

В

ДЛЯ ВУЗОВ

В.В. Янишин

АНАЛИЗ
И ОБРАБОТКА
ИЗОБРАЖЕНИЙ:
принципы
и алгоритмы

МАШИНОСТРОЕНИЕ

В

ДЛЯ ВУЗОВ

В.В. Яншин

АНАЛИЗ И ОБРАБОТКА ИЗОБРАЖЕНИЙ: принципы и алгоритмы

*Рекомендовано Государственным комитетом
Российской федерации по высшему образованию
в качестве учебного пособия для студентов высших
учебных заведений, обучающихся по направлениям
"Информатика и вычислительная техника",
"Радиотехника"*



Москва
• Машиностроение •
1995

ПРЕДИСЛОВИЕ

В настоящее время вопросами обработки изображений занимаются сотни, если не более организаций и предприятий стран содружества, причем большинство решаемых современных задач отличаются от традиционных задач обработки изображений. Эти отличия приводят к необходимости разработки методов и алгоритмов анализа (понимания) изображений.

Указанное обстоятельство, в свою очередь, приводит к необходимости создания специализированных (визуальных) баз данных, обеспечивающих как хранение большого количества изображений, так и быстрый (для дальнейшего манипулирования) доступ к ним. Создание баз данных в свою очередь диктует разработку методологии и выбора наиболее рациональных структур визуальных данных. Изложенное выше объясняет, почему возникла потребность в разработке, осмыслении и доступном изложении совершенно новой методологии и соответствующего алгоритмического обеспечения задач анализа (понимания) изображений и сопутствующих им факторов. Эта методология должна базироваться на том, что традиционные методы обработки изображений (а они в основном и изложены в имеющейся литературе) играют вспомогательную роль (являются алгоритмами "поддержки"). Основная же роль отводится вопросам анализа (понимания) изображений, создания визуальных баз данных и выбору наиболее рациональных структур видеоданных.

К настоящему времени имеется обширная литература по цифровой обработке изображений, причем, как правило, доминируют специализированные, объемные монографии, в которых в основном излагаются традиционные вопросы обработки изображений [10, 14, 31, 36, 39, 40, 56].

Указанные монографии трудны для восприятия их студентами. К сказанному следует добавить, что перечисленные книги пользуются большим спросом и представляют поэтому библиографическую редкость.

Другую, гораздо меньшую по объему группу, представляют собой монографии, в которых изложены (наряду с традиционными) и вопросы анализа (понимания) изображений [14, 31, 47, 57]. Причем и эта группа монографий практически недоступна студентам.

Таким образом, к настоящему времени возникла объективная потребность в издании учебного пособия по изложенной тематике. Настоящая книга и призвана в определенной степени удовлетворить указанную потребность.

ВВЕДЕНИЕ

В настоящее время вопросами переработки изображений занимается большое количество предприятий и организаций стран содружества. Часть из стоящих перед исследователями задач решается уже ставшими традиционными методами предварительной переработки изображений (предварительная обработка, сегментация изображений и распознавание образов).

Другая часть задач требует для своего решения нетрадиционной методологии и алгоритмов: это задачи, ориентированные на анализ сцен (понимание изображений).

Другими словами, системы понимания изображений (анализа сцен) можно считать очередным этапом в развитии так называемых визуальных информационных систем, принципы построения которых изложены в замечательной книге специалиста из США Ш.К.Чена [47].

При разработке систем понимания изображений несколько нарушается общая методология принципов переработки видеоданных (изображений). Традиционные методы предварительной обработки изображений (на более ранних этапах игравшие основную роль) уступают место новым задачам, таким как определение пространственных соотношений, выбор структуры видеоданных, сжатие видеоданных с целью их экономного хранения, создание визуальных баз данных, баз знаний и т.д. При этом традиционные методы переработки видеоданных играют роль "алгоритмов поддержки".

В настоящее время имеется обширная русскоязычная литература по цифровой обработке изображений, причем, как правило, доминируют специализированные объемные монографии, в которых в основном излагаются традиционные вопросы обработки изображений [10, 14, 31, 36, 39, 40, 56].

Перечисленные книги пользуются большим спросом и представляют собой библиографическую редкость.

Другую, еще большую по объему группу, представляют собой монографии, посвященные вопросам распознавания образов [11, 12, 14, 17], причем и эта группа книг, как правило, недоступна читателям. Последнюю, малочисленную и также недоступную группу книг представляют собой монографии, в которых изложены (наряду с традиционными) и вопросы анализа сцен [14, 31, 57].

Таким образом, книжный рынок изданий по вопросам цифровой переработки изображений испытывает потребность в русскоязычной книге по проблемам изложения принципов построения и алгоритмического обеспечения систем понимания изображений, даже небольшой по объему. Предлагаемая книга и призвана в определенной мере удовлетворить указанную потребность.

Содержание книги (и это отражено в ее названии) носит методологический характер и посвящено изложению основных методов (принципов) и некоторых алгоритмов, обеспечивающих понимание изображений.

Книга ориентирована на широкий круг читателей (от интересующихся до специалистов, уже работающих в области переработки видеоданных).

Ограниченный объем книги определяет стиль изложения (в книге мало формул, совсем нет графиков, а есть только иллюстрирующие рисунки). Поэтому (а также учитывая ограниченный объем) в книге не рассмотрены машинные аспекты обработки видеоданных: не описаны языки, основы программного обеспечения и их сопряжения с архитектурой процессоров переработки видеоданных.

Устройства переработки (персональные компьютеры, специализированные процессоры и т.д.) в книге названы условно машинами. Такие термины, как "изображение" и "сцена" считаются синонимами. В основном рассмотрены вопросы переработки полутоновых изображений, меньше рассмотрены вопросы обработки бинарных изображений, и совсем мало рассмотрены вопросы обработки изображений, представленных в виде точек (гл. 7).

