К сожалению, при небольшом изменении условий получения одного и того же объекта могут существенно измениться его цифровое изображение и методы обработки этого изображения: например, изображение, полученное в красном диапазоне электромагнитного спектра и ближнем инфракрасном (см. рисунок).



а



б

Пример космических снимков одной местности в разных спектральных диапазонах: *а* – ближний инфракрасный спектр;

б – красный спектр

Часто возникает следующая ситуация: имеется одно или несколько изображений (очень часто с разными недостатками); требуется распознать представленный на них объект (например, номер автомобиля).

Обработка существенно меняющихся изображений разных типов – это в большей степени искусство, а не ремесло, но искусство, опирающееся на разнообразные знания. Знание методов и опыт поиск нужной последовательности в их применении облегчит решение поставленной задачи. Даже постоянно работая с узким классом изображений (например, ретушь фотопортретов), полезно знать основы названных методов.

ЗАКЛЮЧЕНИЕ

При написании авторы стремились достичь следующих целей:
показать, как формируются цифровые изображения разной природы;

описать основы обработки таких изображений;

показать, как с их помощью решаются конкретные прикладные задачи.

Более глубокие знания можно почерпнуть в других учебниках и специализированной литературе, а также в сети Интернет.

Получение новых типов изображений требует новых методов их обработки. Реализация новых теорий позволяет создавать новые устройства регистрации изображений. Этот процесс получения новых изображений и новой информации из них бесконечен, как путь монахов на гравюре М. Эшера «Спускаясь и поднимаясь», представленной на обложке в несколько искаженном дисторсией виде.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Ватолин, Д.С. Алгоритмы сжатия изображений : метод. пособие / Д.С. Ватолин. – Москва : Изд-во им. М. В. Ломоносова, 1999. – 157 с.
2. Гонсалес, Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс : пер. с англ. – Москва : Техносфера, 2005. – 1072 с.
3. Журавель, И.М. Краткий курс теории обработки изображений [Электронный ресурс] / И.М. Журавель. – Режим доступа: <http://матлаб.хронента.ru/imageprocess/book2/79.php>. – Дата доступа: 15.06.2015.
4. Методы компьютерной обработки изображений / под ред. В.А. Сойфера. – Москва : Физматлит, 2003. – 784 с.
5. Мюррей, Д.Д. Энциклопедия форматов графических файлов / Д.Д. Мюррей, У. ван Райнер : пер. с англ. – Киев : ВНУ, 1997. – 672 с.
6. OpenCV – библиотека компьютерного зрения с открытым исходным кодом: <http://opencv.ru>.
7. Приоров, А.Л. Цифровая обработка изображений : учеб. пособие / А.Л. Приоров, И.В. Апальков, В.В. Хрящев. – Ярославль : ЯрГУ, 2007. – 235 с.
8. Прэтт, У. Цифровая обработка изображений / У. Прэтт : пер. с англ. – Москва : Мир, 1982. – Кн. 2. – 480 с.
9. Старовойтов, В.В. Локальные геометрические методы цифровой обработки и анализа изображений / В.В. Старовойтов. – Минск : Ин-т техн. кибернетики, 1997. – 282 с.
10. Фисенко, В.Т. Компьютерная обработка и распознавание изображений : учеб. пособие / В.Т. Фисенко, Т.Ю. Фисенко. – Санкт-Петербург : СПбГУ ИТМО, 2008. – 192 с.
11. Цифровая обработка изображений в информационных системах : учеб. пособие / И.С. Грузман [и др.]. – Новосибирск : Изд-во НГТУ, 2002. – 352 с.
12. Шапиро, Л. Компьютерное зрение / Л. Шапиро, Дж. Стокман : пер. с англ. – Москва : БИНОМ, 2006. – 752 с.
13. Яне, Б. Цифровая обработка изображений/ Б. Яне : пер. с англ. – Москва : Техносфера, 2007. – 584 с.

Учебное издание

СТАРОВОЙТОВ Валерий Васильевич
ГОЛУБ Юлия Игоревна

**ПОЛУЧЕНИЕ И ОБРАБОТКА
ИЗОБРАЖЕНИЙ НА ЭВМ**

Учебно-методическое пособие

Редакторы *Т.Н. Микулик, Т.В. Мейкшане*
Компьютерная верстка *Е.А. Беспанской*

Подписано в печать 22.05.2018. Формат 60×84 ¹/₁₆. Бумага офсетная. Ризография.
Усл. печ. л. 11,85. Уч.-изд. л. 9,27. Тираж 100. Заказ 993.

Издатель и полиграфическое исполнение: Белорусский национальный технический университет.
Свидетельство о государственной регистрации издателя, изготовителя, распространителя
печатных изданий № 1/173 от 12.02.2014. Пр. Независимости, 65. 220013, г. Минск.

Старовойтов Валерий Васильевич – доктор технических наук, главный научный сотрудник ОИПИ НАН Республики Беларусь, профессор кафедры ИСИТ МИДО БНТУ;

Голуб Юлия Игоревна – кандидат технических наук, старший научный сотрудник ОИПИ НАН Республики Беларусь, доцент кафедры ПОВТиАС ФИТР БНТУ