



В.В. Селянкин
С.В. Скороход

Анализ и обработка изображений в задачах компьютерного зрения



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное
учреждение высшего образования
«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Инженерно-технологическая академия

В.В. Селянкин
С.В. Скороход

Анализ и обработка изображений в задачах компьютерного зрения

Учебное пособие

*Для студентов магистерской подготовки
по направлению «Программная инженерия»*

Таганрог
Издательство Южного федерального университета
2015

Рецензенты:

профессор кафедры информатики Таганрогского института им. А.П. Чехова (филиал РИНХ), доктор технических наук **Витиска Н.И.**;

директор Центра разработки и сопровождения АС – филиала ООО «Технопром» в г. Таганроге, кандидат технических наук **Калачев Д.П.**

Селянкин В.В., Скороход С.В. Анализ и обработка изображений в задачах компьютерного зрения: учебное пособие. – Таганрог: Изд-во ЮФУ, 2015. – 82 с.

В пособии рассматриваются вопросы формирования и анализа изображений, обработки полутоновых и цветных изображений и сегментации изображений. Рассмотренные алгоритмы ориентированы на решение задач компьютерного зрения и распознавания образов. Рассчитано на магистрантов, аспирантов и специалистов, работающих в области практического использования компьютерной графики и компьютерного зрения.

Ил. 15. Библиогр.: 6 назв.

©ЮФУ, 2015

© Селянкин В.В., 2015

© Скороход С.В., 2015

Содержание

Введение	7
1. Средства и методы формирования изображений	12
1.1. Устройства для формирования изображений	12
1.1.1. Камеры на основе приборов с зарядовой связью	12
1.1.2. Видеокамеры	12
1.1.3. Микроденситометры	13
1.1.4. Датчики для цветных и мультиспектральных изображений	13
1.1.5. Датчики рентгеновского изображения	14
1.1.6. Изображения, полученные методом ЯМР	14
1.1.7. Дальнометрические изображения	15
1.1.8. Ультрасонографические изображения	15
1.2. Особенности формирования цифровых изображений	16
1.3. Системы координат для трехмерных изображений	17
1.3.1. Экранная система координат	17
1.3.2. Система координат объекта	17
1.3.3. Система координат камеры	17
1.3.4. Декартова система координат изображения	18
1.3.5. Мировая система координат	18
2. Анализ бинарных изображений	19
2.1. Обработка черно-белых изображений	19
2.2. Подсчет объектов на изображении	20
2.3. Маркировка связных компонент изображения	22
2.4. Морфология бинарных изображений	23
2.4.1. Структурирующие элементы	23
2.4.2. Основные операции бинарной морфологии	24
2.5. Свойства бинарных областей	25
2.5.1. Площадь области и центр тяжести	26
2.5.2. Длина периметра	26
2.5.3. Свойство округлости	27
2.5.4. Описывающий прямоугольник и экстремальные точки	28
2.5.5. Моменты второго порядка для областей	29
2.6. Графы смежности областей	31
2.7. Пороговая бинаризация изображений	32
3. Обработка полутоновых изображений	34
3.1. Увеличение диапазона полутонов	34
3.2. Удаление мелких деталей	36
3.2.1. Удаление шума «соль и перец»	36
3.2.2. Удаление мелких областей	37
3.2.3. Сглаживание изображений	37

3.2.4. Медианный фильтр	38
3.3. Обнаружение краев с помощью дифференциальных масок	38
3.3.1. Дифференцирование одномерных сигналов.....	38
3.3.2. Дифференцирование двумерных изображений	42
3.3.3. Обнаружение краев с помощью фильтрации	43
3.3.4. Обнаружение краев с помощью ЛОГ-фильтра.....	45
3.4. Использование масок для анализа изображения.....	46
3.4.1. Некоторые понятия векторов	46
3.4.2. Анализ на основе базиса Робертса	48
3.4.3. Анализ на основе базиса Фрея – Чена.....	49
4. Обработка цветных изображений	52
4.1. Физические свойства цвета	52
4.2. Цветовые модели.....	53
4.2.1. Цветовая модель RGB.....	53
4.2.2. Цветовая модель CMY.....	54
4.2.3. Цветовая модель HIS.....	54
4.2.4. Пространственные представления цветовых моделей.....	55
4.2.5. Цветовые системы YIQ и YUV.....	55
4.3. Гистограммы цветных изображений	56
4.4. Текстура	57
4.4.1. Понятие текстуры	57
4.4.2. Тексельное описание текстур	58
4.4.3. Количественные характеристики текстур	59
4.4.4. Текстурная сегментация	62
5. Сегментация изображений.....	63
5.1. Понятие задачи сегментации.....	63
5.2. Обнаружение областей	63
5.3. Методы кластеризации	64
5.3.1. Классические алгоритмы кластеризации.....	64
5.3.2. Итерационная кластеризация по математическому ожиданию.....	65
5.3.3. Алгоритм кластеризации ISODATA.....	65
5.3.4. Простые гистограммные методы	66
5.3.5. Рекурсивный гистограммный метод.....	67
5.3.6. Алгоритм графового разбиения Шу	67
5.4. Нарращивание областей	69
5.5. Способы представления областей	70
5.5.1. Оверлейные представления.....	70
5.5.2. Маркированные изображения	70
5.5.3. Кодирование границ.....	70

5.5.4. Квадратные деревья.....	71
5.5.5. Таблицы свойств.....	71
5.6. Обнаружение контуров.....	71
5.6.1. Прослеживание границ областей	72
5.6.2. Обнаружение краев Кэнни.....	72
5.6.3. Группировка контурных фрагментов.....	74
5.6.4. Обнаружение прямых и дуг окружностей.....	76
5.6.5. Обнаружение отрезков прямых.....	77
5.6.6. Обнаружение окружностей.....	79
5.7. Подбор моделей сегментов	80
5.7.1. Подбор прямой.....	80
5.7.2. Подбор других моделей сегментов.....	82
Заключение.....	82
Библиографический список.....	83

Введение

Термин «компьютерное зрение» в современной технической литературе соседствует с похожими понятиями: «техническое зрение» и «машинное зрение». Во многих случаях их можно воспринимать как синонимы, хотя возможны и отличия в некоторых практических приложениях. Например, «машинное зрение» может применяться в контексте промышленных приложений. Для этого случая в русскоязычной литературе в качестве синонима используется и термин «техническое зрение» [1]. В данном учебном пособии термины «компьютерное зрение» и «машинное зрение» считаются синонимами. Здесь можно провести параллель с понятиями компьютерная и машинная графика.

Кроме того, можно отметить, что компьютерное зрение является ответвлением компьютерной графики и ее развитием в прикладном использовании в различных областях обработки и анализа изображений. В качестве примеров задач, в решении которых используется компьютерное зрение, можно указать следующие и охранные системы, системы контроля качества изделий, распознавание объектов и текстовых документов, поиск изображений на основе содержания, медицинскую диагностику, получение и обработку спутниковых изображений, робототехнику, контроль на транспортных магистралях и другие. Здесь можно говорить о некоторых аспектах зрения человека, реализованных техническими средствами. Хотя качество решения этих задач несравнимо с человеком, но даже такой уровень решения задач является существенным прогрессом современных ИТ-технологий.

Известные три классические задачи в применении к графике формулируются следующим образом. Задача **синтеза** изображений предполагает получение изображения по его описанию. Задача **анализа** состоит в получении описания заданного изображения. Наконец, задача **обработки изображений** заключается в получении нового изображения на основании исходного с использованием нужного алгоритма. Компьютерное зрение, как правило, предполагает обработку и анализ изображений [2-4].

Последовательность действий для получения конечного результата решения задачи компьютерного зрения может иметь вид, показанный на рис. 1.

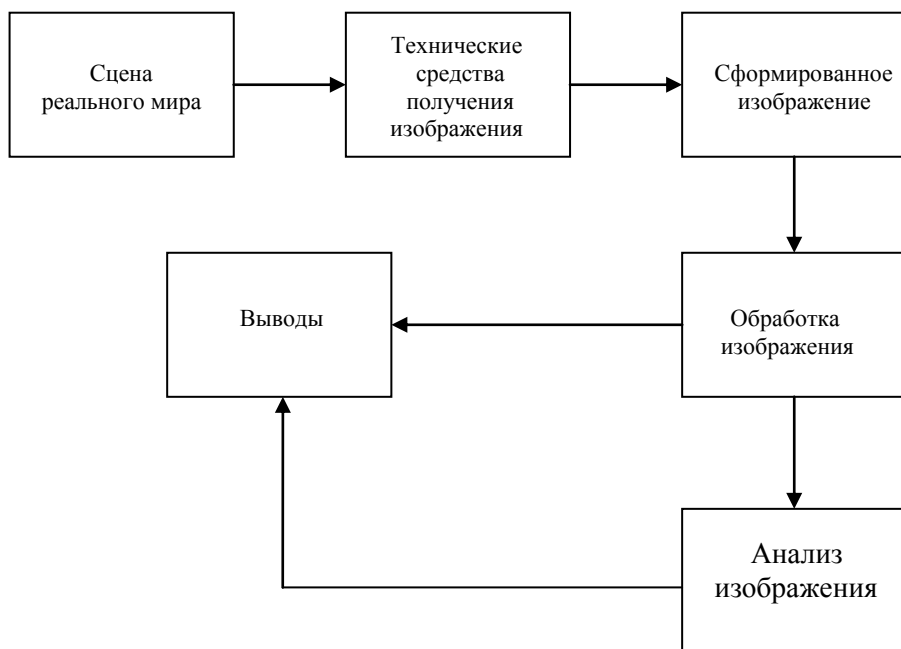


Рис. 1. Схема конвейера решения задачи компьютерного зрения

В зависимости от вида решаемой задачи конкретизируется наполнение каждого блока представленного конвейера. Ниже приведены примеры реализации такого конвейера для различных практических задач.

Задача поиска изображений на основе содержания. Имеется база данных изображений. Такими изображениями могут быть лица людей, различные объекты трехмерного мира, изображения автомобилей, товаров, различных изделий, зданий и т.п. Ставится задача поиска нужного изображения в базе данных. Причем задача может формулироваться как для поиска точного изображения, так и максимально приближенного к заданному. Возможно также выделение класса объектов, к которому принадлежит заданный объект. В этом случае в качестве сцены реального мира выступают соответствующие объекты и для получения базы данных изображений используются необходимые технические средства.

Важной задачей является подбор методов и алгоритмов обработки изображений, которые могут дать качественное решение задачи. Технологии обработки в этом случае зависят от многих параметров:

объема базы данных, размеров на изображении, качества изображений, параметров яркости и контрастности изображений, наличия фона, ракурсы расположения объектов и т.д.

В зависимости от требований задачи возможен широкий спектр требуемых выводов. В простейших случаях возможен двузначный ответ типа «да/нет», в более сложных – выбор нужного объекта на изображении или класса объектов.

Задача обработки медицинских изображений. В качестве изображений используются изображения внутренних органов человека. Для их получения используют эффект ядерного магнитного резонанса для головы, рентгенографию для изображений легких и скелета, компьютерную томографию для различных органов, ультразвукографию для сердца, почек и других органов. Диагностика состояний внутренних органов базируется на привлечении современных медицинских знаний и выработки качественных методов анализа изображений.

Задача обработки текстовых документов. Здесь объектом реального мира выступает текстовый документ, по которому получают цифровую форму его изображения с помощью сканера. Задача компьютерного зрения может формулироваться как распознавание отдельных символов, слов или перевода цифрового изображения в текстовый файл. Кроме того, может ставиться задача и семантического распознавания текста для индексации его в большой базе данных.

Задача сортировки деталей на конвейере. В таких задачах предполагается ограниченный набор деталей, что позволяет сформировать для них набор характерных параметров. Полученные изображения деталей с помощью камеры (или нескольких камер) необходимо обработать в реальном масштабе времени и выполнить сортировку деталей. При распознавании детали может использоваться трехмерная модель, предварительно сформированная с помощью их характерных параметров.

Задача обработки видеопотока для охранных систем. Как правило, такие задачи связаны с распознаванием движения объектов на фоне неподвижной сцены. Здесь также задействованы камеры, настроенные на передачу изображения некоторой сцены. Априорно предполагается, что сцена в процессе времени остается неизменной, или почти неизменной, т.е. динамика движения возможна в ограниченных пределах. Задача заключается в обнаружении движущихся объектов, возможно без распознавания самих объектов. Похожая задача может выполняться и для различных следящих систем.

Анализ транспортной магистрали. Данная задача близка к предыдущей и отличается от нее некоторыми деталями по скорости движения объектов и возможными элементами распознавания типа движущего объекта и номера автомобиля.

Задача обработки спутниковых изображений. Обработка спутниковых изображений может иметь разные аспекты. Могут решаться задачи сельскохозяйственного назначения, анализа и состояния лесных массивов, обнаружения пожаров, оценки состояния снежного покрова перед весенними паводками, задачи распознавания военного назначения и другие. На решение этих задач оказывают существенное значение качество аппаратуры для съемок, погодные условия и скорость обработки информации. Во многих случаях для эффективного решения задач может быть полезным использование мультиспектральных изображений.

В задачах компьютерного зрения существенное значение имеет обработка изображений для получения необходимой информации. Несмотря на существование большого количества приемов такой обработки, их можно разбить на несколько категорий в зависимости от назначения и методики обработки. Например, можно рассмотреть такие категории методов:

- повышения качества изображения;
- получения изображения, пригодного для обработки по заданному алгоритму;
- получение изображения в качестве выходного результата;
- получения результата описательного характера.

Рассмотрим более подробно содержание этих методов [1].

Корректировка окрестностей пикселей. Назначение этих методов состоит в подавлении шумов и упрощении изображения за счет удаления мелких деталей изображения. Иногда применяется корректировка граничных пикселей сплошных областей изображения. Эта операция в некоторых случаях позволяет четче разделить отдельные элементы изображения. Корректировка граничных пикселей заключается в замене соседних пикселей на фоновые значения.

Улучшение качества изображения. Операция выполняется для всего изображения по одному алгоритму. Улучшение достигается за счет изменения яркости (повышения или понижения), а также за счет удаления шумов. Получение нужного результата добиваются подбором соответствующих фильтров. Нередко используются фильтры, позволяющие изменить контрастность изображения или выделить контуры объектов.

Получение изображения по нескольким исходным. В этой категории методов используются операции вычитания или сложения изображений. Операция вычитания позволяет выявлять движущиеся объекты, а сложения – получить новое изображение с суммарными элементами исходных изображений. Как правило, в качестве исходных используются два изображения, но возможны и другие варианты.

Вычисление характеристик изображения. В зависимости от решаемой задачи можно подобрать характерные признаки распознаваемых объектов: длина, ширина, площадь, цвет, характерные контуры, ориентация и прочие. По составленному набору характеристик проводится подсчет их на изображении. Установив некоторые допустимые диапазоны значений заданных характеристик, можно проводить анализ результатов вычислений и делать итоговые выводы.

Получение описательного результата. Эти методы относятся большей частью именно к задачам компьютерного зрения. Результаты решения задачи могут выдаваться в виде подсчета некоторых объектов на изображении, наличия или отсутствия каких-то признаков, в виде текстового файла, полученного по его графическому изображению. Возможно получение описания сцены с указанием перечня выявленных объектов, их характеристиками и ориентацией на сцене.

Задачи компьютерного зрения в настоящее время широко востребованы во многих областях деятельности человека. Это объясняется успехами современных ИТ-технологий в области программно-аппаратных средств вычислительной техники – появление мощных центральных и графических процессоров, языковых и алгоритмических средств. Однако, как уже говорилось, решение задач компьютерного зрения сегодня значительно уступает решению этих задач человеком и требуют дальнейшего совершенствования.

Работа по подготовке и изданию учебного пособия проведена в рамках выполнения базовой части государственного задания (проект № 3442 "Информационно-алгоритмическое обеспечение систем цифрового управления, автономной высокоточной навигации и технического зрения для перспективных летательных аппаратов: разработка теоретических основ проектирования, алгоритмов, способов эффективной и надежной программной реализации, использование высокопроизводительной вычислительной инфраструктуры для экспериментального моделирования").

1. Средства и методы формирования изображений

1.1. Устройства для формирования изображений

1.1.1. Камеры на основе приборов с зарядовой связью

Такие камеры создаются с использованием приборов с зарядовой связью (ПЗС). Свет от внешнего источника (трехмерная сцена) через объектив попадает на плоскость изображения в виде дискретной матрицы, состоящей из твердотельных ячеек, преобразующих световую энергию в электрический заряд. Предварительно очищенные ячейки накапливают заряд в течение некоторого времени, определяемого временем открытия затвора. Матрица ячеек обладает памятью, которая затем считывается и после аналого-цифрового преобразователя передается в буфер кадра. Далее это изображение может передаваться в компьютер для дальнейшей обработки.

1.1.2. Видеокамеры

Видеокамеры предназначены для формирования последовательности кадров, отображающих трехмерную сцену в динамике. Кадры формируются с определенной частотой, обычно 30 кадров в секунду. Для повышения непрерывности восприятия движения человеком используют прием представления кадра двумя блоками с прореженным через строчку изображением. Такой метод носит название чересстрочной развертки кадра. Блоки в этом случае содержат четные и нечетные строки кадра. Частота выдачи кадров при этом составляет 60 раз в секунду, что для человека дает лучшее восприятие изображения за счет свойства инерционности глаза. Современные видеокамеры, ориентированные на машинную обработку, обладают достаточной скоростью воспроизведения, поэтому развертка кадра выполняется построчно без разбиения на полукадры. Следует отметить, что видеокамеры кроме изображения записывают и звуковое сопровождение, а сами кадры разделяются маркерами.

Для уменьшения объема информации видеопоследовательности записываются с использованием каких-то способов сжатия. Для чего существуют различные форматы записи и сжатия.

1.1.3. Микроденситометры

Датчики этого типа предназначены для сканирования слайдов или фотопленок на просвет. В этом случае сканируемые кадры размещаются между источником света и датчиком. Датчик записывает значения, пропорциональные интенсивности прошедшего через кадр света. Так как датчик является точечным, то для получения полного изображения необходимо перемещать кадр для его сканирования. При этом требуется достаточно точная механика перемещения. Преимущество микроденситометров заключается в том, что у них отсутствуют ошибки измерения интенсивности по сравнению с ПЗС-матрицей. Кроме того, они дают большее разрешение изображения. Главным недостатком их является пониженная скорость обработки.