Как указывалось ранее, в издательстве "Мир" выходит (или уже вышла) книга Ш.К.Чена [47] "Принципы проектирования систем визуальной информации". Автор стремился написать книгу, минимально совпадающую по содержанию с указанной.

Поэтому в книгу, например, не вошли такие вопросы, как процедуры получения визуальных знаний, процедуры индексирования и абстракции изображений и многое, что связано с понятием *икон* [47].

Следует отметить также, что в задачах цифровой обработки изображений многие этапы совпадают по своим методам и алгоритмам, и об этом часто упоминается в тексте.

Структура книги определяется тем, что главным в ней являются методы и алгоритмы понимания изображений (определения пространственных соотношений), а остальные этапы переработки видеоданных являются "алгоритмами поддержки".

1.1. ЗАДАЧИ ОБРАБОТКИ ИЗОБРАЖЕНИЙ

В настоящее время во многих областях науки и техники ощущается потребность в переработке видеоинформации, т.е. информации, которая может быть представлена в виде функции двух, трех и более переменных, причем две из перечисленных переменных являются пространственными координатами.

Видеоинформация представляет собой зафиксированное на каком-либо носителе изображение реального мира, (полученное с помощью каких-либо датчиков) таким, каким его видит глаз человека. Отсюда основные свойства видеоинформации: наглядность, избыточность и компактность [17, 18, 48].

Видеоинформация должна отражать реальные свойства двух- и трехмерных реальных изображений, а также те искажения, которыми обязательно сопровождается процесс получения видеоинформации реальными датчиками. Действительно, коль скоро в процессе получения видеоинформации присутствует датчик, то это с неизбежностью ведет к искажению реальных изображений, вызванному по крайней мере двумя причинами: конечной разрешающей способностью самого датчика и наличием разного рода помех (шумов).

Все сказанное относится не только к исходному, аналоговому изображению, но и к его цифровой форме, причем сам процесс представления аналогового изображения в цифровой форме (процессы квантования и дискретизации) приводит к дополнительному искажению [31, 27].

Мы рассматриваем процедуры обработки и анализа только двумерных изображений (полутоновых и бинарных), представленных в цифровой форме в виде числовой матрицы $\|s_{ij}\|$ ($i=1,\dots,M$, $j=1,\dots,N$), каждый элемент которой может принимать 2^L значений (уровней). Обычно $1 \leq L \leq 8$. Каждое из 2^L значений будем называть яркостью данного, исходного изображения в заданной точке s_{ij} (пикселе).

Вначале под обработкой изображений понимались некоторые преобразования, улучшающие его качество, причем эффективность преобразования (улучшение качества) определялась визуально человеческим глазом [2].

В процессе развития теории и практики работы с изображениями под обработкой изображений стали понимать такие преобразования, которые приводят изображение к виду, удобному для дальнейшей их обработки с помощью машин (подчеркивание контуров, изменение диапазона яркостей и т.д.) [2]. В указанных выше случаях в результате преобразований мы все равно получаем изображение.

В настоящее время термин (уже устоявшийся) *обработка изображений* указывает на такую его обработку, в результате которой вновь получается изображение [11, 27].

Сказанному эквивалентен англоязычный термин *Image Processing*. Иногда указанную обработку изображений называют *предварительной обработкой*.

Задачи предварительной обработки изображений включают в себя [31]: подавление шумов (низкочастотная фильтрация); изменение диапазона яркостей; коррекция геометрических искажений; преобразование гистограмм; подчеркивание (выделение) перепадов яркостей и границ; компенсация искажений; высокочастотная и нелинейная фильтрация; медианная и процентильная фильтрация и т.д.

Если известна математическая модель искажений изображений, то компенсацию искажений называют *реставрацией*. Примером задачи реставрации (кроме упомянутой коррекции геометрических искажений) является восстановление расфокусированных или смазанных (за счет движения датчиков) изображений [31].

Некоторые методы и алгоритмы предварительной обработки изображений рассмотрены в гл. 4. Чаще всего исследователей интересует изображение не само по себе, а та информация, которая в нем содержится. Извлечение этой информации и составляет цель дальнейшей (после предварительной) обработки видеоинформации (изображений).

Традиционно дальнейшая обработка изображений состоит из сегментации изображений и обнаружения и распознавания обнаруженных объектов. Последний этап составляет достаточно хорошо разработанную область науки – *классификацию и распознавание образов* [10–12, 16–18, 33, 38–43, 47].

Следует заметить, что при классическом распознавании образов предполагается, что уже имеется исходная информация для распознавания – вектор признаков. В обработке изображений измерение вектора признаков (не всегда известно, как его выбрать) – самостоятельная, далекая от завершения проблема.

Отметим, что результаты сегментации изображений (см. гл.5) и тем более результаты распознавания (см. гл. 7) носят все более "неизобразительный характер".

Упомянем еще об одной (в достаточной мере самостоятельной) проблеме – проблеме кодирования и сжатия изображений. Указанная проблема не нова для задач обработки изображений, и первоначально предназначалась для различного рода связанных задач (передача изображений по каналам связи). Однако в последнее время указанная проблема приобрела новое звучание в связи с разработкой различных визуальных информационных систем [47]. При этом резко обострилась проблема хранения большого количества изображений, отсюда и целесообразность сжатия видеоданных. Проблема кодирования и сжатия видеоданных (изображений) обсуждается в гл. 6.

1.2. ЗАДАЧИ АНАЛИЗА ИЗОБРАЖЕНИЙ

Термин *анализ изображений* равно как и термин *анализ сцен* к настоящему времени не является устоявшимся. В различной литературе в этот термин вкладывают и различный смысл. Но в любом случае, под анализом изображений понимают выделение из изображений неизобразительной информации. Сказанному эквивалентен англоязычный термин *Image Analysis*. Напомним, что и результаты процедур сегментации и распознавания образов носят все более "неизобразительный" характер.