1.1.4. Датчики для цветных и мультиспектральных изображений

Цветные изображения могут получаться в ПЗС-камерах при использовании разноцветного вращающегося диска. Диск изготовлен из прозрачной пленки, окрашенной в основные цвета: красный-зеленый-синий. Сектора разных цветов имеют равные площади. Скорость вращения диска подбирается таким образом, чтобы в течение одного считывания перед датчиком проходил только один цвет. За один оборот диска производится три считывания, в результате чего выполняется формирование трех компонент цветности одного пикселя. Процесс формирования пикселя длится в течение одного оборота диска, что приводит к пространственным искажениям изображения, так как все три компонента считываются в разные моменты времени движущегося спутника.

Мультиспектральные изображения получают в таких же ПЗС-камерах, в которых используются тонкие преломляющие пленки. Преломляющие средства разделяют падающий луч на спектр из 4-х компонент. Каждая компонента фиксируется своим элементом ПЗС-матрицы, что в совокупности формирует набор цветных изображений, полученных для световых волн разной длины.

На спутниках могут устанавливаться сканеры направленного зондирования земной поверхности. При этом каждый участок земной

поверхности сканируется направленной оптической системой с одновременным подавлением излучения остальных участков. Излучение от сканируемого участка проходит через призму, которая разлагает его в спектр. Полученный спектр разбивается на диапазоны, каждый из которых фиксируется ПЗС-матрицей в самостоятельное изображение. По числу диапазонов для каждого пикселя формируется вектор значений интенсивности соответствующей длины. За один проход получается одна строка двумерного изображения. В процессе движения спутника по орбите формируются последовательно все строки двумерного изображения. Данная технология получения мультиспектрального изображения также характеризуется пространственными искажениями. Мультиспектральные изображения показывают удовлетворительные результаты распознавания характера земной поверхности – водные, лесные сельскохозяйственные, горные и другие.

1.1.5. Датчики рентгеновского изображения

Источники и датчики рентгеновского излучения используются в медицине для получения изображений частей тела человека, в машиностроении – для контроля сварных и литых деталей, контроля продукции в пищевой промышленности и других областях. Такие изображения фиксируют интенсивность рентгеновских лучей, которые проходят через анализируемый объект. В зависимости от плотности и качества материала меняется распределение этой интенсивности по изображению.

Данная технология используется и для получения трехмерной картины наблюдаемого объекта. В этом случае применяются рентгеновские томографы. Полное представление картины достигается последовательным сканированием объема тела и формированием вокселей, отображающих объемный характер тела. Полученная таким образом информация о трехмерном объекте позволяет при необходимости анализировать его в нужных направлениях и плоскостях в виде двумерных изображений.

1.1.6. Изображения, полученные методом ЯМР

Другой метод получения трехмерных изображений основан на эффекте ядерного магнитного резонанса для объектов, состоящих из различных веществ. Как правило, ЯМР-томографы используются в медицине для диагностики различных заболеваний. При получении

изображения томограф регистрирует интенсивность сигнала ЯМР для каждого вокселя – элемента трехмерного пространства.

В магниторезонансной ангиографии интенсивность сигнала зависит от скорости потока крови, что позволяет диагностировать многие заболевания, связанные с состоянием сердечно-сосудистой системы пациента. Методику ЯМР можно использовать и в других сферах деятельности человека, но в настоящее время расширение сфер применения сдерживается высокой стоимостью аппаратуры и процедуры сканирования.

1.1.7. Дальнометрические изображения

Существуют устройства, позволяющие получать изображения с отображением интенсивности излучения элементов трехмерных поверхностей и расстояния до них. Эти устройства-дальномеры имеют лазерный излучатель с амплитудной модуляцией. Излучатель посылает лазерный луч и затем принимает отраженный сигнал. Расстояние до отражающей поверхности вычисляется на сравнении фазы излученного и отраженного сигнала. Как правило, таким способом измеряются расстояния в пределах одного периода волны модулирующего сигнала. Для больших расстояний вычисление возможно, если известно число полных периодов сигнала.

Эти датчики позволяют измерять и интенсивность отраженного сигнала. Поэтому в данном случае формируется два совмещенных изображения – изображения дальности и интенсивности в полутонном виде. Эти устройства имеют невысокую скорость обработки, а аппаратура отличается высокой стоимостью. Эти характеристики сужают области их использования. В настоящее время они применяются в горнодобывающей робототехнике, в космических роботах, в автомобильной промышленности и некоторых других.

1.1.8. Ультрасонографические изображения

Исследования с помощью ультразвуковой аппаратуры (УЗИ) широко используют в медицине. Они носят название ультрасонографии – метода получения изображений с использованием высокочастотных звуковых волн. В медицине УЗИ используют для диагностики внутренних органов – печени, желчного и мочевого пузыря, почек, щитовидной железы и других. Для кровеносных сосудов используют ультразвуковое исследование – доплерографию, с помощью которой

определяют направление и скорость потока крови, а также изменения в сосудах, вызванные болезнью, их функциональное состояние.

1.2. Особенности формирования цифровых изображений

При формировании цифровых изображений могут возникать различные **геометрические искажения**. Это обычно связано с несовершенством технических средств и точности их изготовления. Эти искажения дают искажения форм, линий, окружностей. Причем искажения могут проявляться неравномерно по площади изображения.

При съемках объектов, удаленных на большие расстояния, могут оказывать влияние состояния погодных условий, наличия влаги в атмосфере, температуры воздуха. Эти условия влияют на прохождение лучей, их преломление и рассеивание. Такие искажения называются **дисперсией**.

Искажения – **блюминг** (избыточная яркость) – объясняется влиянием элементов ПЗС-матрицы на соседние элементы. В этом случае часть накопленного заряда элемента ПЗС-матрицы при ярком освещении передается на соседние элементы, что приводит к увеличению размеров яркого пятна.

Неоднородность элементов ПЗС-матрицы за счет погрешностей технологии их изготовления приводит к разной их чувствительности к яркости света. В связи с этим элементы с одинаковой освещенностью могут давать различные показания. Снижение этого искажения добиваются частично за счет калибровки элементов ПЗС-матрицы, при которой для них вырабатываются масштабные коэффициенты, позволяющие корректировать считываемые значения яркости. Возможны случаи, когда отдельные элементы матрицы могут вообще не действовать. Таким ячейкам присваиваются значения яркости, вычисленные как средние значения соседних ячеек.

При формировании цифровых изображений следует учитывать **диапазон** допустимых значений интенсивности света. Выход за границы диапазона дадут искажения, существенно изменяющие истинные значения яркости. Чтобы исключить такие искажения, следует увеличивать диапазон возможных значений или вести контроль выхода за его пределы.

Световые волны различной длины преломляются линзой в разной степени. Поэтому световые волны от точечного элемента сцены после

преломления могут попасть сразу на несколько соседних пикселей, давая при этом **эффект размытия**.

Преобразование аналогового сигнала в цифровой дает дополнительное искажение за счет **дискретизации и округления** значения сигнала.

1.3. Системы координат для трехмерных изображений

При формировании трехмерных изображений, их обработке и анализе возможно использование пяти различных систем координат. В каждом конкретном случае могут применяться часть этих систем. С другой стороны, каждый объект трехмерной сцены имеет свою собственную объектную систему координат, и в этом смысле число используемых систем координат будет зависеть от состава трехмерной сцены. Ниже рассматриваются виды этих систем.

1.3.1. Экранная система координат

Эта система координат S имеет две оси X и Y . Ось X направлена слева направо, а ось Y – сверху вниз. Пределы изменения значений определяются разрешающей способностью экрана, а экран представляется целочисленной решеткой пикселей. Каждый пиксель p на экране представляется парой координат $p(x, y)$ и может иметь только положительные значения.

1.3.2. Система координат объекта

Система координат объекта Q используется в трехмерной графике для описания его геометрической модели. Поэтому данную систему иногда называют объектной или модельной. Так как в данном случае рассматриваются трехмерные объекты, то, естественно система координат имеет три измерения. Расположение начала системы координат и направление ее осей определяются конкретно для каждого объекта исходя из требований задачи.

1.3.3. Система координат камеры

Эта система координат C привязана к камере, и она определяет расположение камеры относительно сцены. В зависимости от ориентации камеры изображение сцены имеет разное представление. В связи с этим

важным является выбор такого положения камеры в каждом случае, чтобы оно максимально способствовало решению поставленной задачи.

1.3.4. Декартовая система координат изображения

Данная система координат D отличается от экранной системы расположением ее начала, которое помещается в середине экрана. Направление осей – как и у обычной декартовой системы координат. Поэтому значения координат пикселей могут принимать как положительные, так и отрицательные величины. Еще одно отличие заключается в том, что на эти координаты не накладывается ограничение их целочисленности. Пиксель может представляться трехкомпонентным вектором, в котором третья компонента имеет значение интенсивности пикселя.

1.3.5. Мировая система координат

Мировая система координат W нужна для описания объектов трехмерных сцен. Многие алгоритмы трехмерной графики и компьютерного зрения работают в так называемом объектном пространстве и оперируют координатами мировой системы координат.

На рис. 2 показаны описанные системы координат. Вершина A куба может быть определена своими координатами в каждой из них.

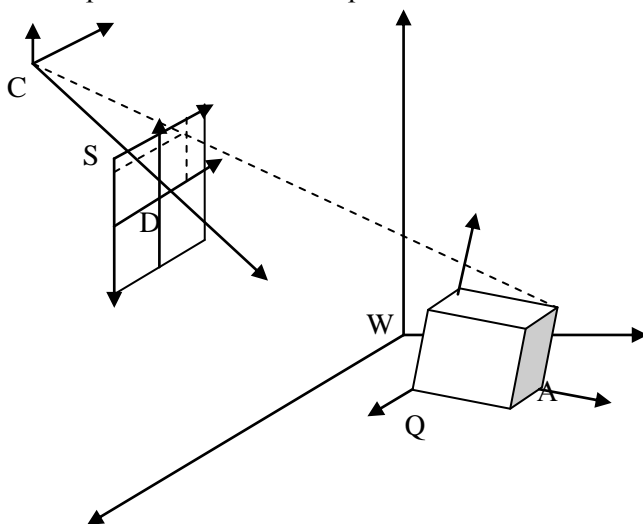


Рис. 2. Системы координат

2. Анализ бинарных изображений

В большинстве задач компьютерного зрения и трехмерной графики используются полноцветные изображения. Однако в процессе решения частных задач нередко возникает необходимость перехода к обработке полутоновых или монохромных изображений. Полутоновые изображения обычно имеют 256 степеней градации яркости, а монохромные, как правило, черно-белые (бинарные), только два уровня, обозначаемые 0 или 1.

2.1. Обработка черно-белых изображений

Работа с двучетными изображениями привлекает разработчиков достаточно простыми алгоритмами, и если удастся решить какую-то часть задачи с помощью бинарного изображения, то такая методика является предпочтительней по сравнению с другими подходами.

Получение бинарного изображения можно выполнить по цветному или полутоновому изображению. В зависимости от решаемой задачи может выбираться и алгоритм перевода исходного изображения в черно-белое. Наиболее распространен метод пороговой бинаризации. В этом случае определяется некоторый порог (диапазон) яркости или цвета, в соответствии с которым каждый пиксель переводится в значение 0 или 1.

При обработке или анализе изображений рассматривают не только обрабатываемый пиксель, но и соседние с ним пиксели. Здесь возможны два варианта соседних пикселей, определяемых связностью рассматриваемых областей. В этом смысле области могут быть 4- или 8-связными. То есть у каждого пикселя может быть 4 или 8 соседних пикселей (рис. 3).

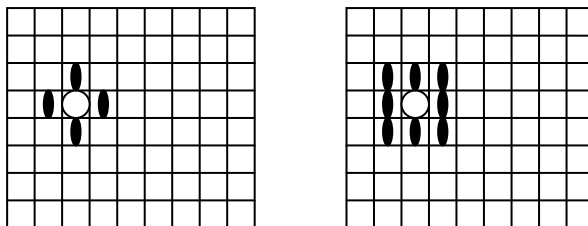


Рис. 3. Примеры 4- и 8-связных областей

Наиболее распространенный метод обработки изображений, в том числе и бинарных, связан с использованием фильтров (масок). Фильтры

представляют собой некую матрицу коэффициентов, обычно квадратную. Размер матрицы обязательно с нечетным числом элементов 3x3, 5x5 и т.д. Этой матрице соответствует окно (апертура), накладываемое на исходное изображение. Обработка заключается в умножении значений пикселей, попавших в окно, на соответствующие коэффициенты. Полученные произведения суммируются и делятся на нормирующий коэффициент, как правило, равный сумме коэффициентов фильтра. Полученный результат присваивается в качестве нового значения пикселя, обычно – центрального. Эта процедура представляет собой шаг обработки одного пикселя. После этого выполняется переход к обработке следующего пикселя. Таким образом, после обработки всех пикселей изображения получается новое результирующее изображение. Иногда полученное изображение может быть подвергнуто новой обработкой с помощью того же фильтра. Такие фильтры называются рекурсивными, т.е. полученное в результате обработки изображение становится исходным для следующей обработки.

Широкий набор фильтров позволяет получать различные изображения, способствующие решению различных задач распознавания и компьютерного зрения.

2.2. Подсчет объектов на изображении

Исходным является бинарное изображение, содержащее некоторые объекты, цвет пикселей которых обозначается значением 0, а фон – цветом 1. Объекты представляются сплошными областями, не содержащими дыр. Каждый прямоугольный объект имеет 4 внешних угла. Если все объекты имеют форму прямоугольника, то достаточно подсчитать число внешних углов и разделить это число на 4. Таков алгоритм подсчета числа объектов на изображении для простейшего случая. Если объект имеет более сложную конфигурацию, то кроме внешних углов появляются и внутренние углы. При этом число внешних углов увеличится на величину, равную числу внутренних углов.

Поэтому алгоритм подсчета объектов для общего случая [1] заключается в подсчете числа внешних углов E и числа внутренних I . Количество объектов определится как

$$Q = (E - I)/4.$$

Каждый пиксель совмещается с левым верхним элементом маски.

Подсчет внешних углов можно выполнить за счет использования масок следующего вида (рис. 4):

0	0
0	1

0	0
1	0

1	0
0	0

0	1
0	0

Рис. 4. Маски для подсчета внешних углов

Необходимо проверить на совпадение этого пикселя и трех его соседей с одной из четырех указанных на рисунке масок. Если произойдет совпадение с какой-то маской, то данный пиксель представляет собой внешний угол объекта. Аналогично подсчитывается число внутренних углов объектов по следующим маскам (рис. 5):

1	1
1	0

1	1
0	1

0	1
1	1

1	0
1	1

Рис. 5. Маски для подсчета внутренних углов

Ниже приводится алгоритм подсчета объектов на бинарном изображении на псевдокоде.

```

{
E = 0;
I = 0;
for L = 0 to MaxX - 1
  for P = 0 to MaxY - 1
    {
      if ext_match(L, P) then E = E + 1;
      if int_match(L, P) then I = I + 1;
    }
Q = (E - I)/4;
}
```

Здесь требуется детализация простых операций определения внешних и внутренних углов, которые достигаются организацией двойного цикла сравнения четырех пикселей каждой из четырех масок и формированием значения true, если четыре пикселя совпадут со значениями одной из масок, и false – в противном случае.

2.3. Маркировка связных компонент изображения

Алгоритм маркировки связных компонент бинарного изображения заключается в выделении всех связных компонент и присвоением каждой из них своей метки (идентификатора). Эти метки отличают компоненты между собой. Связными компонентами являются такие области изображения, в которых можно найти путь между двумя любыми пикселями, то есть они имеют связь между собой. Нахождение пути выполняется с перемещениями по четырем направлениям для 4-связных компонент и по восьми – для 8-связных компонент.

Известны разные алгоритмы решения указанной задачи. Рассмотрим в качестве примера рекурсивный алгоритм [1]. Дается бинарное изображение В размером $\text{MaxX}+1$ и $\text{MaxY}+1$. Необходимо получить маркированное изображение LB, в котором каждому пикселю присвоена метка компоненты, к которой он принадлежит.

Алгоритм строится следующим образом. На первом этапе выполняют смену знака у всех пикселей со значением 1. Это необходимо для различения необработанных пикселей (значение -1) и пикселей, помеченных идентификаторами. При этом изображение В переводится в текущее значение изображения LB. Затем выполняется поиск пикселя со значением -1. Ему присваивается значение очередной метки. Для каждого соседнего пикселя рекурсивно повторяется операция присвоения метки. Алгоритм маркирования бинарного изображения, записанный на псевдокоде, приведен ниже.

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

```
procedure find_components(LB, label);  
{  
  for L = 0 to MaxX  
    for P = 0 MaxY  
      if LB[L, P] == -1 then  
        {
```

```

        label = label + 1;
        search(LB, label, L, P);
    }

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

```

В приведенной процедуре следует только уточнить несложные процедуры присвоения отрицательных значений negate, выбора соседних пикселей neighbors и печати print.

2.4. Морфология бинарных изображений

Операция определения морфологии бинарных изображений имеет своей целью определения свойств формы и структуры каких-либо объектов. В компьютерном зрении под морфологией бинарных изображений обычно понимают описание свойств формы областей на изображении. Применяется операция, как правило, после маркировки бинарных изображений. Для выполнения морфологии изображений используют некоторый набор операций и структурирующие элементы.

2.4.1. Структурирующие элементы

Для выполнения операций бинарной морфологии необходимо бинарное изображение B и структурирующий элемент S (structuring element). Структурирующий элемент представляет собой некоторую область в виде бинарного изображения, которая описывает форму какого-либо объекта. Существует набор наиболее часто используемых стандартных структурирующих элементов, дополняемый при необходимости другими элементами в зависимости от решаемой задачи. Примерами структурирующих элементов могут быть области в виде прямоугольника, эллипса, тора, уголка, двутавра, полосы и т.д.

Один из пикселей структурирующего элемента выбирается в качестве начала координат и является точкой привязки при наложении его на обрабатываемое изображение. Такой точкой можно выбирать центр элемента, любую угловую точку или вообще любой пиксель. Используя начало координат, можно выполнять наложение структурирующего элемента на исходное изображение для выполнения необходимых операций и анализа результата.

2.4.2. Основные операции бинарной морфологии

К основным операциям относят операции наращивания (dilation), эрозии (erosion), замыкания (closing) и размыкания (opening). Смысл этих бинарных операций заключается в увеличении области изображения операцией наращивания, уменьшения – операцией эрозии, заливки дыр внутри областей – операцией замыкания и удаления несущественных выступающих деталей на изображении сплошных областей. Поясним более подробно суть выполнения операций.

Операция **наращивания** математически определяется как операция объединения:

$$B \oplus S = \bigcup_{b \in B} S_b .$$

Она применяется для всех пикселей исходного бинарного изображения В. Для этого начало координат структурирующего элемента совмещается с очередным пикселем изображения В и выполняется операция объединения и результат записывается в выходное изображение. После этого производится переход к следующему пикселю. По завершении обработки всех пикселей получают итоговое бинарное изображение.

Операция **эрозии** бинарного изображения В структурирующим элементом S определяется как

$$B - S = \{b \mid b + s \in B \forall s \in S\} .$$

При выполнении этой операции также выполняется обработка каждого пикселя изображения В наложением структурирующего элемента. Если в некотором положении каждый единичный пиксель структурирующего элемента совпадает с единичным пикселем бинарного изображения, то выполняется логическое сложение центрального пикселя структурирующего элемента с соответствующим пикселем выходного изображения.

Операция **замыкания** бинарного изображения задается выражением

$$B \bullet S = (B \oplus S) - S.$$

Операция **размыкания** бинарного изображения задается выражением:

$$B \circ S = (B - S) \oplus S.$$

Кроме описанных морфологических операций возможно использование и других бинарных операций в зависимости от требований практической задачи. Бинарная морфология может использоваться для извлечения характерных признаков объекта, которые затем могут служить информацией для распознавания объектов. Такими признаками, которые получаются с помощью бинарной морфологии, могут быть конфигурация объектов, геометрические размеры, пропорции его частей, периметр, центр тяжести и другие.

Еще одним примером использования бинарной морфологии является выделение на бинарном изображении компонент, у которых форма и размеры удовлетворяют заданным параметрам. В этих случаях можно подобрать соответствующую форму структурирующего элемента, с помощью которого можно получить необходимый результат. При этом для дальнейшей обработки могут потребоваться и необработанные компоненты. Для таких случаев была введена еще одна операция под названием условного наращивания (conditional dilation). Определим операцию условного наращивания. Пусть B – исходное бинарное изображение, C – обработанное бинарное изображение, S – структурирующий элемент. И пусть $C_0 = C_n$, и

$$C_n = (C_{n-1} \oplus S) \cap B.$$

Условным наращиванием изображения C элементом S по отношению к B называется операция

$$C \oplus|_B S = C_m,$$

где индекс m равен наименьшему значению, при котором $C_m = C_{m-1}$.

Согласно определению множество $C = C_0$ циклически наращивается структурирующим элементом S , и на каждом шаге результат уменьшается до подмножества пикселей, которые имеют единичные значения на исходном изображении B .

2.5. Свойства бинарных областей

После обработки бинарного изображения и получения множества бинарных областей на нем нередко требуется проведение анализа этих областей для определения их свойств, которые будут служить исходной информацией для последующего распознавания объектов. При этом

может стоять задача вычисления набора свойств. Примерами таких наиболее распространенных свойств могут быть наборы:

- геометрических свойств – площадь области, размеры, центр тяжести;
- свойств формы – прямоугольные, округлые, вытянутые, зигзагообразные;
- свойства интенсивности – яркие, темные, средние;
- свойства текстуры – характерные рисунки, повторяющиеся узоры, статистика областей.

Ниже приводятся определения некоторых часто используемых геометрических свойств и свойств формы областей. В этих определениях множество пикселей области обозначается буквой R , и каждый пиксель считается квадратным.

2.5.1. Площадь области и центр тяжести

Площадь области A определяется суммой единичных пикселей области

$$A = \sum_{(x,y) \in R} 1.$$

Определение координат $(x_{\text{цт}}, y_{\text{цт}})$ центра тяжести области вычисляется следующим образом:

$$x_{\text{цт}} = \frac{1}{A} \sum_{(x,y) \in R} x;$$
$$y_{\text{цт}} = \frac{1}{A} \sum_{(x,y) \in R} y.$$

2.5.2. Длина периметра

Периметр области представляет собой множество граничных пикселей области. В этом определении предполагается отсутствие внутренних отверстий в областях. Пиксель области R является граничным, если один из его соседей не принадлежит R . Для четырехсвязных областей граничные пиксели образуют восьмисвязные области, и, наоборот, у восьмисвязных областей граничные пиксели четырехсвязные. Множества пикселей периметра P_8 для 4- и P_4 для 8-связных областей определяются следующим образом:

$$P_8 = \{(x, y) \in R \mid N_4(x, y) - R \neq 0\};$$

$$P_4 = \{(x, y) \in R \mid N_8(x, y) - R \neq 0\}.$$

Длина периметра $|P|$ вычисляется по упорядоченной цепочке соседних пикселей периметра $P = \langle (x_0, y_0), (x_1, y_1), \dots (x_{k-1}, y_{k-1}) \rangle$ по следующему выражению:

$$|P| = |\{k \mid (x_{k+1}, y_{k+1}) \in N_4(x_k, y_k)\}| + \sqrt{2} |\{k \mid (x_{k+1}, y_{k+1}) \in N_8(x_k, y_k) - N_4(x_k, y_k)\}|,$$

где k – длина последовательности пикселей;

$k+1$ – вычисляется по модулю K (K равно длине последовательности пикселей).

В соответствии с этим определением длины периметра два соседних четырехсвязных пикселя увеличивают длину периметра на единицу, а восьмисвязных – на величину $\sqrt{2}$.

2.5.3. Свойство округлости

На основании понятий площади A и периметра P вводится понятие округлости (circularity) области как квадрата длины периметра, деленного на площадь области:

$$C_1 = \frac{|P|^2}{A}.$$

Ввиду того, что значение округлости имеет различные значения для одной и той же области в зависимости от определения ее 4- или 8-связной, а также от того, что она принимает минимальное значение не для окружности, а для ромба и восьмиугольников, было предложено [5] другое определение округлости:

$$C_2 = \frac{\mu_R}{\sigma_R},$$

где μ_R – среднее отклонение граничных пикселей от центра тяжести области;

σ_R – среднеквадратичное отклонение граничных пикселей от центра тяжести области.

Указанные значения отклонений рассчитываются по формулам, приведенным ниже. Здесь множество пикселей (x_k, y_k) , $k = 0, \dots, K-1$ принадлежит периметру P рассматриваемой области. Второе определение округлости ведет себя похожим образом для цифровых и непрерывных образов фигур.

$$\mu_R = \frac{1}{K} \sum_0^{K-1} \|(x_k, y_k) - (x_{um}, y_{um})\|;$$

$$\sigma_R = \left(\frac{1}{K} \sum_0^{K-1} [\|(x_k, y_k) - (x, y)\| - \mu_R]^2 \right)^{1/2}.$$

2.5.4. Описывающий прямоугольник и экстремальные точки

Для каждой области бинарного изображения можно построить описывающий прямоугольник в виде минимаксной оболочки по минимальным и максимальным значениям координат x и y . Эта оболочка для i области будет иметь следующие координаты:

$$x_{\min}(i), x_{\max}(i), y_{\min}(i), y_{\max}(i).$$

Прямоугольная оболочка будет иметь стороны, параллельные осям координат.

На каждой стороне описывающего прямоугольника в общем случае можно отметить по две экстремальные точки: на верхней и нижней стороне – крайние левые и крайние правые, на левой и правой стороне – крайние верхние и крайние нижние. Эти экстремальные точки обозначают контур бинарной области (рис. 6).

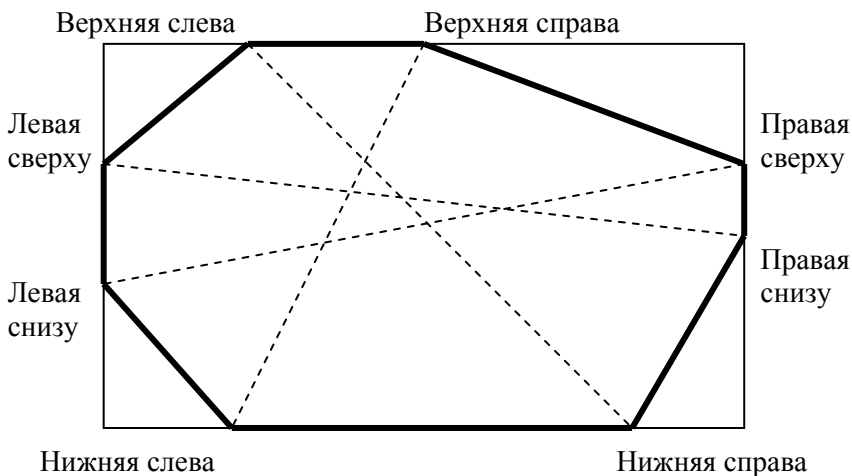


Рис. 6. Экстремальные точки описывающего многоугольника

Линии соединения противоположных экстремальных точек дают соединительные оси, для которых, в некоторых случаях, может быть полезным определить длину и направление.

Длина экстремальной оси может быть вычислена так:

$$D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} + Q(\theta),$$

где $Q(\theta)$ – поправка, учитывающая погрешность перехода к целочисленным координатам;

θ – угол, определяющий направление экстремальной оси.

Значения поправки лежат в интервале от 0 до 1 и определяются

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

2.5.5. Моменты второго порядка для областей

Для описания формы области применяют и моменты второго порядка. Такими моментами являются центральные моменты по строке, столбцу и смешанные. Вычисляются эти моменты по нижеприведенным формулам.

Центральный момент второго порядка по строке:

$$\mu_{xx} = \frac{1}{A} \sum_{(x,y) \in R} (x - x_{um})^2.$$

Центральный момент второго порядка по столбцу:

$$\mu_{yy} = \frac{1}{A} \sum_{(x,y) \in R} (y - y_{um})^2.$$

Смешанный центральный момент второго порядка:

$$\mu_{xy} = \frac{1}{A} \sum_{(x,y) \in R} (x - x_{um})(y - y_{um}).$$

У некоторых битовых областей центральные моменты относительно некоторых направлений могут сильно различаться. В этом случае можно выделить главную ось, которая будет являться характерной осью по сравнению с другими. Главной осью объекта на плоском бинарном изображении будет та ось, относительно которой пиксели области имеют наименьший момент второго порядка. По аналогии с механикой эта ось является осью, относительно с которой момент инерции объекта

минимален. Можно заметить, что момент инерции кругового диска относительно всех осей имеет одинаковое значение.

Известно, что главная ось, соответствующая минимальному моменту инерции, проходит через центр тяжести $(x_{цт}, y_{цт})$ множества пикселей области. Множество моментов относительно заданного набора осей может служить набором дополнительных характеристик бинарных изображений при распознавании образов. Для этого используется формула расчета моментов второго порядка для произвольной оси.

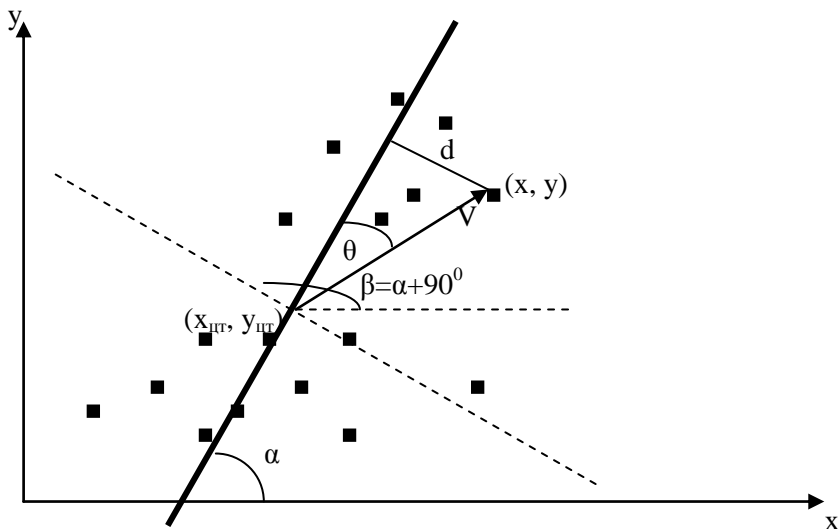


Рис. 7. Пример бинарной области с осью, проходящей через центр тяжести

На рис. 7 представлена бинарная область с указанием произвольной оси, направленной через центр тяжести под углом α . Для определения момента второго порядка множества пикселей, относительно указанной оси, необходимо для всех пикселей вычислить сумму квадратов расстояний d до оси и выполнить нормировку делением на количество пикселей области. Здесь d определяется как проекция вектора V на ось, перпендикулярную к выбранной оси ($\beta = \alpha + 90^\circ$), через скалярное произведение векторов.

$$\mu(x_{цт}, y_{цт}, \alpha) = \frac{1}{A} \sum_{(x,y) \in R} d^2 = \frac{1}{A} \sum_{(x,y) \in R} (V \circ (\cos \beta, \sin \beta))^2 =$$

$$= \frac{1}{A} \sum_{(x,y) \in R} ((x - x_{um}) \cos \beta + (y - y_{um}) \sin \beta)^2.$$

С помощью приведенной формулы можно считать моменты второго порядка относительно любых осей. Часто используются такие моменты относительно горизонтальных, вертикальных или диагональных осей, например, для распознавания текста.

Наибольший интерес представляют моменты второго порядка относительно осей с углами α^* , дающих экстремальные значения момента инерции. Для этого необходимо исследовать выражение для $\mu(x_{цт}, y_{цт}, \alpha)$ на экстремум путем дифференцирования по углу α .

$$\begin{aligned} \operatorname{tg} 2\alpha^* &= \frac{2 \sum (x - x_{um})(y - y_{um})}{\sum (x - x_{um})(x - x_{um}) - \sum (y - y_{um})(y - y_{um})} \\ &= \frac{\frac{1}{A} 2 \sum (x - x_{um})(y - y_{um})}{\frac{1}{A} \sum (x - x_{um})(x - x_{um}) - \frac{1}{A} \sum (y - y_{um})(y - y_{um})} \\ &= \frac{2\mu_{xy}}{\mu_{xx} - \mu_{yy}}. \end{aligned}$$

Углы осей α^* с максимальными и минимальными значениями моментов инерции различаются на 90° . В каждом конкретном случае их можно легко различить.

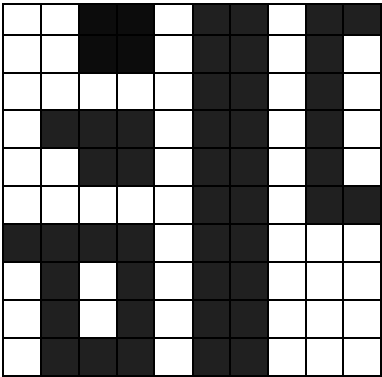
2.6. Графы смежности областей

При анализе бинарных изображений, кроме свойств областей, интерес представляет структура распределения областей. На изображении рассматриваются области двух видов: переднего плана и фона. Ранее рассматривался алгоритм маркировки областей переднего плана. Аналогично может быть поставлена задача маркировки и областей фона. Если маркировка переднего плана использовала натуральный ряд чисел, то удобно промаркировать области фона числами 0, -1, -2 и т.д.

После выполнения маркировки можно построить граф смежности областей, в котором вершины соответствуют областям переднего плана и фона, а ребрами – соседство (смежность) областей. Области считаются смежными, если они имеют соседние пиксели. Причем соседство может рассматриваться как для 4-, так и 8-связных компонент.

Алгоритм построения графа смежности областей заключается в анализе изображения по две строки. Такой подход позволяет обнаруживать смежность областей. В случае наличия соседства двух разных областей переднего плана и фона, в граф добавляется соответствующая связь для этих областей.

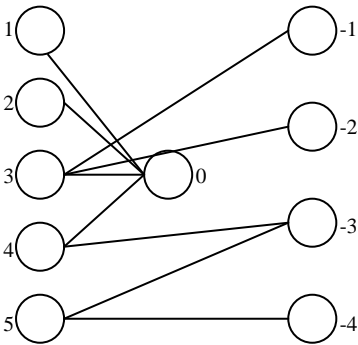
Ниже на рис. 8 приведен пример бинарного изображения, маркировки и графа смежности областей.



Бинарное изображение

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

Маркировка областей



Граф смежности

Рис. 8. Пример маркировки и графа смежности бинарного изображения

2.7. Пороговая бинаризация изображений

Бинарное изображение обычно получают по полутоновому изображению. Поэтому при наличии цветного изображения ведут его

обработку последовательно: цветное – полутоновое – бинарное. Первое преобразование цвет – полутон выполняется вычислением среднеарифметического значения по трем базовым цветам. Преобразование полутонового изображения в бинарное изображение может выполняться по одному из трех вариантов:

- по верхнему пороговому значению;
- нижнему пороговому значению;
- диапазону.

Например, в первом случае, если значение яркости превышает порог, то данный пиксель считается пикселем переднего плана. Аналогично поступают и в остальных случаях. В этой процедуре важным является выбор порогового значения или диапазона, так как это существенно влияет на получение конечного результата решения задачи распознавания.

Выбор порога пользователь может выполнять произвольно методом «проб и ошибок» или, используя гистограмму изображения. В последнем случае возможен выбор порога, который хорошо различает границу раздела светлых и темных пикселей. Известны методы автоматического выбора порогового значения бинаризации. В некоторых случаях выполняют динамическую пороговую бинаризацию, при которой для разных частей изображения подбирают разные пороги.

Наконец, существуют приемы пороговой бинаризации на основе априорной информации. В этом случае порог бинаризации выбирается на основе предварительной информации о распознаваемых объектах и характере их расположения на изображении.

3. Обработка полутоновых изображений

Обработка изображений предполагает получение выходного изображения в результате выполнения одной или нескольких операций над исходным изображением. В качестве операций обработки изображения могут выполняться фильтрации, улучшения изображений, изменения качественных характеристик, выделения каких-то параметров и т.д. Обработка изображения может основываться на анализе окрестности отдельного пикселя или на основе характеристик всего изображения.

Если функция преобразования изображения определяет новое значение пикселя по значению одного исходного пикселя, то она является точечным оператором. Примером такого точечного оператора является оператор усиления контраста, в результате применения которого усиливаются детали изображения. Важным свойством точечных операторов является возможность независимой обработки отдельных пикселей. Это позволяет выполнять параллельную обработку изображения.

В данном разделе рассматриваются различные методы улучшения изображения операциями подавления шумов, сглаживания, повышения контраста и усиления краев.