В [2] под анализом изображений понимают следующие задачи: 1) выделение признаков на изображении; 2) обнаружение и распознавание объектов (в частности букв в тексте для создания читающих автоматов); 3) выделение и идентификацию текстур (в основном для обработки аэрофотоснимков); 4) анализ сцен при создании систем технического зрения для роботов.

Содержание четвертой задачи в [2] не расшифровывается; в отношении первых трех задач очевидно можно сказать, что

они относятся либо к классификации и распознаванию образов, либо частично к задаче сегментации (выделение текстур).

В [14] под анализом сцен понимаются следующие задачи: 1) задачи, позволяющие получать такие методы описания сцены, с помощью которых машины способны давать ответы на вопросы типа "изображен ли на картинке стул?" или выполнять команды типа "опиши сцену"; 2) задачи следующего типа: опознавание знаков; обнаружение определенных событий путем анализа треков на фотографиях, биомедицинский анализ препаратов, изучаемых под микроскопом.

Первые две задачи из второго типа также, на наш взгляд, относятся к задачам классификации и распознавания образов. О последней задаче второго типа и о задачах первого типа поговорим чуть позже.

В [17] под анализом изображений понимается три основных класса задач: 1) распознавание объектов; 2) содержательная интерпретация изображений; 3) получение справочной информации в архивах видеоданных.

Относительно первого класса задач мы уже говорили (он относится, на наш взгляд, к задачам классификации и распознавания образов). Второй и третий класс задач (особенно, если запрос на получение справочной информации представляется в виде описания) практически совпадает с первым классом задач из [14].

Наряду с терминами анализ изображений и анализ сцен существует близкий к ним термин – *распознавание изображений*, под которым понимается следующее [11, 18]: получение описания заданного изображения; отношение предъявленного изображения к одному из заданных классов.

Вторая задача, на наш взгляд, все же ближе к задачам классификации и распознавания образов, с той лишь особенностью, что число возможных классов настолько велико, что задача практически становится нерешаемой [31]. Первая же задача (аналогично второму и третьему классу задач из [17]) совпадает с первым классом задач из [14].

В [31] под анализом изображений фактически подразумеваются системы понимания изображений, целью которых является составление описания изображения (сцены) в "неизобразительной" форме.

Задачи [31], позволяющие получать такие методы описания, с помощью которых можно давать ответы в "неизобразительной" форме (первый класс задач из [14]), будем называть *задачами понимания изображений*.

В [57] дано следующее понятие анализа изображений. Суть этого понятия заключается в следующем. Рассмотрим пример обработки находящихся на изображении группы солдат.

Человеческая система глаз – мозг легко выстраивает группу солдат по росту (не измеряя его!). В то же время для системы глаз–мозг трудно оценить фактический рост каждого солдата (ранжирование солдат по росту в человеческой системе глаз – мозг происходит без точного измерения роста каждого солдата). Дело обстоит иначе, если эту же задачу ранжирования поручить технической системе обработки и анализа изображений. В этом случае процедура ранжирования будет происходить в обратном порядке: сначала (после сегментации) происходит измерение роста всех солдат, и лишь затем возможно их ранжирование (или, другими словами, классификация солдат по единственному признаку – величине роста).

Таким образом, согласно [57] под анализом изображений понимается техническая система, производящая на изображении измерение количественных данных (длины, периметра, координат местоположения, размеров и т.д.), на основе которых и делается соответствующий вывод об изображении (проводится анализ изображения). В [57] под анализом изображений понимается также построение гистограммы ингредиентов предъявленного образца материала.

В качестве другого примера анализа изображений, где измерения играют основную роль, в [57] рассмотрен другой случай: оценивание качества металла по средней площади различных зерен, составляющих микроструктуру металла. Одним из самых быстрых методов оценивания средней площади заключается в выделении всех черных областей на фотографии микросреза и подсчете их площади, поделенной на общую площадь. Результат такого измерения и определяет качество материала, причем результат анализа выдается в логической форме (да или нет).

Нетрудно видеть, что описанные примеры из [57] совпадают по сути с биомедицинским анализом препаратов, изучаемых под микроскопом (третья из второго класса задач из [14]).

Таким образом, в [57] под термином *анализ изображений* понимается измерительный аспект обработки видеоданных. Следовательно, термин анализ изображений по данным литературных источников несет тройственный смысл: 1) классификация и распознавание образов; 2) измерение различных количественных данных изображения; 3) смысл задач понимания изображений. Термин анализ изображений будет интерпретироваться нами как класс задач понимания изображений.

Основу задач понимания изображений составляют описания изображений. Под описанием изображений (сцен) будем понимать более простое, чем само изображение*, представление исходного изображения в некоторой "неизобразительной" форме: от локализации простейших выделенных и распознанных (классифицированных) объектов вплоть до содержательной интерпретации всего изображения либо в форме графовых моделей, либо в словесной форме. Под последними понимается в том числе и подробное перечисление находящихся на изображении объектов с указанием, в каких отношениях эти объекты находятся между собой.

Системы понимания изображений должны (в результате переработки видеоинформации) давать (в неизобразительной форме) ответы на два типа вопросов: логических (имеется ли озеро такой-то площади юго-западнее дома?); количественных (сколько озер находится юго-западнее дома и на каком расстоянии?).

Естественно, для ответа на эти вопросы необходимо предварительно выделить (и классифицировать) все озера, имеющиеся на изображении (равно как и не озера) и все дома (равно как и не дома). Другими словами, под анализом изображений (пониманием изображений) мы подразумеваем такую переработку видеоинформации (изображений, сцен), которая позволяет определить (качественно и количественно) пространственные соотношения между всеми объектами, интересующими исследователя (пользователя) на исходном изображении.

Таким образом, все этапы переработки информации, предшествующие этапу понимания изображений (предварительная обработка, кодирование и сжатие, сегментация, а также классификация и распознавание образов), должны быть организованы так, чтобы наилучшим образом решалась конечная

* Иначе само описание теряет смысл.

задача – задача понимания изображений (определение пространственных соотношений) Например, если ставится задача определения пространственных соотношений между озерами, имеющимися на исходном изображении, с площадью, значительно превышающей площадь домов, то вопросы сегментации и классификации домов не возникают и объекты с малой площадью могут быть отселектированы на этапе предварительной обработки (или сегментации), например по бинарной копии исходного изображения, по алгоритмам, изложенным в [50]

В рассматриваемом случае все классы алгоритмов, предшествующих алгоритмам понимания изображений, могут быть названы "алгоритмами поддержки"

1.3. ДРЕВОВИДНЫЕ МОДЕЛИ ИЗОБРАЖЕНИЙ

Древовидные модели изображений относятся к иерархическим моделям, показывающим связь между исходным изображением и его крупными частями, между крупными частями и его мелкими частями и т.д. Конечность "разбиения" исходного изображения определяется существом задачи, т.е. степенью детализации анализа изображения. Приведем пример получения древовидных моделей изображения [31].

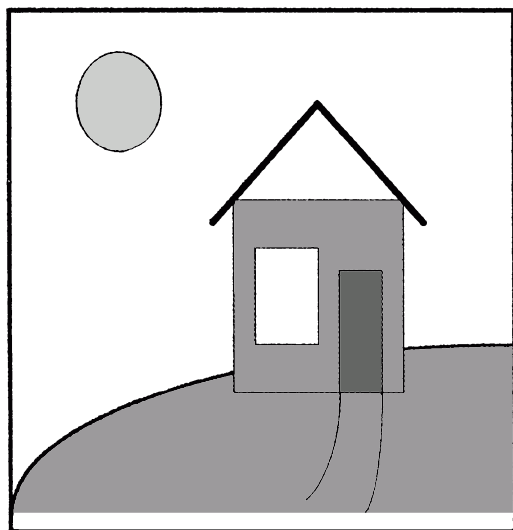


Рис. 1.1. Исходное изображение

На рис. 1.1. изображена исходная сцена (графический рисунок) На изображении находятся "фоновые объекты" (небо и земля) Кроме фоновых объектов еще имеются дом и солнце. На объекте дом имеется дверь и окно, от двери дома по земле проложена тропинка.

Фактически мы произвели словесное описание сцены, изображенной на рис. 1.1.

Модель изображения на рис. 1.1 в виде дерева представлена на рис. 1.2 и не требует комментариев [31]. Корень дерева – это исходная сцена, вершины (уровни) – это части исходной сцены. Ясно, что сцена, изображенная на рис. 1.1, может быть представлена (если это диктуется задачей анализа) более подробно: могут быть изображены дверная ручка, форточка, труба с дымом (или без) и т.д. При этом составление древовидной модели по аналогии с рис. 1.2 выполняется легко.