3.1. Увеличение диапазона полутонов

Увеличение диапазона интенсивности изображения существенно улучшает качество изображения за счет усиления контраста. Для этого используют три метода – линейная растяжка, нормализация и выравнивание гистограмм. Все эти методы направлены на расширение диапазона интенсивности до максимального интервала от 0 до 255.

Линейная растяжка гистограммы заключается в преобразовании уровней яркости исходного изображения. Преобразование выполняется по формуле

$$g_i = a + b \cdot f_i,$$

где f_i – значение яркости i -го пикселя исходного изображения;

g_i – значение яркости выходного изображения;

a, b – коэффициенты линейной растяжки.

При таком подходе диапазон яркости исходного изображения от f_{\min} до f_{\max} преобразуется в диапазон от 0 до 255. Исходя из этого коэффициенты a, b могут быть рассчитаны по формулам:

$$a = 255 \frac{f_{\min}}{f_{\min} - f_{\max}}; \quad b = \frac{255}{f_{\max} - f_{\min}}.$$

Нормализация гистограммы также предусматривает растяжку гистограммы, но не всего диапазона яркости исходного изображения (от f_{\min} до f_{\max}), а только ее части – наиболее интенсивного участка (от \hat{f}_{\min} до \hat{f}_{\max}). При этом исключают из рассмотрения малоинформативные «хвосты». Поэтому предварительно необходимо установить преобразуемый интервал яркости ($\hat{f}_{\min} - \hat{f}_{\max}$), а затем выполнить преобразование по тем же формулам линейной растяжки.

Выравнивание гистограммы (эквализация) ставит своей задачей такое преобразование гистограммы, после которого все уровни яркости имеют примерно одинаковую частоту. При этом гистограмма яркостей будет иметь равномерный закон распределения.

Эту операцию выполняют с условием использования всех допустимых значений интенсивности (использование всего диапазона) и выходное изображение должно иметь примерно равное количество пикселей для каждого значения интенсивности. Таким образом, в идеале гистограмма, преобразованная с указанными условиями, дает гистограмму, у которой 256 уровней, а количество пикселей каждого уровня равно $(\text{MaxX} * \text{MaxY}) / 256$.

Для определения функции преобразования f , растягивающей динамический диапазон изображения, необходима гистограмма исходного изображения. Для каждого уровня интенсивности гистограммы $H_{in}[i]$ указывается количество пикселей, имеющих значение интенсивности z_i .

Первое пороговое значение интенсивности t_1 определяется наращиванием индекса i на гистограмме входного изображения до тех пор, пока суммарное количество пикселей с меньшими значениями интенсивности не превысит порог q_1 . Всем пикселям входного изображения со значением интенсивности $z_k < t_1 - 1$ на выходном изображении будут присвоены значения z_1 . Пороговое значение t_1 будет определяться с помощью выражения:

$$\sum_{i=1}^{t_1-1} H_{in}[i] \leq q_1 < \sum_{i=1}^{t_1} H_{in}[i].$$

Из выражения следует, что для интенсивности t_1 имеется не более q_1 пикселей с меньшими значениями.

В общем случае для k -го порогового значения выражение имеет вид

$$\sum_{i=1}^{t_k-1} H_{in}[i] \leq (q_1 + q_2 + \dots + q_k) < \sum_{i=1}^{t_k} H_{in}[i].$$

Для записи преобразования f обычно используют таблицу преобразования (lookup table, LUT), которая формируется на основе приведенного выше выражения.

3.2. Удаление мелких деталей

За редким исключением, мелкие детали на изображении являются шумом. Удаление таких областей повышает качество изображения. Для этого используются разные методы, которые позволяют удалять как отдельные пиксели, так и области, интерпретируемые в качестве шума на изображении. Ниже приводятся два таких алгоритма.

3.2.1. Удаление шума «соль и перец»

Шум под названием «соль и перец» представляет собой одиночные темные пиксели внутри светлых областей (перец), или одиночные светлые пиксели внутри темных областей (соль). Он может возникать в результате пороговой бинаризации изображения, а также после аналого-цифровых преобразований. Возможны и другие случаи возникновения шума. Удаление шума «соль и перец», как правило, основано на применении масок, с помощью которых выявляется наличие шума, и масок, которые удаляют этот шум. На рис. 9 и 10 показаны маски для случаев бинарного изображения и бинарного маркированного изображения. Кроме того, указаны маски для 4- или 8-связных областей.



Рис. 9. Выявление и устранение шума на бинарном изображении



Рис. 10. Выявление и устранение шума на бинарном маркированном изображении 4- и 8-связных областей

Здесь левые маски в каждой паре служат для выявления того или иного случая, а правые – для удаления шума.

3.2.2. Удаление мелких областей

Удаление мелких областей, которые могут быть отнесены к шумам, выполняется на основе расчета их признаков. Такими признаками могут быть размер области (площадь, количество пикселей), геометрические особенности (вытянутость, маленький размер одного из измерений габаритов). В каждом конкретном случае требуется установить набор таких параметров. После выделения компонент связности битового изображения выполняется удаление тех областей, которые по своим свойствам подпадают под определение шума. В некоторых случаях может требоваться сохранение информации об удаленных областях, если возникнет необходимость возврата к исходному изображению.

3.2.3. Сглаживание изображений

Если изображение содержит кроме основного идеального изображения некоторый случайный шум типа гауссова шума, то улучшение изображения можно достичь применением сглаживающего фильтра. Гауссовый шум описывается нормальным распределением с математическим ожиданием, равным нулю, и среднеквадратичным отклонением σ . Самый простой вариант сглаживающего фильтра содержит единицы для каждого пикселя попадающего в окно апертуры. Размер окна обычно может быть 3×3 или 5×5 . При наложении окна на изображение выполняется расчет нового значения пикселя, который находится в центре окна. Для этого выполняется суммирование всех пикселей в окне и деление полученной суммы на 9 (окно 3×3) или 25 (окно 5×5). То есть новое значение представляет собой среднее арифметическое значение всех пикселей в окне.

Лучшее качество сглаживания получается при использовании фильтра Гаусса, который отличается от описанного тем, что весовые коэффициенты фильтра зависят от расстояния до центрального пикселя. Эти коэффициенты уменьшаются с увеличением расстояния. В этом случае наибольшее влияние на результат оказывают ближние пиксели, а наименьшее – более удаленные. Весовая функция рассчитывается по формуле

$$g(x, y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{d^2}{2\sigma^2}},$$

где $d = \sqrt{(x - x_c)^2 + (y - y_c)^2}$ – расстояние пикселя (x, y) от центрального пикселя (x_c, y_c) .

3.2.4. Медианный фильтр

Устранение шумов сглаживающим фильтром в некоторых случаях приводит к нежелательным результатам, например, к размытию границ или контуров. Поэтому в таких случаях широко используется медианный фильтр, который заменяет значение обрабатываемого пикселя на медиану значений пикселей из его окрестности. Медиана представляет собой значение срединного элемента упорядоченного массива. Если задан упорядоченный массив значений

$$A = (A_0, A_1, \dots, A_n),$$

то медианой этого массива будет значение элемента с индексом $n/2$. При четном числе элементов массива медианой может считаться один из элементов с номерами $(n-1)/2$ или $(n+1)/2$. Иногда медианой считают среднее значение из этих двух элементов.

Применение медианного фильтра позволяет устранять шумы и сохранять структуру границ на изображении. Особенностью этого метода сглаживания состоит в достаточно большом объеме вычислений за счет сортировки массивов.

3.3. Обнаружение краев с помощью дифференциальных масок

Границы раздела областей (края) на изображении обычно определяются вычислением разности значений интенсивности в локальных областях изображения. Такие области разграничения можно выявлять с помощью масок.

3.3.1. Дифференцирование одномерных сигналов

Пусть задан одномерный сигнал в виде числовой последовательности значений некоторой функции f . Тогда производная для такого одномерного сигнала определяется как

$$f'(x_i) \approx \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}}.$$

Если интервал отчета значений функции принять $\Delta x = 1$, тогда производная определяется разностью двух последовательных отчетов. В этом случае подсчет производной одномерного сигнала можно

выполнять с помощью маски $M' = [-1, +1]$. Если в последовательности отчетов функции f взять два соседних отчета и умножить их значения на маску M' и сложить, то получится значение производной на этом интервале

$$f'(x) \approx -f(x_{i-1}) + f(x_i).$$

Аналогично можно рассчитать и вторую производную, используя маску M'' . Для второй производной сигнала имеем

$$\begin{aligned} f''(x_i) &= f'(x_i) - f'(x_{i-1}) = f(x_i) - f(x_{i-1}) - f(x_{i-1}) + f(x_{i-2}) = \\ &= f(x_i) - 2f(x_{i-1}) + f(x_{i-2}). \end{aligned}$$

В этом случае маска будет иметь вид $M'' = [+1, -2, +1]$. Следует отметить, что при вычислении абсолютных значений производных можно использовать и маски с инвертированными значениями коэффициентов:

$$M' = [+1, -1]; M'' = [-1, +2, -1].$$

Такие абсолютные значения производных используются для выявления точек контраста без определения направления смены знака производной.

Кроме указанных масок иногда используется и маска для вычисления первой производной из трех элементов $M = [-1, 0, +1]$, которая используется для последовательных отчетов с номерами $(i-1)$, i , $(i+1)$. Так как $\Delta x = 2$, то вычисленный результат с применением этой маски следует разделить на 2.

Ниже рассматриваются примеры использования масок для выявления различных случаев границы областей:

1. Ступенчатый край, направленный вверх

Отсчеты S_i	12	12	12	12	12	24	24	24	24	24
$M \times S_i$	0	0	0	0	12	12	0	0	0	0

2. Ступенчатый край, направленный вниз

Отсчеты S_i	24	24	24	24	24	12	12	12	12	12
$M \times S_i$	0	0	0	0	-12	-12	0	0	0	0

3. Наклонный край, направленный вверх

S_i	12	12	12	12	15	18	21	24	24	24
$M \times S_i$	0	0	0	3	6	6	6	3	0	0

4. Яркостный импульс

S_i	12	12	12	12	24	12	12	12	12	12
$M \times S_i$	0	0	0	12	0	-12	0	0	0	0

Из этих примеров видно, что маска первой производной из трех элементов дает отклик в двух дискретных элементах для случая ступенчатого края.

Рассмотрим отклик второй производной для тех же случаев. Маска $M = [-1, +2, -1]$.

1. Ступенчатый край, направленный вверх

Отсчеты S_i	12	12	12	12	12	24	24	24	24	24
$M \times S_i$	0	0	0	0	-12	12	0	0	0	0

2. Ступенчатый край, направленный вниз

Отсчеты S_i	24	24	24	24	24	12	12	12	12	12
$M \times S_i$	0	0	0	0	12	-12	0	0	0	0

3. Наклонный край, направленный вверх

S_i	12	12	12	12	15	18	21	24	24	24
$M \times S_i$	0	0	0	-3	0	0	0	3	0	0

4. Яркостный импульс

S_i	12	12	12	12	24	12	12	12	12	12
$M \times S_i$	0	0	0	-12	24	-12	0	0	0	0

Для второй производной характерным является то, что для контраста отклик дает значения с переходом через ноль. Совместное использование первой и второй производных позволяет выявить ряд локальных особенностей исходного сигнала.

Аналогичное применение масок используется для сглаживания сигнала. Маски имеют трехкомпонентный размер, и сумма ее элементов

равна 1. Например, усредняющая маска $M = [1/3, 1/3, 1/3]$. Ниже приведен результат ее применения для ступенчатого края и яркостного импульса.

1. Ступенчатый край, направленный вверх

Отсчеты S_i	12	12	12	12	12	24	24	24	24	24
$M \times S_i$	12	12	12	12	16	20	24	24	24	24

2. Яркостный импульс

Отсчеты S_i	12	12	12	12	24	12	12	12	12	12
$M \times S_i$	12	12	12	16	16	16	12	12	12	12

Маска гауссова сглаживания $M = [1/4, 1/2, 1/4]$ для тех же случаев дает следующий результат:

1. Ступенчатый край, направленный вверх

Отсчеты S_i	12	12	12	12	12	24	24	24	24	24
$M \times S_i$	12	12	12	12	15	21	24	24	24	24

2. Яркостный импульс

Отсчеты S_i	12	12	12	12	24	12	12	12	12	12
$M \times S_i$	12	12	12	15	18	15	12	12	12	12

Можно сформулировать некоторые свойства масок для вычисления производных:

- производные в областях сигнала с высоким контрастом имеют разные знаки;
- сумма элементов масок производных равна 0, поэтому в областях постоянного сигнала дают нулевой отклик;
- точкам высокого контраста соответствуют большие абсолютные значения отклика масок первой производной;
- отклик масок второй производной для высокого контраста дает значения, переходящие через ноль.

Свойства сглаживающих масок:

- элементы сглаживающих масок положительны и в сумме дают единицу; поэтому отклик масок на постоянные значения сигнала совпадает с исходными значениями;
- степень сглаживания и снижения шума пропорциональна размеру маски;
- ступенчатые края сглаживаются пропорционально размеру маски.

3.3.2. Дифференцирование двумерных изображений

Направление контраста функции интенсивности f двумерного изображения определяется вычислением градиента $\left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$. Для дискретной функции частные производные заменяются разностями изменения значений в окрестности пикселя. С целью устранения шума и локальных всплесков используют окрестности размером 3×3 . В связи с этим градиент по двум направлениям будет вычисляться по следующим формулам:

$$\begin{aligned} \frac{\partial f}{\partial x} = f_x \approx \frac{1}{3} [& (I(x+1, y) - I(x-1, y)) / 2 + \\ & + (I(x+1, y-1) - I(x-1, y-1)) / 2 + \\ & + (I(x+1, y+1) - I(x-1, y+1)) / 2]. \end{aligned}$$

$$\begin{aligned} \frac{\partial f}{\partial y} = f_y \approx \frac{1}{3} [& (I(x, y+1) - I(x, y-1)) / 2 + \\ & + (I(x-1, y+1) - I(x-1, y-1)) / 2 + \\ & + (I(x+1, y+1) - I(x+1, y-1)) / 2]. \end{aligned}$$

С целью упрощения вычислений можно опускать деление на 6. При этом отклик масштабируется, что не влияет на определение направления контраста.

Вычисления по указанным выражениям можно выполнять с помощью масок:

$$M_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad M_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}.$$

Иногда контраст определяется с помощью фильтра Собеля и соответствующих ему масок:

$$M_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad M_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}.$$

Здесь отличие состоит в увеличении веса центральной строки и центрального столбца.

Направление градиента и его величину с использованием приведенных выше масок определяют следующим образом:

$$\frac{\partial f}{\partial x} = f_x \approx \frac{1}{6} (M_x \circ N(x, y));$$

$$\frac{\partial f}{\partial y} = f_y \approx \frac{1}{6} (M_y \circ N(x, y));$$

$$|\nabla f| \approx \sqrt{f_x^2 + f_y^2};$$

$$\theta \approx \arctg\left(\frac{f_y}{f_x}\right).$$

Операция скалярного произведения выполняется перемножением элементов масок на соответствующие значения двумерного изображения и суммированием этих произведений.

В случаях, когда предъявляются более жесткие требования к объему вычислений вычисления градиента можно выполнять по упрощенным формулам с небольшой потерей качества вычислений:

$$\max(|f_x|, |f_y|); \quad |f_x| + |f_y|; \quad \frac{f_x^2 + f_y^2}{2}.$$

Можно отметить, что градиентные операторы дают хорошие результаты при обнаружении границ изолированных объектов. Но эти границы могут теряться из-за их кривизны, или за счет плавных переходов полутонов.

3.3.3. Обнаружение краев с помощью фильтрации

Здесь рассматривается фильтрация, использующая функцию Гаусса, и ЛОГ-фильтрация, основанная на применении оператора Лапласа к функции Гаусса.

Функция Гаусса от одной переменной имеет вид

$$g(x) = ce^{-\frac{x^2}{2\sigma^2}},$$

где σ – среднеквадратичное отклонение,
 c – некоторый масштабный множитель.

Эта же функция Гаусса для двух переменных имеет следующий вид:

$$g(x, y) = ce^{-\frac{x^2+y^2}{2\sigma^2}}.$$

Значения масштабного множителя подбираются таким образом, чтобы значения всех элементов маски для фильтрации были бы целочисленными.

Обнаружение краев выполняется путем обработки изображения поэлементно для каждого пикселя и его окрестности с помощью функции Гаусса и его первой и второй производных. Интеграл этой функции (площадь криволинейной трапеции, ограниченной функцией) равен 1. Поэтому использование функции в качестве сглаживающего фильтра не изменяет содержимого областей постоянных значений.

Если принять $c = 1/\sqrt{2\pi}$, функция Гаусса и его производные будут иметь вид:

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

$$g'(x) = \frac{-1}{\sqrt{2\pi}\sigma^3} x e^{-\frac{x^2}{2\sigma^2}} = \frac{-x}{\sigma^2} g(x).$$

$$g''(x) = \left(\frac{x^2}{\sqrt{2\pi}\sigma^5} - \frac{1}{\sqrt{2\pi}\sigma^3} \right) e^{-\frac{x^2}{2\sigma^2}} = \left(\frac{x^2}{\sigma^4} - \frac{1}{\sigma^2} \right) g(x).$$

Используя соотношение $r = \sqrt{x^2 + y^2}$ можно построить выражение для двумерной функции $g(x,y)$ и ее производных. Эти двумерные функции имеют форму поверхностей вращения их одномерных аналогов вокруг вертикальной оси. Каждое сечение этих поверхностей будет соответствовать одномерному гауссиану.

В качестве примеров можно привести две маски аппроксимации гауссиана размерами 3x3 и 7x7. Первая маска получена матричным перемножением вектора-строки $[1 \ 2 \ 1]$ на вектор-столбец $[1 \ 2 \ 1]$:

$$G_{3 \times 3} = \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

Значения для второй маски вычислены по выражению

$$g(x, y) = ce^{\frac{x^2 + y^2}{2\sigma^2}}.$$

Затем выбран коэффициент $c = 90$ с тем расчетом, чтобы минимальный элемент маски был равен 1.

$G_{7 \times 7} =$

1	3	7	9	7	3	1
3	12	26	33	26	12	3
7	26	55	70	55	26	7
9	33	70	90	70	33	9
7	26	55	70	55	26	7
3	12	26	33	26	12	3
1	3	7	9	7	3	1

Эти маски можно использовать для сглаживания двумерных изображений.