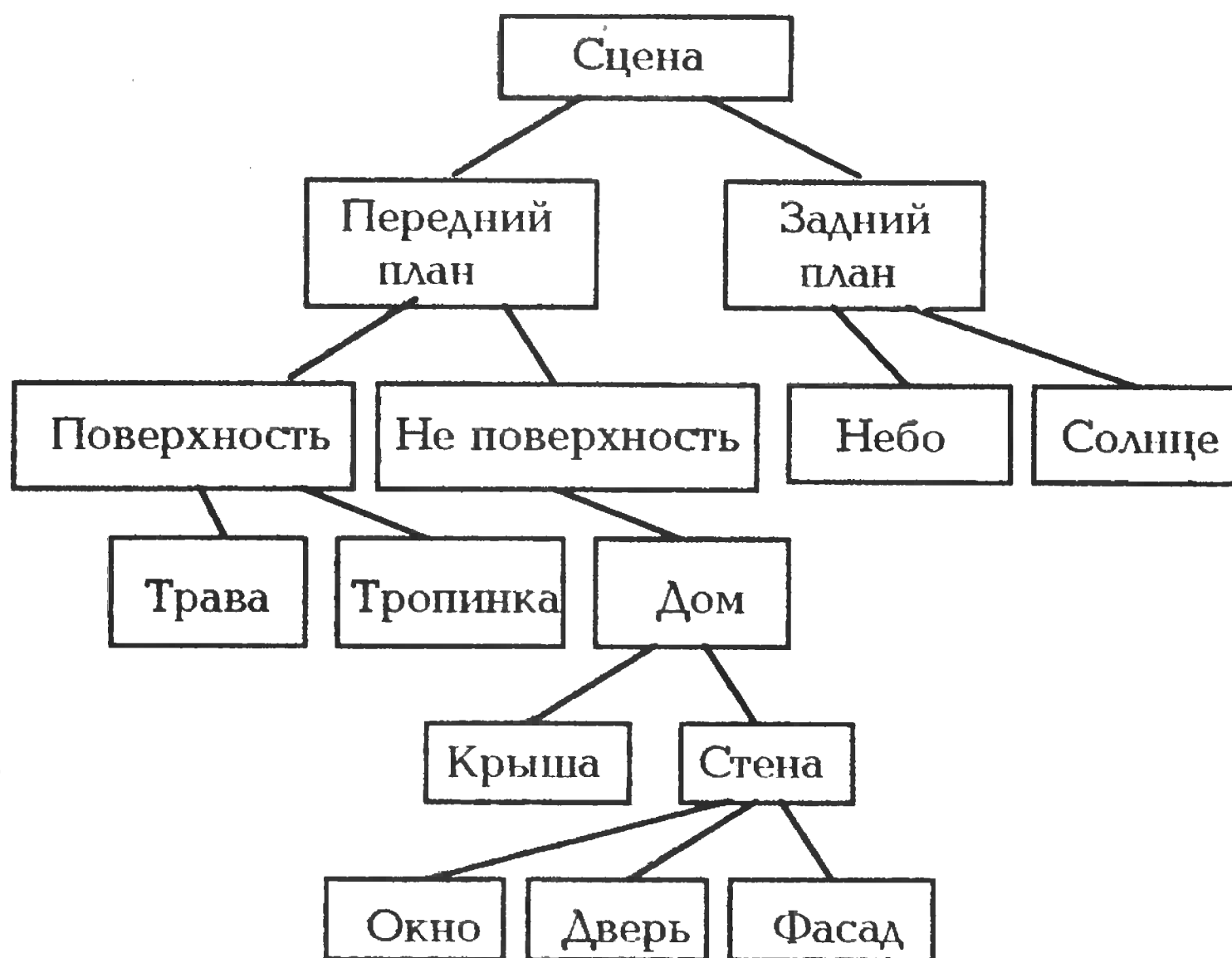


Рис. 1.2. Древовидная модель сцены, изображенной на рис. 1.1

1.4. ГРАФОВЫЕ МОДЕЛИ ИЗОБРАЖЕНИЙ

Древовидные модели, описанные в предыдущем разделе, дают лишь грубое описание исходных изображений: в древовидных моделях указаны лишь связи между двумя элементами изображения (солнце "связано" только с небом, дверь "связана" только с домом и т.д.).



Рис. 1.3. Графовая модель сцены, изображенной на рис. 1.1

Иногда бывает нужна и другая, более подробная информация о связях (отношениях) между элементами исходного изображения. Такие отношения (связи) можно отразить с помощью направленного (есть стрелки) графа отношений. На рис. 1.3 представлена модель сцены, изображенной на рис. 1.1, в виде направленного графа отношений.

В указанной модели ребра (дути) графа, связывающие элементы исходного изображения, размечены по виду связи и пространственных отношений между элементами исходного изображения.

Таким образом, графовые модели (в виде направленного графа отношений) представляют собой более богатые описания по сравнению с древовидными моделями.

В [34] знания об отношениях между элементами исходного изображения были использованы для построения систем понимания изображений. Более подробно об этом изложено в [31].

2.1. ОСНОВНЫЕ ПОНЯТИЯ

Для задач хранения большого количества изображений, характеризующих крупными размерами, и соответственно требующих значительных объемов памяти, проблемы, связанные с хранением, поиском и манипуляцией изображениями, становятся одними из основных. Следовательно, возникает задача такого представления или описания изображений, чтобы по возможности (а эти требования противоречивы) они были адекватны исходным изображениям, их удобно было бы обрабатывать и интерпретировать, и в то же время требовался сравнительно небольшой объем памяти для хранения указанных представлений и описаний. Под структурой видеоданных будем понимать либо изображения, чем-то отличающиеся от исходных, либо описание изображений с указанием соотношений между частями описанного изображения [1].

Очевидно, что описаний у одного и того же изображения существует несколько: все диктуется содержательной частью задачи. Для задачи понимания изображений описания и соответствующие структуры данных должны отражать пространственные соотношения между частями исходной сцены, выделенными (обнаруженными, сегментированными и распознанными) объектами, их частями и т.д.

При этом при составлении описаний следует иметь в виду (и на это неоднократно указывалось в [1, 11, 48]), что изображения как таковые обладают весьма большой избыточностью – статистической, психовизуальной и содержательной*, что вселяет известный оптимизм в возможность решения проблемы

* Так как в книге рассматриваются только вопросы машинной переработки видеоданных, то вопросы устранения психовизуальной избыточности здесь не рассматриваются. Некоторые вопросы устранения статистической и содержательной избыточности рассматриваются в гл.6.

разработки экономных структур данных. И здесь, в первую очередь, необходимо избавляться от содержательной избыточности, под которой понимается то, что не все объекты, имеющиеся на изображении, и их соотношения представляют интерес в каждой конкретной задаче [11, 48].

Например, при обработке запроса "найти все дома юго-западнее озера А" нас не интересуют элементы, находящиеся севернее и восточнее озера А. Также нас не интересуют все соотношения между остальными объектами (как, впрочем, и сами объекты), имеющимися на сцене.

Пример показывает, что при разработке структур данных, необходимо ориентироваться на такие манипуляции с данными, которые порождаются перечнем потенциальных запросов.

В одном крайнем случае будет одна, но очень громоздкая структура данных, во втором, также крайнем случае будут храниться несколько (а возможно и много) структур данных, но каждая структура данных будет относительно проста.

2.2. КЛАССИФИКАЦИЯ СТРУКТУР ДАННЫХ

В самом общем случае структуры данных классифицируются на две большие группы [48]: позиционное представление данных и структурное представление данных. Конечно, когда это определяется существом задачи, возможно объединение обеих групп данных (комбинированные структуры данных).

В случае позиционного представления данных все изображение (т.е. матрица) разбивается на одинаковые квадраты, которые для рассматриваемой задачи считаются неделимыми. В качестве "яркости" этого квадрата берется какая-либо характеристика данных, составляющих квадрат (например, среднее значение) [48]. Позиционное представление данных бывает матричным, пирамидальным и с помощью дерева квадрантов [18, 48]. Основным недостатком позиционного представления является требование большого объема памяти.