3.3.4. Обнаружение краев с помощью ЛОГ-фильтра

Дифференциальный фильтр второго порядка, полученный применением оператора Лапласа к функции Гаусса, называется ЛОГ-фильтром (Laplacian of the Gaussian). Приблизительно ЛОГ-фильтр можно представить в виде разности двух гауссианов:

$$g''(x) \approx c_1 e^{\frac{x^2}{2\sigma_1^2}} - c_2 e^{\frac{x^2}{2\sigma_2^2}}.$$

Чтобы центральная выпуклость функции лежала в положительной полуплоскости, нужно выполнять соотношение $\sigma_1 < \sigma_2$, а значение σ_2 выбирается с учетом значения σ_1 так, чтобы точки перехода через ноль среди значений отклика фильтра располагались корректно, а суммы отрицательных и положительных коэффициентов совпадали.

ЛОГ-фильтр дает значимый отклик для разности интенсивностей в двух случаях – для небольших пятен с центром, совпадающим с центром маски фильтра и для широких ступенчатых краев, расположенных близко от центра маски. Маски малого размера, например, 3×3 , позволяют обнаруживать мелкие детали изображения, вплоть до размера в один пиксель. Маски больших размеров детектируют более крупные детали. При этом в использовании этих масок имеется и такая особенность, как объем вычислений, который обладает значительными вычислениями для масок больших размеров.

Существует гипотеза Марра – Хильдрет, которая предполагает, что в основе человеческого зрения лежит ЛОГ-фильтрация, с помощью которой зрительная система человека строит некоторый эскиз видимой сцены, состоящий из двумерных примитивов (прямые, области, пятна). Для получения полноценной информации о наблюдаемой сцене ЛОГ-фильтрация выполняется с различными значениями σ , что позволяет проработать мелкие и крупные детали. Далее эта информация подвергается обработке на более высоком уровне, при которой происходит соединение результатов фильтрации. Наличие стереографического зрения позволяет получить и трехмерную картину изображения. Предложенная модель неплохо объясняет свойства зрения человека и животных. Однако последние исследования в этой области говорят о необходимости более сложной модели зрительной системы человека. Но, несмотря на это, методы обработки изображений с использованием гауссианов и ЛОГ-фильтрации переменного размера в настоящее время применяются достаточно широко.

3.4. Использование масок для анализа изображения

Решение различных задач распознавания связано с анализом сходства некоторой окрестности изображения с маской. С этой целью разрабатываются наборы масок, предназначенных для выделения признаков конкретного вида. Такой анализ изображений носит название согласованной фильтрации. Она может рассматриваться как для одномерных, так и двумерных изображений, считая первые частным случаем вторых.

3.4.1. Некоторые понятия векторов

Множество всех векторов, состоящих из n действительных чисел, образует векторное пространство. С каждым вектором $S = (s_1, s_2, \dots, s_n)$ связано понятие длины, определяемой как корень квадратный из суммы квадратов его координат:

$$d = \sqrt{s_1^2 + s_2^2 + \dots + s_n^2}.$$

В области обработки сигналов, кроме длины вектора, часто оперируют понятием энергии сигнала. Энергия сигнала Q равна квадрату длины вектора или сумме квадратов его координат:

$$Q = d^2 = s_1^2 + s_2^2 + \dots + s_n^2.$$

Два произвольных вектора S_1 и S_2 являются ортогональными, если их скалярное произведение равно 0:

$$S_1 \circ S_2 = 0.$$

Базис n -мерного векторного пространства представляет собой множество n векторов $\{w_1, w_2, \dots, w_n\}$, которые являются линейно независимыми и покрывают векторное пространство. Покрывание векторного пространства заключается в том, что любой вектор V можно представить в виде линейной комбинации базисных векторов:

$$V = a_1 w_1 + a_2 w_2 + \dots + a_n w_n,$$

где a_i – числовые коэффициенты.

Свойство линейной независимости означает, что никакие из этих базисных векторов w_i нельзя представить в виде линейной комбинации остальных базисных векторов.

Из всех возможных представлений произвольного вектора особый интерес заслуживает представление вектора с помощью ортогонального базиса. Пусть задан базис из трех векторов $\{w_1, w_2, w_3\}$. Любой сигнал $S = \{a_1, a_2, a_3\}$ можно представить как линейную комбинацию базисных векторов $S = a_1 w_1 + a_2 w_2 + a_3 w_3$. Тогда скалярное произведение сигнала S на произвольный вектор из базиса будет равно

$$S \circ w_i = a_1 (w_1 \circ w_i) + a_2 (w_2 \circ w_i) + a_3 (w_3 \circ w_i) = a_i (w_i \circ w_i) = a_i.$$

Здесь

$$w_i \circ w_j = \begin{cases} 0, & \text{при } i \neq j, \\ 1, & \text{при } i = j. \end{cases}$$

С помощью стандартного ортогонального базиса единичной длины любой вектор $V = (v_1, v_2, v_3)$ можно представить в следующем виде:

$$V = (v_1, v_2, v_3) = v_1 (1, 0, 0) + v_2 (0, 1, 0) + v_3 (0, 0, 1).$$

Рассмотрим пример другого множества базисных векторов $\{w_1, w_2, w_3\}$, у которого

$$w_1 = (-1, 0, 1); \quad w_2 = (1, 1, 1); \quad w_3 = (-1, 2, -1).$$

Этот базис обладает тем свойством, что любые его два вектора являются ортогональными, так как скалярное произведение $w_i \circ w_j = 0$ при $i \neq j$. Если сделать эти векторы единичной длины, то они будут иметь значения:

$$w_1 = \frac{1}{\sqrt{2}} (-1, 0, 1); \quad w_2 = \frac{1}{\sqrt{3}} (1, 1, 1); \quad w_3 = \frac{1}{\sqrt{6}} (-1, 2, -1).$$

Можно проиллюстрировать представление некоторого сигнала с помощью этого ортонормированного базиса.

Пусть мы имеем сигнал $S = (8, 16, 20)$. Тогда, используя стандартный базис, можем записать:

$$S \circ w_1 = \frac{1}{\sqrt{2}}(-8 + 9 + 20);$$

$$S \circ w_2 = \frac{1}{\sqrt{3}}(8 + 16 + 20);$$

$$S \circ w_3 = \frac{1}{\sqrt{6}}(-8 + 32 - 20).$$

Отсюда сигнал можно записать как

$$\begin{aligned} S &= (S \circ w_1)w_1 + (S \circ w_2)w_2 + (S \circ w_3)w_3 = \\ &= \frac{12}{\sqrt{2}}w_1 + \frac{44}{\sqrt{3}}w_2 + \frac{4}{\sqrt{6}}w_3. \end{aligned}$$

Тогда энергия будет равна

$$Q = d^2 = \frac{144}{2}w_1 + \frac{44^2}{3}w_2 + \frac{16}{6}w_3 = 72 + 645\frac{1}{3} + 2\frac{2}{3} = 720.$$

Энергию сигнала можно вычислить и непосредственно через квадрат длины вектора

$$Q = d^2 = 8^2 + 16^2 + 20^2 = 64 + 256 + 400 = 720.$$

Таким образом, используя ортонормированный базис, полную энергию можно получить в виде суммы энергетических составляющих, связанных с каждым базисным вектором.

Если учесть, что значение скалярного произведения двух векторов лежит в интервале $(-1, +1)$, то каждое конкретное значение этого произведения может служить мерой схожести векторов, так как оно равно $+1$, если векторы равны и -1 , когда они имеют противоположное направление.

3.4.2. Анализ на основе базиса Робертса

Для анализа полутоновых изображений используется базис Робертса, исследующий окрестности размером 2×2 и состоящий из четырех векторов $\{w_1, w_2, w_3, w_4\}$. Любая окрестность 2×2 действительных значений интенсивности может быть уникальным образом представлена в виде суммы четырех этих базисных векторов, взятых с соответствующими масштабными множителями. Относительная величина масштабного множителя характеризует сходство окрестности

изображения с базисным вектором и поэтому может использоваться для определения структуры окрестности.

Ниже представляется базис Робертса:

$$w_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad w_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad w_3 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad w_4 = \frac{1}{\sqrt{2}} \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}$$

С помощью этого базиса можно представлять различные окрестности изображения размером 2×2 в виде произведения константного базисного изображения на число.

Например, область постоянных значений интерпретируется следующим образом:

$$\begin{bmatrix} 5 & 5 \\ 5 & 5 \end{bmatrix} = 20/2 \left(\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \right) = 10w_1 + 0w_2 + 0w_3 + 0w_4.$$

Примеры ступенчатых краев вертикально и диагонально ориентированных имеют такую интерпретацию:

$$\begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} = 0w_1 + \frac{2}{\sqrt{2}} w_2 + \frac{-2}{\sqrt{2}} w_3 + 0w_4.$$

$$\begin{bmatrix} +1 & +1 \\ -3 & +1 \end{bmatrix} = 0w_1 + \frac{4}{\sqrt{2}} w_2 + 0w_3 + \frac{-4}{2} w_4.$$

И пример для линии:

$$\begin{bmatrix} +1 & +1 \\ -3 & +1 \end{bmatrix} = 8w_1 + 0w_2 + 0w_3 + 8w_4.$$

3.4.3. Анализ на основе базиса Фрея – Чена

Данный базис предназначен для обработки окрестности изображений размером 3×3 . Стандартный базис для окрестностей размером 3×3 содержит 9 векторов, каждый из которых имеет одну единицу и 8 нулей следующего вида (рис. 11):

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \dots \quad \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Рис. 11. Стандартный базис Фрея – Чена

С помощью этого стандартного базиса можно представлять любую матрицу 3x3 в виде суммы соответствующих векторов стандартного базиса, умноженных на масштабный множитель. Например, (рис. 12):

$$\begin{array}{|c|c|c|} \hline 1 & 7 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 4 & 0 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline 1 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} + 7 \begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} + 4 \begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline \end{array}$$

Рис. 12. Пример представления матрицы в виде суммы векторов стандартного базиса Фрея – Чена

Однако разложение по стандартному базису не дает представления о структуре двумерной окрестности. Для анализа двумерной структуры более подходит базис Фрея – Чена, состоящий из множества ортонормированных масок.

Представление окрестности изображения с помощью базиса Фрея – Чена позволяет интерпретировать энергию изображения, рассматривая ее составляющие, соответствующие маскам градиента, пульсации, линии и т.п. Значение энергетической составляющей определяется степенью соответствия структуры изображения некоторому базисному вектору (маске).

Каждый базисный вектор базиса Фрея – Чена разработан специальным образом. Они имеют следующую структуру:

$$\begin{array}{l} \text{Градиент: } W_1 = 1/\sqrt{8} \end{array} \begin{array}{|c|c|c|} \hline 1 & \sqrt{2} & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -\sqrt{2} & -1 \\ \hline \end{array} \quad \begin{array}{l} W_2 = 1/\sqrt{8} \end{array} \begin{array}{|c|c|c|} \hline 1 & 0 & -1 \\ \hline \sqrt{2} & 0 & -\sqrt{2} \\ \hline 1 & 0 & -1 \\ \hline \end{array}$$

$$\begin{array}{l} \text{Пульсации } W_3 = 1/\sqrt{8} \end{array} \begin{array}{|c|c|c|} \hline 0 & -1 & \sqrt{2} \\ \hline 1 & 0 & -1 \\ \hline -\sqrt{2} & 1 & 0 \\ \hline \end{array} \quad \begin{array}{l} W_4 = 1/\sqrt{8} \end{array} \begin{array}{|c|c|c|} \hline \sqrt{2} & -1 & 0 \\ \hline -1 & 0 & 1 \\ \hline 0 & 1 & -\sqrt{2} \\ \hline \end{array}$$

$$\begin{array}{l} \text{Линия } W_5 = 1/2 \end{array} \begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline -1 & 0 & -1 \\ \hline 0 & 1 & 0 \\ \hline \end{array} \quad \begin{array}{l} W_6 = 1/2 \end{array} \begin{array}{|c|c|c|} \hline -1 & 0 & 1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 0 & -1 \\ \hline \end{array}$$

$$\begin{array}{l} \text{Лапласиан } W_7 = 1/6 \end{array} \begin{array}{|c|c|c|} \hline 1 & -2 & 1 \\ \hline -2 & 4 & -2 \\ \hline 1 & -2 & 1 \\ \hline \end{array} \quad \begin{array}{l} W_8 = 1/6 \end{array} \begin{array}{|c|c|c|} \hline -2 & 1 & -2 \\ \hline 1 & 4 & 1 \\ \hline -2 & 1 & -2 \\ \hline \end{array}$$

Константа $W_9 = 1/3$

1	1	1
1	1	1
1	1	1

Рассмотрим пример представления полутоновой окрестности с использованием базиса Фрея – Чена. В качестве полутоновой окрестности возьмем матрицу

$$M = \begin{array}{|c|c|c|} \hline 10 & 10 & 5 \\ \hline 10 & 5 & 5 \\ \hline 10 & 10 & 5 \\ \hline \end{array}$$

Можно определить каждую компоненту для окрестности в направлении каждого базисного вектора. Скалярные произведения окрестности на базисные векторы и энергии компонент приведены в таблице:

Таблица

i	$N \circ W_i$	Q
1	0	0
2	$\frac{10+5\sqrt{2}}{\sqrt{8}} \approx 6$	36,5
3	$\frac{5-5\sqrt{2}}{\sqrt{8}} \approx -0,73$	0,53
4	$\frac{-5+5\sqrt{2}}{\sqrt{8}} \approx 0,73$	0,53
5	2,5	6,25
6	0	0
7	3,3	11,1
8	-0,8	0,7
9	23,3	544,4

Поскольку базис является ортонормированным, то полная энергия изображения равна сумме энергий его компонент. В данном случае эта сумма составляет 600. Полную энергию можно вычислить и как произведение $M \circ M = 600$. Основную энергетическую составляющую дает средний уровень интенсивности вектора W_9 . Наибольшая энергия остальных базисных векторов приходится на градиентный вектор W_2 . Если бы задачей поиска было обнаружение градиентов направления вектора W_2 , то в этом случае можно говорить о нахождении такого признака.

Алгоритм обнаружения окрестностей, имеющих высокую энергию для заданных векторов W_i , можно построить следующим образом.

Исходное полутоновое изображение In анализируется на выявление энергии подпространства, заданного вектором S , превышающей пороговый уровень Q_p . Бинарное выходное изображение Out содержит значение 0 для пикселей с энергией окрестности N , ниже уровня шума и порогового уровня и 1 – в противном случае. Вектор S длиной 8 имеет для j -го элемента $S_j = 1$, если вектор W_j входит в заданное подпространство векторов, и -0 , в противном случае.

1. Цикл по i
2. Цикл по j
3. Если $In[i,j]$ граничный пиксел, то $Out[i,j] = 0$
4. Вычисляется энергия $Q_9 = (N \circ W_9)^2$. Задается энергия подпространства $Q_s = 0$
5. Цикл по k
6. Если $S(k)=1$, тогда $Q_s = Q_s + (N \circ W_k)^2$
7. Конец цикла по k
8. Если $Q_s < Q_{min}$, тогда $Out[i,j] = 0$
9. Если $\frac{Q_s}{N \circ N - Q_9} < Q_p$, тогда $Out[i,j] = 0$
10. $Out[i,j] = 1$.
11. Конец цикла по j
12. Конец цикла по i

4. Обработка цветных изображений

4.1. Физические свойства цвета

В задачах компьютерного зрения цвет играет основную роль, так как обработка цветных изображений по сравнению с полутоновыми изображениями обладает более широким спектром подходов, методов и возможностей их решения. В настоящее время имеется широкий набор аппаратных и программных возможностей по работе с цветными изображениями.

Электромагнитные излучения с длинами волн в диапазоне 400 – 800 нанометров представляют собой видимый свет. В этот диапазон входят все составляющие спектр света от фиолетового до красного. Белый свет получается смешением световых компонент видимого диапазона.

Кроме видимого диапазона, имеется аппаратура детектирования излучения и за его пределами – ультрафиолетового (длина волн меньше

400 нм) и инфракрасного (длина волн больше 800 нм), что дает возможность обрабатывать и эту информацию в задачах компьютерного зрения.

При освещении объектов источником света возникает отраженный свет, который и воспринимается наблюдателем или датчиком камеры. При этом можно фиксировать три основных фактора:

- **энергетический спектр**, который дает распределение световой энергии по длинам волн;
- **отражательная способность** поверхности объекта как результат отраженного света по отношению к падающему;
- **спектральная чувствительность** датчика, на который падает отраженный свет.

Кроме указанных основных факторов, на восприятие сцены влияют и ряд других, связанных как с самими объектами, так и с датчиками, принимающими свет. Для объектов сцены – это материал, из которых они изготовлены, качество и обработка поверхности, а также взаимное влияние объектов на степень их освещенности. Факторы, связанные с датчиками, заключаются в основном с технологическими особенностями камер. Если говорить о восприятии сцены человеком, то следует учитывать особенности зрительной системы человека.

4.2. Цветовые модели

4.2.1. Цветовая модель RGB

В графических системах компьютера широко применяется трехцветная модель RGB (красный-зеленый-синий). Современные системы имеют основной трехбайтовый (24 бита) формат кодирования цвета, при котором на каждую компоненту цветности выделяется один байт памяти. Это дает возможность получать более 16 миллионов цветов. Кроме этого формата еще используются варианты 15 бит (по 5 разрядов на компоненту цвета, 32768 цветов) и 16 бит (красный – 5 бит, зеленый – 6 бит, синий – 5 бит, 65536 цветов).

Один байт памяти дает возможность получения 256 оттенков каждой компоненты цвета от 0 до 255. Коды (255, 0, 0), (0, 255, 0), (0, 0, 255) дают соответственно цвета красный, зеленый, синий. Цвета, у которых коды каждой компоненты равны между собой, имеют полутоновые оттенки, число которых равно 256. Для полутоновых изображений достаточно использовать формат кодирования размером в один байт.

Система RGB называется аддитивной цветовой системой, так как ее цвета получаются суммированием основных цветов с черным цветом (0, 0, 0). В обработке информации иногда используют нормировку цветовых компонент по следующим соотношениям:

- интенсивность $I = (R + G + B)/3$;
- цветовые компоненты
 $r = R/(R + G + B)$; $g = G/(R + G + B)$; $b = B/(R + G + B)$.

При этом для всех цветов сумма $r + g + b = 1$.

4.2.2. Цветовая модель CMY

Цветовая модель CMY используется для цветной печати на белой бумаге. Она носит название субтрактивной и ее цвета получаются на основе вычитания основных цветов из белого цвета. Название системы образовано от английских слов Cyan-Magenta-Yellow (голубой-пурпурный-желтый). Эти цвета соответствуют печатным краскам, поглощающих цвет одного из основных цветов RGB:

голубой цвет = $RGB - R$ (голубая краска поглощает красный цвет);

пурпурный цвет = $RGB - G$ (пурпурная – зеленый цвет);

желтый цвет = $RGB - B$ (желтая – синий цвет).

В этой цветовой модели белый цвет кодируется как (0, 0, 0), а черный – (255, 255, 255). Соответственно цвета голубой, пурпурный, желтый – кодируются (255, 0, 0), (0, 255, 0), (0, 0, 255).

4.2.3. Цветовая модель HIS

Модель Тон-Насыщенность-Интенсивность (Hue-Saturation-Intensity) в своей кодировке имеет значение интенсивности света I и два значения цветности – тон H и насыщенность S . Система используется в графическом дизайне (графических редакторах) из-за удобства контроля яркости и тона, а также в некоторых алгоритмах компьютерного зрения. Последнее применение объясняется более тесной связью модели со свойствами поверхности объектов.

Тон H определяется как угол между вектором насыщенности и осью красного цвета пространственной модели цветовой системы, и его значение меняется от 0 до 2π . При этом красному цвету соответствует угол 0, зеленому – $2\pi/3$, синему – $4\pi/3$. Насыщенность S характеризует чистоту цвета. Значение $S = 1$ соответствует чистому (насыщенному) цвету, а $S = 0$ – ненасыщенному цвету (оттенок серого). Интенсивность I лежит в пределах от 0 до 1.

4.2.4. Пространственные представления цветowych моделей

Для представленных цветowych моделей существует их пространственное представление (интерпретация) – для моделей RGB и CMY в виде единичного куба (рис. 13), а для HIS – в виде шестигранного конуса (рис. 14).

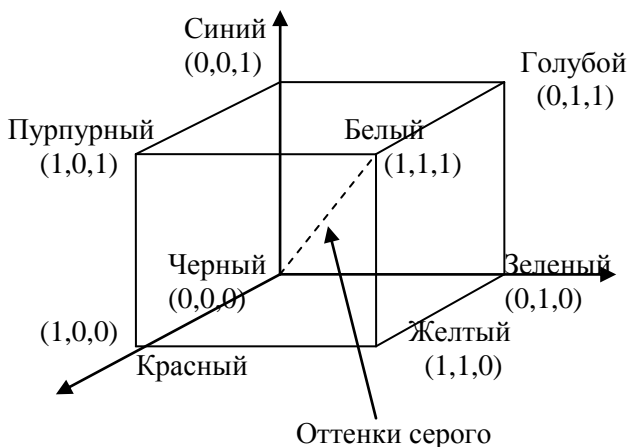


Рис. 13. Единичный куб цветовой модели RGB

Здесь представлен цветовой куб в нормированных координатах RGB, на котором отмечены и цвета модели CMY.

Ниже дается модель цветового пространства для системы HIS. Вертикальная ось этой модели соответствует диагонали цветового куба RGB (оттенки серого) и отображает интенсивность I . Тон H имеет значение угла, отсчитываемого от оси красного цвета. При насыщенности $S = 0$ цвет лежит на вертикальной оси I . Максимальное его значение равно 1.

4.2.5. Цветовые системы YIQ и YUV

Телевизионный стандарт NTSC использует цветовую модель YIQ, содержащую компоненту яркости Y и две цветowych компоненты I и Q . Черно-белые телевизоры используют только компоненту яркости. Линейное преобразование цветовой модели RGB в модель YIQ выполняется по следующим соотношениям:

Яркость	$Y = 0,30R + 0,59G + 0,11B;$
Красный–голубой	$I = 0,60R - 0,28G - 0,32B;$
Пурпурный–зеленый	$Q = 0,21R - 0,52G + 0,31B.$

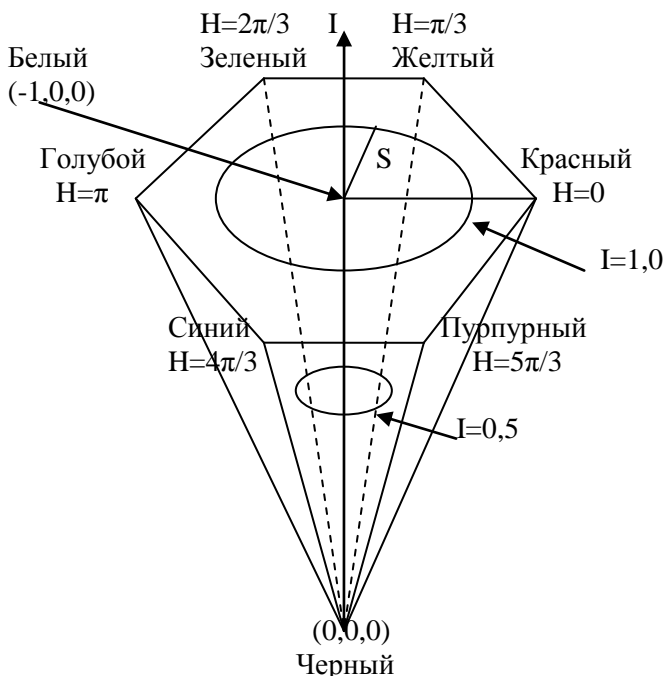


Рис. 14. Шестигранный конус цветовой модели HIS

Учитывая большую чувствительность зрения человека к яркости, чем к цвету при кодировании на компоненту Y выделяют больше разрядов, чем на I и Q .

В алгоритмах сжатия и в некоторых цифровых видеосистемах используется похожая на YIQ цветовая модель YUV . Преобразование цветовой системы RGB в цветовую модель YUV выполняется по соотношениям:

$$Y = 0,30R + 0,59G + 0,11B;$$

$$U = 0,493 (B - Y);$$

$$V = 0,877 (R - Y).$$

4.3. Гистограммы цветных изображений

Гистограмма служит для представления распределения цветов на изображении. С этой целью выполняется подсчет количества пикселей каждого оттенка цвета и отображения их в виде столбчатой диаграммы. Подсчет пикселей ведется раздельно для каждой компоненты цвета.

В некоторых случаях используется построение приближенных гистограмм, которые требуют меньшего объема вычислений с

приемлемой точностью результатов. Например, для 24-битного представления цвета используются 6-битовые коды, составленные из двух старших разрядов каждого байта цветов RGB. В этом случае гистограмма имеет всего лишь 64 значения пикселей.

Гистограммы используются в решении задач поиска в базах данных и решении задач распознавания. Для этого вводится понятие пересечения гистограмм $Int(I, M)$ тестируемого изображения и гистограммы-образца, которое определяется как сумма минимальных значений из каждой пары соответствующих пикселей двух указанных гистограмм:

$$Int(I, M) = \sum_{j=1}^K \min\{I(j), M(j)\},$$

где $I(j)$, $M(j)$ – j -е элементы гистограмм тестируемого изображения и образца соответственно.

Если величину пересечения гистограмм нормировать общим числом пикселей, то полученное значение можно использовать как меру сходства $S(I, M)$ гистограмм и соответственно их изображений:

$$S(I, M) = \frac{Int(I, M)}{\sum_{j=1}^K M(j)}.$$

4.4. Текстура

4.4.1. Понятие текстуры

Текстура играет важную роль в компьютерном зрении и распознавании. Она определяет многие свойства изображения, что оказывает существенное значение для классификации областей изображения.

Текстура дает пространственное распределение цветов и интенсивности на изображении. Анализ этих характеристик позволяет выделять и классифицировать содержание изображения, определять качество материала объектов – вода, почва, деревья, камень, песок облака и т.д.

Понятие текстуры можно определить с двух точек зрения:

1. **Структурный подход.** С точки зрения структуры текстура – это множество примитивных текселей, расположенных регулярным порядком или повторяющимся.

2. **Статистический подход.** В этом случае текстура характеризуется количественными характеристиками распределения значений интенсивности на изображении.

Далее будут рассмотрены эти два варианта описания текстур более подробно.

4.4.2. Тексельное описание текстур

При тексельном описании текстуры необходимо дать описание текселей и их пространственные взаимосвязи. По методу Тусерьяна и Джейн в качестве текселей выбираются области изображения, выделяемые в результате обработки изображения на основе, например, пороговой бинаризации. А пространственная взаимосвязь текселей определяется построением для них диаграммы Вороного.

Диаграммы Вороного строятся так. Имеется множество текселей S . Для двух текселей P и Q из этого множества проводится перпендикуляр, делящий отрезок PQ пополам. Плоскость таким образом делится на две полуплоскости такие, что все точки одной полуплоскости ближе лежат к P , а другой – к Q . Полуплоскость, содержащую точки, более близкие к точке P по отношению к серединному перпендикуляру отрезка PQ обозначим через $H^Q(P)$. Такую операцию можно провести для всех точек Q из множества S . Для каждой точки таким образом строятся многоугольники Вороного, которые содержат все точки, более близкие к ним, чем к любым другим из множества S . Например, для круговых текселей многоугольники имеют вид, показанный на рис. 15.

После построения многоугольников Вороного можно вычислить признаки формы многоугольников и использовать их для группировки многоугольников в кластеры, которые будут задавать области изображения с однородной структурой.

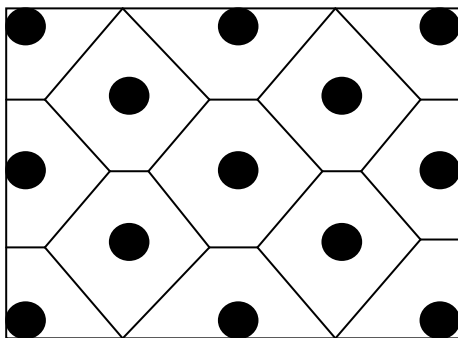


Рис. 15. Многоугольники Вороного для круговых текселей

4.4.3. Количественные характеристики текстур

При анализе естественных изображений выделение текстур является достаточно сложной задачей. Для них лучше подходит вычисление количественных характеристик, с помощью которых удобнее проводить обработку изображений.

Наиболее подходящей характеристикой в этом плане является обнаружение краев. Если области содержат фиксированное количество краев, то это может служить признаком для обнаружения заданных объектов. Кроме количества краев можно использовать и характеристику их направления.

Введем методику подсчета количественных характеристик текстур. Для некоторой области, содержащей N пикселей, выполняется обработка градиентным детектором краев, и для каждого пикселя p подсчитывается модуль (величина) градиента $Gr(p)$ и направление градиента $Dir(p)$. Затем подсчитывается плотность краев, как количество краев на единицу площади:

$$F = \frac{|\{p | Gr \geq T\}|}{N},$$

где T – некоторое пороговое значение градиента.

Набор характеристик можно дополнить свойствами ориентации текстур, используя нормированные гистограммы модуля и направления градиента для пикселей области R . Тогда количественное описание текстуры будет содержать соответствующие гистограммы:

$$F(h) = (H_{Gr}(R), H_{Dir}(R)).$$

Гистограммы обычно имеют небольшое количество разрядов (около 10), и их нормирование выполняется делением на размер области в пикселях. Для сравнения гистограмм двух изображений можно использовать параметр $L(H_1, H_2)$, называемый расстоянием между n -разрядными гистограммами:

$$L(H_1, H_2) = \sum_{i=1}^n |H_1(i) - H_2(i)|.$$

Для бинарных изображений в качестве гистограммы модуля градиента можно использовать двухразрядное представление: один разряд – темный пиксель, второй – светлый. А гистограммы градиента направления выбирать с числом разрядов, соответствующим числу выбранных направлений детектирования. Например, можно выбрать три направления – вертикальное, горизонтальное и диагональное.

Кроме гистограмм градиента модуля и градиента направления в качестве количественной характеристики текстуры используют гистограмму интенсивности каждого пикселя по отношению к его соседним пикселям. Эта характеристика называется локальным двоичным разбиением. Вычисляется она для каждого пикселя в виде 8-битового значения. Каждый разряд b_i этой характеристики равен 0, если интенсивность i -го соседнего пикселя меньше или равна интенсивности самого пикселя, и – 1, в противном случае. В качестве меры сходства двух изображений также можно вычислять расстояние между их гистограммами, по формуле, аналогичной рассмотренной выше.

Еще одна используемая количественная характеристика для текстур – вычисление характерных признаков на основе матрицы вхождений, которая отображает некоторую пространственную структуру распределения значений пикселей на изображении.

Матрица вхождений представляет собой двумерный массив значений $C(i, j)$, которые указывают, сколько раз значение i встречалось в некотором пространственном отношении со значением j . Например, пространственное отношение может определяться так: «значение i является левым соседом для значения j ».

Ниже дается пример матрицы вхождений, в котором множеством допустимых значений для пикселей является множество значений яркости. Для каждой матрицы вхождений задается вектор пространственного отношения значений яркости. Этот вектор определяет взаимное положение двух пикселей, участвующих в вычислении характеристики. Индексы строк и столбцов составлены из всех вариантов значений яркости. Таким образом, подсчитывается количество вхождений в изображение двух яркостей со значениями (i, j) для пикселей, расположенных относительно друг друга на расстоянии dr по строке и dc по столбцу. Полутоновая матрица вхождений C_d для изображения I будет определяться следующим образом:

$$C_d(i, j) = \left| \{ (r, c) \mid I(r, c) = i \text{ и } I(r + dr, c + dc) = j \} \right|,$$

где $d = (dr, dc)$ – вектор взаимного расположения пикселей.

Здесь под яркостью подразумеваются значение уровня полутонового изображения или цвет для цветного изображения.

Существует понятие стандартной полутоновой матрицы вхождений, которая может быть в двух вариантах – нормированной и симметричной. Нормированная матрица определяется по формуле

$$N_d(i, j) = \frac{C_d(i, j)}{\sum_i \sum_j C_d(i, j)}.$$

Симметричная полутоновая матрица вхождений имеет симметричный вид относительно главной диагонали и ее значения вычисляются по матрице вхождений следующим образом:

$$S_d(i, j) = C_d(i, j) + C_{-d}(i, j).$$

На практике вместо матриц вхождений обычно используют набор характерных признаков, вычисленных на основе нормированной матрицы вхождений. Такой подход более эффективен с точки зрения объема вычислений. Такими характерными признаками являются энергия (Q), энтропия (P), контраст (K), однородность (D), корреляция (Cr):

$$\begin{aligned} Q &= \sum_i \sum_j N_d^2(i, j); \\ P &= - \sum_i \sum_j N_d(i, j) \log_2 N_d(i, j); \\ K &= \sum_i \sum_j (i - j)^2 N_d(i, j); \\ D &= \sum_i \sum_j \frac{N_d(i, j)}{1 + |i - j|}; \\ Cr &= \frac{\sum_i \sum_j (i - \mu_i)(j - \mu_j) N_d(i, j)}{\sigma_i \sigma_j}, \end{aligned}$$

где μ_i, μ_j – средние значения;

σ_i, σ_j – среднеквадратичные ожидания сумм $N_d(i) N_d(j)$, вычисленных в направлении строк и столбцов изображения:

$$\begin{aligned} N_d(i) &= \sum_j N_d(i, j); \\ N_d(j) &= \sum_i N_d(i, j). \end{aligned}$$

Наряду с перечисленными числовыми характеристиками в некоторых случаях для анализа текстур используют энергетические текстурные характеристики, а для обнаружения повторяющихся образов и описания зернистости текстурных элементов – вычисления автокорреляционной функции и спектр мощности [1].

Энергетические текстурные характеристики Лавса служат для оценки изменения содержания текстуры в пределах окна фиксированного

размера. Для этого используется набор из девяти масок размерами 5x5. Результаты вычислений энергетических характеристик представляются для каждого пикселя вектором из 9 компонент. Для вычисления масок используются следующие векторы:

$$L5 = [1 \ 4 \ 6 \ 4 \ 1];$$

$$E5 = [-1 \ -2 \ 0 \ 2 \ 1];$$

$$S5 = [-1 \ 0 \ 2 \ 0 \ -1];$$

$$R5 = [1 \ -4 \ 6 \ -4 \ 1].$$

Названия векторов определяются их назначением – для вычисления среднего уровня (Level), выявления краев (Edge), пятен (Spot), ряби (Ripple). С помощью этих одномерных векторов получают двумерные маски перемножением пары одномерных масок из указанного выше набора. Одна из них используется как вектор-столбец, а вторая – как вектор-строка. Количество таких масок равно 16.

Перед расчетом энергетических характеристик выполняется предварительная обработка изображения путем вычитания из пикселей среднего локального значения интенсивности, в результате чего средняя интенсивность каждой пиксельной окрестности будет иметь значения, близкие к нулю. Окно обработки изображения для естественных сцен выбирается размером 15x15.

Затем в результате применения к обработанному изображению масок получают 16 отфильтрованных изображений F , по которым строятся энергетические текстурные карты E :

$$E_k(r, c) = \sum_{j=c-7}^{c+7} \sum_{i=r-7}^{r+7} |F_k(i, j)|,$$

где k – порядковый номер фильтра.

Из полученных 16 карт путем комбинирования симметричных пар, типа L5S5/S5L5, получают девять энергетических карт: L5E5, L5S5, L5R5, E5E5, E5S5, E5R5, S5R5, S5S5, R5R5. Эти карты можно объединить в один трехмерный массив, в котором каждому пикселю в пределах окна 5x5 соответствует вектор из 9 текстурных атрибутов.

4.4.4. Текстурная сегментация

Текстурные характеристики могут использоваться для поиска в базах данных, задачах распознавания, а также для сегментации изображения на области с похожей текстурой. В последнем случае используются два подхода – анализ областей и анализ границ. Первый подход заключается в поиске группы или кластера пикселей, имеющих близкие текстурные

характеристики. При втором подходе решается задача обнаружения краев в текстурах и анализа параметров этих краев.

Для сегментации изображений, кроме прочих методов, широко используются энергетические текстурные характеристики. В общем случае сегментация естественных сцен в настоящее время остается нерешенной задачей.

5. Сегментация изображений

5.1. Понятие задачи сегментации

Суть сегментации изображений заключается в разбиении его на множество областей, покрывающих все изображение. При этом необходимо выделять такие области, которые представляют интерес в соответствии с решаемой задачей. В связи с этим форма и размеры областей могут иметь разный характер – линии, окружности, прямоугольники, многоугольники и т.п.

В процессе сегментации решаются две задачи. Первая – сегментация изображения на области с последующим анализом полученного результата. Вторая – изменение формы описания изображения, которое выполняется с учетом полученного разбиения.