Структурное представление данных основано на описании исходных изображений, ориентированное на сокращение содержательной избыточности, что приводит и к уменьшению требуемого объема памяти, и к уменьшению времени обработки запросов. Заметим, что одной из основных проблем в теории распознавания образов (см. гл. 7) является проблема определения и извлечения (измерения) признаков.

Поэтому и изображения (точнее выделенные и классифицированные объекты) могут быть описаны их признаками: при этом также достигается значительное сжатие исходного изображения, и как следствие уменьшение требуемого объема памяти. Следовательно, само структурное представление видеоданных является сжатым (см. гл. 6) изображением и является частью решения задачи распознавания, так как при этом происходит описание с помощью признаков (см. гл. 7).

Структурное представление данных исходит из того, что исходное изображение состоит из объектов и соотношений между ними. Объекты в свою очередь (при структурном представлении) можно представить как соединение непримитивных элементов (см. п. 7.7), либо как соединение составляющих его геометрических элементов (линия, точка, плоскость и т.д.) – это с одной стороны. С другой стороны, каждый объект можно описать через его форму или контур.

Оба подхода весьма близки к друг другу, так как, например, в качестве непримитивных элементов (см. гл. 7) можно взять либо геометрические элементы объекта, либо элементарный вектор при кодировании контуров по Фримену, или при комплекснозначном кодировании по Фурману.

При структурном представлении данных всегда необходимо исходить из семантических соображений (существа, смысла решаемой задачи), которые и диктуют способ представления данных.

2.3. ПОЗИЦИОННОЕ ПРЕДСТАВЛЕНИЕ ДАННЫХ

2.3.1. Матричное представление данных

В этом случае изображение представляет собой матрицу $\|s_{ij}\|$ ($i=1,\dots,M$, $j=1,\dots,N$). В каждом элементе (квадрате) указанного плоского изображения находятся некоторые характеристики исходного изображения, например [48]: значения яркостей (например, среднее значение данных, составляющих квадрат); идентификаторы принадлежностей к классу имеющихся объектов (символьные изображения); ссылки к табличной записи и т.д. В зависимости от размеров указанных элементов изображения (квадратов) матричное представление данных может быть растровым и клеточным.

При растровом представлении данных размер элемента изображения определяется разрешающей способностью датчика изображения.

При клеточном представлении данных размер элемента изображения больше, чем разрешающая способность датчика, и определяется исходя из требуемой точности представления изображения. Ясно, что клеточное представление данных требует меньшего объема памяти, чем растровое представление. Основное достоинство матричного представления в виде значения яркостей – это сохранение структуры исходного изображения (пространственной организации яркости и цвета – если изображение цветное). Иногда такое представление видеоданных называют прямым, и оно позволяет легко реализовать последовательную или параллельную обработку всего изображения, причем чаще используется локальная фильтрация данных (см. гл. 4) [18].

2.3.2. Пирамидальное представление данных

Пирамидальное представление данных (в [11] это представление именуется тетрарным деревом) представляет собой иерархический набор матриц яркостей одного и того же изображения. Другими словами, исходное изображение описывается упорядоченной последовательностью нескольких изображений различного разрешения, сходящихся к исходному и обычно располагаемых одно под другим. Отсюда и название термина "пирамида" [2, 18, 27, 48]. Каждое из изображений, расположенных в пирамиду, называют уровнем (слоем) пирамиды r_i ($i=0, \dots, L$); где нулевой уровень – исходное изображение, а L -й – изображение размером в один пиксел.

Наиболее часто исходное изображение разбивается на равные квадраты, возможно пересекающиеся. В качестве элемента квадрата принимают среднее значение яркостей составляющих его пикселей. Однако это не обязательно: можно брать любую функцию, включая медиану и другие порядковые статистики.

Для хранения всей пирамиды требуется емкость памяти приблизительно на 33% большая по сравнению с требуемой емкостью памяти для хранения исходного изображения [27].

Хранение и восстановление любого уровня пирамиды осуществляется следующим способом [27]. Пусть четыре пиксела, образующие квадрат, имеют следующие значения яркостей:

s_0, s_1, s_2, s_3 Пусть элемент квадрата образуется как среднее значение от исходных уровней яркости, т.е. $g_0 = \sum_{i=0}^3 s_i$.

Определим еще три числа $g_i = s_i - g_0$ ($i=1,2,3$). Полученные числа должны храниться в специальном массиве. При этом никакой потери информации не происходит, т.к. с помощью выражения $s_i = g_i + g_0$ ($i=1,2,3$) и $s_0 = g_0 - \sum_{i=1}^3 g_i$ восстанавливается предыдущий уровень пирамиды.

Смысл пирамидального представления данных заключается в том, что поиск (обнаружение, выделение) объектов происходит на верхних уровнях пирамиды (с низкой разрешающей способностью), и в случае необходимости осуществляется переход на более низкие уровни пирамиды (с более высоким разрешением). В целом повышается эффективность обработки изображений. В [48] указывается, что при этом достигается уменьшение времени подчеркивания перепадов яркости на 45 – 64%. Заметим, что пирамидальное представление данных обладает фильтрующими свойствами. Так, если в качестве нового элемента (квадрата) верхнего уровня брать среднее значение от четырех элементов более низкого уровня, то пирамидальное представление эквивалентно низкочастотной фильтрации.

Если в качестве нового элемента брать какую-либо порядковую статистику, то пирамидальное представление данных обладает свойствами процентильной и медиальной фильтрации (см. гл. 4).

2.3.3. Представление данных с помощью дерева квадрантов

Представление данных с помощью дерева квадрантов имеет смысл тогда, когда не все фрагменты исходного изображения содержат информацию, необходимую пользователю. Поэтому те фрагменты которые не содержат, необходимую пользователю информацию нецелесообразно хранить и обрабатывать.

Описываемая структура предписывает разделение исходного изображения последовательно на квадранты до тех пор, пока не останутся квадранты (возможно размером в один пиксел), состоящие только из нулей и единиц [48].