Следует отметить, что не существует универсального метода решения задачи сегментации. Как правило, необходимо выбирать некоторый алгоритм из множества существующих решений для конкретной задачи. Возможно также использование нескольких алгоритмов с последующим выбором окончательного решения. В некоторых случаях можно разрабатывать алгоритм, комбинируя известные подходы.

5.2. Обнаружение областей

Можно сформулировать некоторые требования к выделенным областям, которым они должны удовлетворять. Например, следующие:

- области должны быть связными и однородными по какому-либо признаку – интенсивность, цвет, текстура;
- области должны иметь простую форму и не содержать внутри мелких отверстий и других деталей;
- смежные области сегментированного изображения должны существенно различаться по значению характерного признака;

- границы областей должны быть гладкими, без разрывов и зазубрин.

Перечисленные требования понятны, но не приходится надеяться на то, что они могут быть выполнены одновременно. В каждом конкретном случае можно пополнить список перечисленных требований, выполнение которых будет существенно облегчать решение задачи.

Кластеризация в задачах распознавания представляет собой процесс разбиения множества векторов признаков на подмножества, называемые кластерами. Существуют разнообразные алгоритмы кластеризации. Ниже будут рассмотрены следующие методы кластеризации:

- классические алгоритмы;
- простые гистограммные методы;
- рекурсивный гистограммный метод Оландера;
- метод графового разбиения Ши.

5.3. Методы кластеризации

5.3.1. Классические алгоритмы кластеризации

Задача кластеризации состоит в разбиении векторов на подмножества векторов. В этих алгоритмах векторами обычно являются пиксели или окрестности пикселей. Множество векторов может содержать следующие компоненты:

- значения интенсивности;
- коды цвета или цветовые характеристики;
- вычисленные характерные признаки;
- текстурные характеристики.

Кластеризацию можно выполнять по любому характерному признаку. После кластеризации областей с помощью алгоритмов маркировки связанных компонент можно легко найти связанные области. Ошибка кластеризации D может быть вычислена по формуле

$$D = \sum_{k=1}^K \sum_{x_i \in C_k} \|x_i - m_k\|^2,$$

где C_1, C_2, \dots, C_k – кластеры;

m_1, m_2, \dots, m_k – математические ожидания.

Эта величина показывает близость исходных данных к полученной кластеризации. Поиск оптимального решения путем перебора всех вариантов кластеризации с подсчетом величины D позволяет найти

хорошее решение. Но этот путь требует большого объема вычислений. Поэтому на практике используют приближенные методы.

5.3.2. Итерационная кластеризация по математическому ожиданию

Решение задачи строится итерационным методом поиска экстремума. На первой итерации ($it=1$) случайным образом задается K значений математических ожиданий для кластеров:

$$m_1(1), m_2(1), \dots, m_k(1).$$

Затем для каждого вектора x_i вычисляется расстояние

$$D = \sum_{k=1} \sum_{x_i \in C_k} \|x_i - m_k(it)\|^2.$$

Расчеты расстояния выполняются для всех кластеров. По результатам расчета вектор x_i перемещается в кластер C_j с ближайшим вектором математического ожидания. После этого выполняется переход на новую итерацию ($it = it + 1$) с уточнением значений математического ожидания и расчетом новых расстояний D .

Итерации повторяются до тех пор, пока для всех кластеров будет выполняться условие

$$C_k(it) = C_k(it+1),$$

т.е. кластеры остаются неизменными при переходе на очередную итерацию.

Алгоритм можно начинать не с задания случайных значений математических ожиданий, а со случайного разбиения кластеров и расчета для них математических ожиданий. Данный алгоритм в общем случае дает локально оптимальный результат, однако он хорошо сходится и имеет умеренный объем вычислений.

5.3.3. Алгоритм кластеризации ISODATA

Алгоритм ISODATA является разновидностью итерационных алгоритмов, в основе которого лежит метод разделения-объединения кластеров. В этом методе используется ковариационная матрица кластера вида:

$$\Sigma_k = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \dots & \sigma_{1n} \\ \sigma_{21} & \sigma_{22} & \dots & \sigma_{2n} \\ \dots & \dots & \dots & \dots \\ \sigma_{n1} & \sigma_{n2} & \dots & \sigma_{nn} \end{bmatrix},$$

где $\sigma_{ii} = \sigma_i^2$ дисперсия i -й векторной компоненты, а $\sigma_{ij} = \rho_{ij}\sigma_i\sigma_j$ представляет собой ковариацию i -й и j -й векторных компонент;
 ρ_{ij} – коэффициент корреляции между i -й и j -й компонентами;
 σ_i – среднееквадратичное отклонение i -й компоненты;
 σ_j – среднееквадратичное отклонение j -й компоненты.

Алгоритм формирования кластеров строится следующим образом. Имеется множество кластеров C_1, C_2, \dots, C_k с математическими ожиданиями m_1, m_2, \dots, m_k и ковариационная матрица Σ_k для k -го кластера. Шаги алгоритма:

1. Поместить вектор x_i в кластер 1, для которого достигается минимум

$$D_M = (x_i - m_l)' \Sigma_l^{-1} (x_i - m_l).$$

2. Объединить кластеры i и j , если

$$|m_i - m_j| < \tau_v,$$

где τ_v – пороговое значение дисперсии.

3. Разделить кластер k , если максимальное собственное значение Σ_k превышает порог τ_v .
4. Завершить работу, если для каждого кластера I выполняется условие

$$|m_i(t) - m_i(t+1)| < \varepsilon,$$

или если достигнуто максимально допустимое значение итераций.

5.3.4. Простые гистограммные методы

Гистограммные алгоритмы с точки зрения вычислений являются более эффективными, так как процедура вычислений является однопроходной.

В качестве примера ниже рассматривается алгоритм поиска моды гистограммы для кластеризации в пространстве измерений. Данный метод хорошо работает с изображениями, у которых однородные объекты образуют кластеры в пространстве измерений. В данном случае таким пространством измерений является гистограмма.

Сегментация изображений выполняется путем обратного отображения кластеров из пространства измерений на изображения. Максимальные связные компоненты на гистограмме используются для выделения сегментов изображения. Для полутоновых изображений в пространстве измерений кластеризация может быть выполнена методом поиска точек минимума на гистограмме. В качестве кластеров выбираются интервалы значений между точками минимума. Маркировка

кластеров на гистограмме выполняется индексами i . Все пиксели, попадающие в один кластер, маркируются соответствующим индексом.

Реализацию алгоритма можно строить на двух принципах. Первый состоит в поиске кластеров между точками минимума гистограммы. А второй – должен учитывать характеристики качества изображения в соответствии с решаемой задачей.

5.3.5. Рекурсивный гистограммный метод

Данный алгоритм Оландера является развитием гистограммного метода кластеризации. Идея алгоритма заключается в рекурсивной процедуре поиска мод. Первоначально поиск мод выполняют на гистограмме полного изображения. Затем этот поиск выполняют на гистограммах областей каждого сформированного кластера. Процедура выполняется, пока удастся производить дальнейшее разделение сегментированных областей.

Разбиение на кластеры выполнялось следующим образом. Сначала формировалась маска для выбора всех пикселей изображения. Затем строилась гистограмма изображения, и выполнялось разбиение на кластеры в пространстве измерений. На основе сформированных кластеров устанавливалась принадлежность пикселей изображения найденным кластерам. Для каждого кластера выполнялась маркировка связных компонент, и строилось множество связных областей, соответствующих кластерам. Затем для каждой связной компоненты генерировалась новая маска и помещалась в стек.

Все маски в стеке предназначены для дальнейшей сегментации. В последующем каждая маска определяла пиксели для построения гистограммы. С помощью гистограммы выполнялось новое разбиение на кластеры в пространстве измерений. Процесс продолжается до исчерпания масок в стеке.

Существует разновидность метода Оландера, по которой гистограммы строятся не для компонент RGB, а для набора из трех переменных $(R+G+B)/3$, $(R-B)/2$ и $(2G-R-B)/4$. Такой алгоритм был предложен Ота в 1980 г.

5.3.6. Алгоритм графового разбиения Ши

Описанные выше алгоритмы Оландера и Ота хорошо работают на простых цветных изображениях с одноцветными областями, характерными для искусственных объектов и сцен. Естественные сцены и изображения обычно содержат много мелких деталей, сложную

текстуру и большое число сегментированных областей. Для изображений с такими характеристиками был предложен алгоритм сегментации изображения на основе графовой модели, на которой решается задача разбиения графа [6].

Модель строится в виде взвешенного графа $G = (V, E)$, вершины которого представляют точки в пространстве измерений. Ребрам ставится в соответствие веса $w(i, j)$, определяющие степень сходства вершин i и j . Ставится задача разбиения графа на непересекающиеся множества V_1, V_2, \dots, V_m , которые имеют наибольшие сходства элементов внутри множеств и наименьшие сходства элементов разных множеств.

Если множество вершин V разбить на два непересекающихся подмножества A и B , то характеристика различия этих подмножеств может быть выражена как сумма весов ребер, соединяющих вершины из разных подмножеств. Удаление этих ребер приводит к разрезу (сечению) графа на два подграфа с вершинами A и B . Сечение графа определяется так:

$$cut(A, B) = \sum_{u \in A, v \in B} w(u, v).$$

С точки зрения решаемой задачи сегментации изображения ставится задача нахождения минимального разреза графа и повторение этой процедуры до тех пор, пока сегментированные области не станут достаточно однородными. В таком виде алгоритм может привести к формированию небольших множеств изолированных вершин, что является нежелательным при наличии больших однородных областей.

В связи с этим Ши предложил использовать другую характеристику – нормированного разреза. Для этого введена еще одна характеристика – ассоциация множества A и полного множества V :

$$asso(A, V) = \sum_{u \in A, t \in V} w(u, t).$$

Тогда нормированный разрез определяется следующим образом:

$$Ncut(A, B) = \frac{cut(A, B)}{asso(A, V)} + \frac{cut(A, B)}{asso(B, V)}.$$

Такая характеристика исключает возможность получения малых величин для небольших множеств изолированных вершин. Поэтому задача поиска минимального нормированного разреза в таком случае будет соответствовать решаемой задаче сегментации.

Характеристика связей вершин внутри множеств A и B определяется другой характеристикой – нормированной ассоциации:

$$Nasso(A, B) = \frac{asso(A, A)}{asso(A, V)} + \frac{asso(B, B)}{asso(B, V)}.$$

Связь характеристик нормированного разреза и ассоциации связаны между собой следующим соотношением:

$$Ncut(A, B) = 2 - Nasso(A, B).$$

Данный алгоритм обладает неплохими результатами сегментации цветных изображений реальных сцен, но ввиду большого объема вычислений имеет ограниченное применение, особенно для систем реального времени.

5.4. Нарращивание областей

Нарращивание областей начинается с некоторой точки изображения путем присоединения соседних пикселей, не сильно отличающихся от пикселей области. Здесь рассматривается алгоритм, предложенный Харалик и Шапиро.

Сформированная область R из N пикселей характеризуется средним значением и разбросом:

$$\bar{X} = \frac{1}{N} \sum_{(r,c) \in R} I(r, c);$$

$$S^2 = \sum_{(r,c) \in R} (I(r, c) - \bar{X})^2.$$

Если все пиксели области R и пиксель y (кандидат на присоединение) независимы и их интенсивности описываются одинаковым нормальным распределением, то случайная величина

$$T = \left[\frac{(N-1)N}{(N+1)} (y - \bar{X})^2 / S^2 \right]^{\frac{1}{2}}$$

будет описываться t -распределением (распределением Стьюдента) с $(N-1)$ степенями свободы. Если значение T достаточно мало, то пиксель y присоединяется к области R . После этого выполняется пересчет среднего значения области и разброса.

Получение новых значений можно выполнять с учетом предыдущих значений по формулам:

$$\bar{X}_{new} = \frac{(NX_{old} + y)}{N+1};$$

$$S_{new}^2 = S_{old}^2 + (y - \bar{X}_{new})^2 + N(\bar{X}_{new} - \bar{X}_{old})^2.$$

Если значение T велико, что пиксель y не присоединяется к области. Если значение y отличается от всех соседних областей, то данный пиксель берется за основу новой области.

5.5. Способы представления областей

5.5.1. Оверлейные представления

Данный метод представления областей используется для полутоновых изображений и заключается в выводе сегментированных областей поверх исходного изображения. Вывод границ областей выполняется цветными линиями или белым цветом, который хорошо контрастирует на изображении. При этом, как правило, используют толщину в несколько пикселей.

Во многих графических пакета обработки изображений данный подход реализован в виде стандартной процедуры.

5.5.2. Маркированные изображения

Маркированные изображения используются в процессе обработки изображений как промежуточная процедура подготовки изображения для дальнейшей обработки. Суть операции заключается в индексировании областей уникальными идентификаторами. По исходному изображению строится новое, в котором каждая область изображения представляется в полутоновом виде в соответствии с присвоенными им идентификаторами. Для лучшего представления маркированного изображения используют методы выравнивания гистограмм изображения.

5.5.3. Кодирование границ

В некоторых случаях удобно представлять области их границами. В этом случае информация о сегментированных областях хранится отдельно от изображения, например, в виде линейного списка.

Примером такого рода линейного списка является цепной код Фримана. Принцип кодирования реализуется путем наложения границ областей на квадратную сетку, которая аппроксимирует кривую границы области. Для кодирования выбирается начальная точка, от которой идет переход по нужным направлениям к соседним точкам. Восемь возможных направлений кодируются от 0 до 7, которые соответствуют углам от 0° до 315° . Разница между соседними углами составляет 45° .

Для привязки точек границы к системе координат достаточно хранить координаты начальной точки и сам цепной код. Такой способ

представления областей требует достаточно небольших объемов памяти. При наличии внутри областей отверстий или других областей, они кодируются отдельно.

5.5.4. Квадратные деревья

Квадратные деревья представляют собой еще один способ изображения сегментированных областей. При этом хранится информация не только о границе, но обо всей области в целом. Каждый узел квадратного дерева соответствует некоторой квадратной области и может иметь три состояния – заполненный (full, F), пустой (empty, E), частично заполненный (mixed, M).

Если узел помечен как заполненный, то все пиксели квадратной области принадлежат описываемой области. Если узел помечен как пустой, то все пиксели квадратной области не входят в область. Узлы, помеченные частично заполненными соответствуют квадратным областям, в которых могут быть пиксели, входящие в описываемые области и пиксели, не входящие в них. Дочерние узлы квадратного дерева имеют только узлы, помеченные как частично заполненные.

Чем больше разрешающая способность изображения, тем большую глубину имеет квадратное дерево. Способ представления областей в виде квадратных деревьев широко используется в геоинформационных системах.

5.5.5. Таблицы свойств

Таблицы свойств представляют собой способ хранения информации областей в виде набора свойств: цвет, интенсивность, текстура, размеры, геометрические свойства и другие. В этом случае каждой области соответствует строка таблицы, а столбцы представляют свойства областей.

Таблицы свойств особенно удобны в решении задач поиска в базах данных. В этих случаях используются наборы свойств, позволяющие находить нужные изображения.

5.6. Обнаружение контуров

В некоторых задачах анализа изображений требуется работа не с выделенными областями, а с их границами, которые могут состоять из разных геометрических примитивов – отрезков прямых, дуг окружностей, эллипсов, произвольных кривых. Существуют различные

алгоритмы обнаружения и выделения контуров. Ниже приводятся некоторые из них.

5.6.1. Прослеживание границ областей

Алгоритм прослеживания границ работает с изображениями, в которых выполнена сегментация областей или маркировка связных компонент. Простое решение задачи заключается в том, что перебираются все точки изображения и формируется список, содержащий первые граничные точки каждой области. Затем выполняется обработка каждой области, начиная с первой точки, путем поиска соседних пикселей границы. Поиск ведется по часовой стрелке до возвращения в исходную точку контура.

Описанный подход удобен для небольших изображений, которые располагаются полностью в памяти. Ниже рассматривается алгоритм, работающий с большими изображениями.

Алгоритм работает с предварительно маркированными изображениями путем просмотра всех пикселей слева-направо и сверху-вниз. В результате обработки изображения на выходе получают границы областей, каждая из которых представлена в виде списка пикселей, в порядке обхода границ по часовой стрелке.

Предполагается, что фоновая область на изображении одна, и для тех областей, границы которых не входят в перечень обнаруживаемых, также присваивается метка фоновой области. Процесс организуется таким образом, что при последовательной обработке пикселей изображения, обнаруживаемый пиксель границы области относится к очередной точке границы соответствующей области. В каждый текущий момент обработки изображения в общем случае имеются границы областей, которые уже обработаны, частичные границы областей, находящихся в обработке, и границы областей, ожидающих обработки. Результаты работы алгоритма хранятся в виде структуры данных.

5.6.2. Обнаружение краев Кэнни

Алгоритм работает с полутоновыми изображениями и предназначен для выделения контуров. Детектор краев Кэнни генерирует тонкие фрагменты контуров изображения. Алгоритм оперирует с параметром сглаживания σ и двумя пороговыми значениями интенсивности.

Алгоритм обнаружения краев Кэнни, написанный на псевдокоде [1], приводится ниже.

I[x, y] : входное полутоновое изображение;
 σ : размах, используемый при гауссовском сглаживании;
E[x, y] : выходное бинарное изображение;
IS[x, y] : сглаженное полутоновое изображение;
Mag[x, y] : величина градиента;
Dir[x, y] : направление градиента;
T_{low} — нижнее пороговое значение интенсивности;
T_{high} — верхнее пороговое значение интенсивности;

```

procedure Canny( I[],  $\sigma$ );
{
  IS[] = изображение I[], сглаженное гауссовским фильтром G $\sigma$ (x, y);
  Оператор Робертса для вычисления Mag [x, y] и Dir[x, y] по IS[];
  Suppress_Nonmaxima(Mag[], Dir[], Tlow, Thigh);
  Edge_Detect(Mag[], Tlow, Thigh, E[]);
}
procedure Suppress_Nonmaxima(Mag[], Dir[]);
{
  define +Del[4] = (1,0), (1,1), (0,1) (-1,1);
  define -Del[4] = (-1,0), (-1,-1), (0,-1) (1,-1);
  for x := 0 to MaxX-1;
    for y := 0 to MaxY-1;
      {
        direction := (Dir[x, y] +  $\pi/8$ ) modulo  $\pi/4$ ;
        if (Mag[x, y] ≤ Mag[(x, y)+Del[direction] ])
          then Mag[x, y] := 0;
        if (Mag[x, y] ≤ Mag[(x, y)+-Del[direction] ])
          then Mag[x, y] := 0;
      }
    }
procedure Edge_Detect(Mag[], Tlow, Thigh, E[]);
{
  for x := 0 to MaxX - 1;
    for y := 0 to MaxY - 1;
      {
        if (Mag[x, y] ≥ Thigh) then Follow_Edge(x, y, Mag[],
                                                    Tlow, Thigh, E[]);
      }
    }
procedure Follow_Edge(x, y, Mag[], Tlow, Thigh, E[]);
{
  E [x, y] := 1;
  while Mag[u, v] > Tlow для некоторого 8-связного соседа [u, v]
    пикселя [x, y]
    {
      E[u, v] := 1;
      [x, y] := [u, v];
    }
}

```

Сначала изображение сглаживается гауссовским фильтром с размахом σ . Затем для каждого пикселя сглаженного изображения

вычисляется величина и направление градиента. Направление градиента используется для утончения контуров за счет удаления пикселей, для которых величина градиента не превышает величины градиента в двух соседних пикселях по обе стороны от обрабатываемого вдоль направления градиента. Данный подход используется во многих алгоритмах для получения тонких контуров.