В результате указанная иерархическая структура имеет вид дерева, корень которого соответствует исходному изображению (как нижний уровень при пирамидальном представлении данных), а конечные вершины соответствуют квадрантам, не требующим конечного деления [48]. Дерево квадрантов можно построить как для бинарных, так и полутоновых изображений [48].

Приведем пример представления данных с помощью дерева квадрантов на примере исходного бинарного изображения размером 8x8, которое представлено на рис. 2.1. (заштрихованные пиксеты соответствуют единицам, а не заштрихованные – нулям) [48].

Цифрами (от 1 до 4) на рис. 2.1. обозначены квадранты минимального размера (в один пиксел); а буквами (от A до I) обозначены квадранты разных размеров, определяемых исходным изображением.

Квадрант D (состоящий только из единиц) и квадрант E (состоящий только из нулей) имеют размер 4x4. Остальные квадранты (A, B, C, F, G, H и I) имеют размер 2x2. Соответствующее рис 2.1 дерево квадрантов представлено на рис. 2.2 (пустые квадраты соответствуют квадрантам, состоящим только из нулей, заштрихованные квадраты соответствуют квадрантам, состоящим только из единиц. Кружки соответствуют квадрантам, состоящим как из нулей, так и единиц) В [48] приведены сведения, утверждающие, что для распознавания и интерпретации полутонового изображения достаточно хранить лишь 30% данных всего изображения (имеется в виду последний уровень структуры) В гл 6 (п 6.6) изложена суть структур в виде бинарных деревьев. Для представления видеоданных в машинах также широко используются различные преобразования, основная суть которых изложена в гл 6 (п. 6.5)

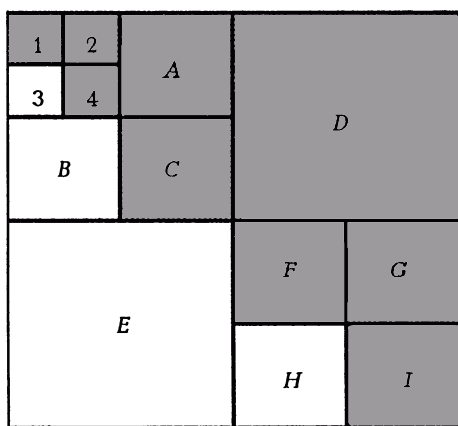


Рис. 2.1. Исходное бинарное изображение и его декомпозиция на блоки

2.4. СТРУКТУРНОЕ ПРЕДСТАВЛЕНИЕ ДАННЫХ

Структурное представление данных всего изображения может быть описано либо в терминах объектов и отношений между ними, либо в терминах формы каждого объекта [48]

Какие элементы изображения брать в качестве объектов, определяется сущностью задачи (см гл 1)

В целом структурное представление данных позволяет получать экономное описание изображений ("сжимать" их)

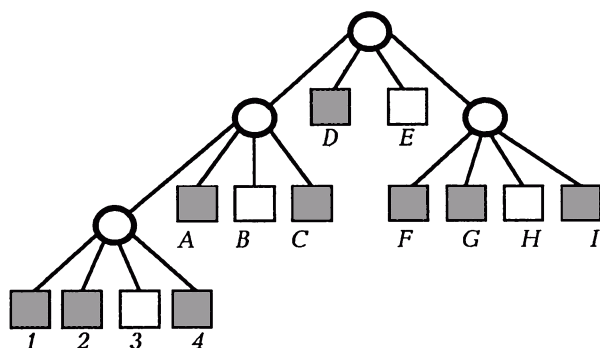


Рис. 2.2. Дерево квадратов

2.4.1. Представление формы объектов с помощью контуров

Существует несколько представлений формы объектов с помощью контуров (цепное кодирование, комплексно-значное кодирование, *P*-представление контура и т.д.) [45, 46, 48]

Рассмотрим цепное кодирование (с помощью кода Фримена), сущность которого заключается в том, что число направлений от рассматриваемой точки контура к следующей равно восьми, и эти направления кодируются числами от 0 до 7 (рис 2 3) [45]

Следовательно, любую кривую (включая и замкнутый контур) на исходном (или представленном с помощью позиционной структуры данных) изображении можно закодировать последовательностью чисел от 0 до 7 [14, 27, 31]

Цепное кодирование позволяет вычислять ряд характеристик формы, таких как длину кривой, площадь под кривой и т.д.

Обобщение цепного кодирования, осуществляемое путем представления изображения на плоскости комплексного переменного, можно назвать комплексно-значным кодированием (кодирование по Фурману) [46].

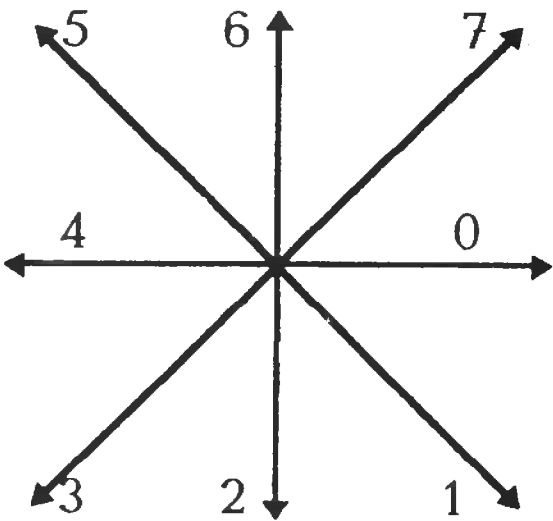


Рис. 2.3. Геометрическая иллюстрация цепного кодирования

В табл. 2.1 представлены соотношения между кодом Фримена и комплексно-значным кодом Фурмана и соответственно его мнимой (*Im*) и действительной (*Re*) частями ($i = \sqrt{-1}$).