После обработки пикселей по величине градиента выполняется отслеживание непрерывных сегментов контуров. Анализ контуров начинается с краевых пикселей, у которых величина градиента превышает верхнее пороговое значение. Далее отслеживание контура выполняется на всех пикселях, значение градиента которых находится не меньше нижнего порогового значения. Обычно нижнее пороговое значение составляет половину величины верхнего порога.

5.6.3. Группировка контурных фрагментов

Если исходным изображением является изображение, содержащее маркированное контурное изображение с бинарными значениями пикселей, то задача отслеживания контуров становится более сложной, чем для маркированного изображения. В этом случае контурные линии не обязательно являются замкнутыми, и они могут образовывать сегменты с начальными, конечными пикселями, в виде сегментов-стыков и сегментов-углов. Алгоритм отслеживания контурных фрагментов должен выполнять операции:

- обнаружения нового сегмента;
- добавления нового пикселя к сегменту;
- завершение фрагмента (обнаружение конца);
- выявление стыка;
- выявление угла.

Результаты обработки формируются в виде структуры данных. Работа алгоритма заключается в анализе окрестности очередного пикселя. При этом определяется, является ли заданный пиксель изолированной точкой, начальной точкой нового сегмента, точкой стыка или углом. Если пиксель является концевой или внутренней точкой существующего сегмента, то алгоритм определяет и его идентификатор. Если он является угловой точкой или точкой стыка, то алгоритм определяет списки пикселей, входящих в эти сегменты и идентификаторов сегментов.

Алгоритм отслеживания контуров на бинарном контурном изображении, записанный на псевдокоде, приводится ниже.

S[R, C] — входное маркированное контурное изображение.

NLINES — количество строк изображения.

NPIXELS — количество пикселей в строке.

IDNEW — идентификатор нового сегмента.

INLIST — список идентификаторов входящих сегментов, возвращаемый функцией pixeltype.

OUTLIST — список идентификаторов исходящих сегментов, возвращаемый функцией pixeltype.

procedure edge_track(S);

```
{
  IDNEW := 0;
  for R := 1 to NLINES
    for C := 1 to NPIXELS
      if S[R,C] ≠ фоновый пиксел
        {
          NAME := address(R, C);
          NEIGHB := neighbors(R, C);
          T := pixeltype(R,C,NEIGHB,ID,INLIST,OUTLIST);
          case
            T = изолированная точка : next;
            T = начальная точка нового сегмента:
              {
                IDNEW := IDNEW + 1;
                make_new_segment(IDNEW,NAME);
              };
            T = внутренняя точка старого сегмента : add(ID,NAME);
            T = конечная точка старого сегмента :
              {
                add(ID,NAME);
                output(ID); free(ID)
              };
            T = точка стыка или угловая точка:
              for каждый идентификатор ID в списке INLIST
                {
                  add(ID,NAME);
                  output(ID);
                  free(ID);
                };
              for каждый пиксел в OUTLIST
                {
                  IDNEW := IDNEW + 1;
                  make_new_segment(IDNEW,NAME);
                };
          }
        }
  }
```

5.6.4. Обнаружение прямых и дуг окружностей

Определение прямых и кривых линий на полутоновых и цветных изображениях выполняют с помощью преобразования Хафа. В алгоритме используется массив, размерность которого определяется числом неизвестных параметров в уравнении семейства искомых кривых. При проверке каждого пикселя и его окрестности сначала определяется, присутствует ли в данном пикселе достаточно выраженный край. Если присутствует, то определяются параметры уравнения прямой $y = ax + b$, проходящей через данный пиксель.

Накопление данных об отрезках прямых на полутоновом изображении S в аккумуляторном массиве A.

S[R, C] — входное полутоновое изображение.

NLINES — количество строк на изображении.

NPIXELS — количество пикселей в строке изображения.

A[DQ, THETAQ] — аккумуляторный массив.

DQ — дискретное расстояние от прямой до начала координат.

THETAQ — дискретный угол между направлением строк и перпендикуляром к прямой, опущенным из начала координат.

```
procedure accumulate_lines(S,A);
{
  A := 0;
  PTLIST := NIL;
  for R := 1 to NLINES
    for C := 1 to NPIXELS
      {
        DR := row_gradient(S,R,C);
        DC := col_gradient(S,R,C);
        GMAG := gradient(DR,DC);
        if GMAG > gradient_threshold
          {
            THETA := atan2(DR,DC);
            THETAQ := quantize_angle(THETA);
            D := abs(C*cos(THETAQ) - R*sin(THETAQ));
            DQ := quantize_distance(D);
            A[DQ,THETAQ] := A[DQ,THETAQ] + GMAG;
            PTLIST(DQ,THETAQ) :=
              append(PTLIST(DQ,THETAQ),[R,C])
          }
      }
}
```

После обработки всего изображения выполняется анализ полученных параметров. Для этого отыскиваются локальные максимумы в полученном массиве. Эти максимумы будут соответствовать параметрам наиболее вероятных прямых на изображении.

Аккумуляторный массив позволяет находить параметры бесконечно протяженных прямых и кривых, но он не дает возможность обнаруживать отрезки прямых и сегменты кривых. Для этого случая требуется дополнительная детализация алгоритма, по которой выполняется анализ состава пикселей, образующих данный сегмент.

5.6.5. Обнаружение отрезков прямых

Уравнение прямой $y = ax + b$ не подходит для обнаружения вертикальных прямых и отрезков. Более удобным является использование уравнения прямой в виде $d = x \cos\theta + y \sin\theta$, где d – длина перпендикуляра к данной прямой, опущенного из начала координат, а θ – угол между этим перпендикуляром и горизонтальной осью. Если перевести это уравнение в экранную систему координат, то уравнение примет вид $d = x \cos\theta - y \sin\theta$. Экранная система координат ориентирована на дисплее таким образом, что начало системы координат находится в левом верхнем углу, горизонтальная ось координат направлена слева-направо, а ось y – сверху-вниз.

Аккумуляторный массив, который формируется в результате обработки изображения, содержит параметры d и θ , обнаруженных отрезков прямых. При этом вводится их дискретизация. Например, в алгоритме О'Гормана и Кловса [Шапиро], такая дискретизация составляла для d и θ значения 3 и 10° соответственно. Указанный алгоритм приведен ниже.

Алгоритм в виде процедуры *accumulate_lines* обрабатывает полутоновое изображение и формирует аккумуляторный массив **A** и массив списков **PTLIST**. Две функции *row_gradient* и *column_gradient* вычисляют оценки компонент градиента по направлению строк и столбцов окрестностей пикселей, по которым другая функция *gradient* вычисляет значение градиента. Стандартная функция *atan2* дает значение угла.

Процедура *accumulate_lines*, реализующая преобразование Хафа, затем используется в заключительном алгоритме выделения отрезков прямых процедурой *find_lines*. В этой процедуре формируется аккумуляторный массив дискретных значений. Функция *pick_greatest_bin*

возвращает максимальное значение аккумуляторного массива. Функция *reorder* упорядочивает список точек в элементе массива по координате столбца при $\theta < 45^\circ$ или $\theta > 135^\circ$ и по координате строки при $45^\circ \leq \theta \leq 135^\circ$.

Поиск списков точек, соответствующих отдельным прямолинейным отрезкам.

A[DQ, THETAQ] — аккумуляторный массив, заполненный данными о прямых на изображении.

DQ — дискретное расстояние от прямой до начала координат.

THETAQ — дискретный угол между направлением строк и перпендикуляром к прямой, опущенным из начала координат.

```

procedure find_lines;
{
  V := pick_greatest_bin(A,DQ,THETAQ);
  while V > value_threshold
  {
    list_of_points := reorder(PTLIST[DQ,THETAQ]);
    for каждая точка [R, C] в списке list_of_points
      for каждый соседний пиксел [R',C'] для [R, C],
        не входящий в list_of_points
      {
        DPRIME := D[R',C'];
        THETAPRIME := THETA[R',C'];
        GRADPRIME := GRADIENT[R',C'];
        if GRADPRIME > gradient_threshold
          and abs(THETAPRIME-THETAQ) ≤ 10
        then {
          merge(PTLIST[DQ,THETAQ],PTLIST[DPRIME,
            THETAPRIME]);
          set_to_zero[A,DPRIME,THETAPRIME];
        }
      }
    final_list_of_points := PTLIST[DQ,THETAQ];
    create_segments(final_list_of_points);
    set_to_zero[A,DQ,THETAQ];
    V := pick_greatest_bin[A,DQ,THETAQ];
  }
}

```

Процедура *merge* объединяет список точек соседнего пикселя со списком точек данного пикселя. Процедура *set_to_zero* обнуляет элемент аккумуляторного массива для исключения его повторного выбора.

Процедура *create_segments* просматривает окончательно упорядоченного списка точек и ищет в нем промежутки длиной более одного пикселя. Затем она сохраняет множество отрезков прямых, разделенных промежутками.

5.6.6. Обнаружение окружностей

Преобразование Хафа может быть также использовано для обнаружения окружностей и других параметрических кривых.

Поиск списков точек, соответствующих отдельным прямолинейным отрезкам.

A[DQ, THETAQ] — аккумуляторный массив, заполненный данными о прямых на изображении.

DQ — дискретное расстояние от прямой до начала координат.

THETAQ — дискретный угол между направлением строк и перпендикуляром к прямой, опущенным из начала координат.

```

procedure find_lines;
{
  V := pick_greatest_bin(A,DQ,THETAQ);
  while V > value_threshold
  {
    list_of_points := reorder(PTLIST[DQ,THETAQ]);
    for каждая точка [R, C] в списке list_of_points
      for каждый соседний пиксел [R',C'] для [R, C],
        не входящий в list_of_points
      {
        DPRIME := D[R',C'];
        THETAPRIME := THETA[R',C'];
        GRADPRIME := GRADIENT[R',C'];
        if GRADPRIME > gradient_threshold
          and abs(THETAPRIME-THETAQ) ≤ 10
        then {
          merge(PTLIST[DQ,THETAQ],PTLIST[DPRIME,
            THETAPRIME]);
          set_to_zero[A,DPRIME,THETAPRIME];
        }
      }
    final_list_of_points := PTLIST[DQ,THETAQ];
    create_segments(final_list_of_points);
    set_to_zero[A,DQ,THETAQ];
    V := pick_greatest_bin[A,DQ,THETAQ];
  }
}

```

Координаты точек окружности в экранной системе координат можно вычислять по формулам:

$$r = r_0 + d \sin\theta; c = c_0 - d \cos\theta;$$

где r, c – координаты точек окружности;

r_0, c_0 – координаты центра окружности;

d – радиус окружности;

θ – угол между радиусом, соединяющим точку окружности с ее центром и горизонтальной осью.

5.7. Подбор моделей сегментов

При обработке изображений и выделении контуров изображения может возникать задача замены выделенного контура на близкую ему математическую модель в виде прямой, дуги, окружности, эллипса и т.п. При таком подходе выполняется аппроксимация выделенного контура соответствующей моделью. Для подбора такой модели обычно используют метод наименьших квадратов. Например, при выявлении на изображении прямой или отрезка прямой с помощью преобразования Хафа, этот метод можно использовать для проверки пикселей, проголосовавших за данный элемент, на соответствие их уравнению прямой.

При использовании метода наименьших квадратов должен быть известен метод, с помощью которого выполняют подбор соответствующей модели из большого числа возможных вариантов. Ниже рассматривается более детально метод подбора прямой.

5.7.1. Подбор прямой

В качестве модели прямой можно взять уравнение с двумя параметрами:

$$y = f(x) = c_1 x + c_0.$$

Можно взять некоторое множество точек $\{(x_i, y_i), j = 1, n\}$, выявленных как точки прямой, и проверить, лежат ли они на прямой. Для этого нужно подсчитать сумму квадратов отклонений точек от прямой:

$$S = \sum_{j=1}^n (f(x_j) - y_j)^2.$$

Из всех просчитанных моделей выбирается та, у которой достигается минимум суммы S .

Другой метод подбора заключается в использовании среднеквадратичной ошибки, которая определяется по формуле:

$$RS = \left[\sum_{j=1}^n ((f(x_j) - y_j)^2) / n \right]^{1/2}.$$

В качестве критерия согласования модели с множеством точек может использоваться и критерий максимального отклонения:

$$M = \max(\{|(f(x_j) - y_j)|\}_{j=1,n}).$$

Следует отметить, что данный критерий дает локальную характеристику соответствия модели, которую применять можно не во всех случаях.

Аналитический расчет параметров модели прямой по методу наименьших квадратов можно выполнить следующим образом.

Ошибка рассогласования модели может быть вычислена как

$$\varepsilon(c_1, c_0) = \sum_{j=1}^n (c_1 x_j + c_0 - y_j)^2.$$

Тогда минимум этой функции находится в точке (c_1, c_0) , в которой выполняются равенства для частных производных

$$\frac{\partial \varepsilon}{\partial c_1} = 0; \quad \frac{\partial \varepsilon}{\partial c_0} = 0.$$

Вычислив эти частные производные, получим:

$$\frac{\partial \varepsilon}{\partial c_1} = \sum_{j=1}^n 2(c_1 x_j + c_0 - y_j) x_j = 0;$$

$$\frac{\partial \varepsilon}{\partial c_1} = 2 \left(\sum_{j=1}^n x_j^2 \right) c_1 + 2 \left(\sum_{j=1}^n x_j \right) c_0 - 2 \sum_{j=1}^n x_j y_j;$$

$$\frac{\partial \varepsilon}{\partial c_0} = \sum_{j=1}^n 2(c_1 x_j + c_0 - y_j) = 0;$$

$$\frac{\partial \varepsilon}{\partial c_0} = 2 \left(\sum_{j=1}^n x_j \right) c_1 + 2 \sum_{j=1}^n c_0 - 2 \sum_{j=1}^n y_j;$$

В матричной форме этот результат можно записать в следующем виде:

$$\begin{bmatrix} \sum_{j=1}^n x_j^2 & \sum_{j=1}^n x_j \\ \sum_{j=1}^n x_j & \sum_{j=1}^n 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_0 \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^n x_j y_j \\ \sum_{j=1}^n y_j \end{bmatrix}.$$

Решив эту систему уравнений, можно найти параметры наилучшей аппроксимирующей прямой. Иногда, после анализа полученного результата, выявляются точки, лежащие в стороне от идеальной прямой. В этом случае для улучшения результата отбрасывают такие точки, являющимися «выбросами», и повторяют расчет модели. Приближение вновь полученной модели к идеальной прямой может оказаться лучше, так как на нее не влияют точки-выбросы.

5.7.2. Подбор других моделей сегментов

Кроме подбора модели прямых и отрезков прямых могут выполняться аналогичные приемы для кривых, плоскостей и иных сегментов. В простейших случаях предполагается, что известны вид модели и набор исходных точек. В более сложных случаях, когда контурные цепочки могут быть представлены сочленением набора различных форм, применяют сегментацию кривых с последующим поиском моделей.

Реальные изображения могут содержать контурные сегменты в виде лент, углов и более сложные конфигурации. В зависимости от постановки задачи могут использоваться описанные выше алгоритмы и их модификации.

Заключение

Настоящее учебное пособие является первой частью методических материалов по дисциплине «Компьютерное зрение и 3d-графика». Вторая часть будет посвящена распознаванию образов, поиску изображений на основе содержания, обнаружению движения на двумерных изображениях, моделированию трехмерных сцен по двумерным изображениям, а также системам виртуальной реальности.

Библиографический список

1. Компьютерное зрение / Л. Шапиро, Дж. Стокман; пер. с англ. – М.: БИНОМ. Лаборатория знаний, 2006. – 752 с.
2. Визильтер Ю.В. и др. Обработка и анализ изображений в задачах машинного зрения: курс лекций и практических занятий. – М.: Физматкнига, 2010. – 672 с.
3. Цифровая обработка изображений / А.А. Лукьяница, А.Г. Шишкин. – М.: «Ай-Эс-Эс Пресс», 2009. – 518 с.
4. Форсайт, Дэвид А., Понс, Жан. Компьютерное зрение. Современный подход.: пер. с англ. – М.: Издательский дом «Вильямс», 2004.
5. Haralick, R. M. 1974. A measure of circularity of digital figures. IEEE Trans. Syst., Man, and Cybern., v. SMC-9:62-66.
6. Shi, J., and J. Malik. 1997. Normalized cuts and image segmentation. IEEE Conf. Comput. Vision and Pattern Recog., 731-737.

Учебное издание

Селянкин Владимир Васильевич
Скороход Сергей Васильевич

**Анализ и обработка изображений
в задачах компьютерного зрения**

Ответственный за выпуск Селянкин В.В.
Редактор Надточий З.И.
Корректор Чеканенко Л.В.

Подписано к печати 20.12.2015
Заказ № 173 Тир. 50 экз.
Формат 60 x 84¹/₁₆. Печ. л. – 5,1. Уч.-изд. л. – 4,9.

Издательство Южного федерального университета
344091, г. Ростов-наДону, пр. Стачки, 200/1
Тел. (863) 2434166, 2434266
Отпечатано в Секторе обеспечения полиграфической
продукцией кампуса в г. Таганроге отдела полиграфической,
корпоративной и сувенирной продукции
ИКП КИБИ МЕДИА ЦЕНТРА ЮФУ,
ГСП 17 А Таганрог, 28, Энгельса, 1
Тел. (8634) 371717