Таблица 2.1

Элементы кода Фримена	0	1	2	3	4	5	6	7
Элементы кода Фурмана	1	1-i	-i	-1-i	-1	-1+i	+i	1+i
Im	0	-1	-1	-1	0	1	1	1
Re	1	1	0	-1	-1	-1	0	1

Переход от кода Фримена к коду Фурмана позволяет рассматривать контур как комплексный сигнал и, как следствие этого, применять хорошо разработанную теорию обработки одномерных сигналов для обработки двумерных сигналов [46].

При Р-представлении контуров в месте расположения объектов на исходном изображении строится ряд равноотстоящих сечений параллельно либо оси абсцисс, либо оси ординат [48]. В первом случае Р-представление имеет вид

$$P_k = \left[y_{\min}^k, y_{\max}^k, Y_1^k, Y_2^k, \dots, Y_{n_k}^k \right], \text{ где } y_{\min}^k (y_{\max}^k) - \text{минимальная (максимальная) ордината контура; } n_k - \text{число сечений,}$$

$$Y_i^k = (r_{ik}, x_1^{ik}, x_2^{ik}, \dots, x_{r_{ik}}^{ik}), \text{ где } r_{ik} - \text{число точек пересечения кон-}$$

тура. Достоинство Р-представления – это простота выполнения таких расчетов, как нахождение периметров и площадей; в то же время Р-представление не является экономным.

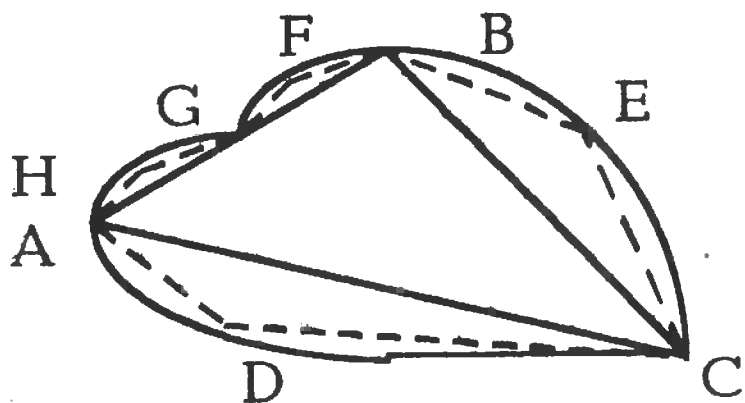


Рис. 2.4. Контур и аппроксимирующие его сегменты

Иногда форму протяженных объектов представляют с помощью скелетона [14, 48].

Контур, или незамкнутые кривые, можно аппроксимировать с помощью линейных сегментов, координаты концов которых запоминаются; при этом получается наиболее

полное описание контуров. Структурой, аппроксимирующей контур, может быть дерево. На рис. 2.4 представлен контур вместе с аппроксимирующими его сегментами.

На рис. 2.5 представлено дерево аппроксимации контура.

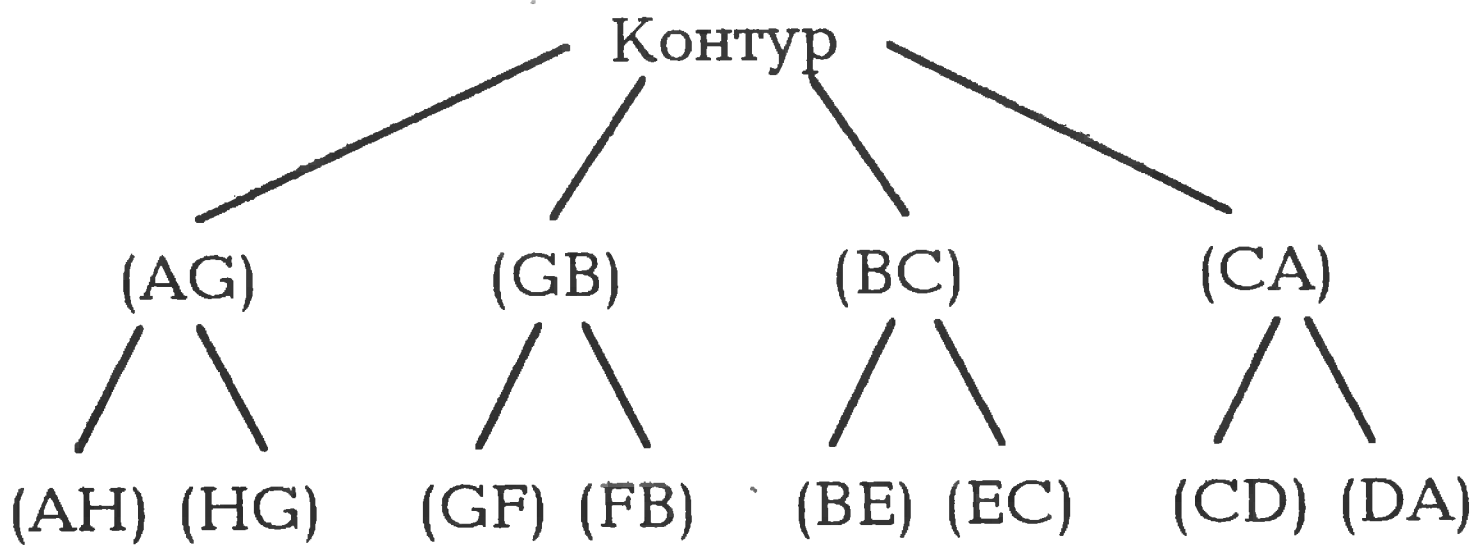


Рис. 2.5. Дерево аппроксимации

2.4.2 Аналитическое представление формы

Одним из наиболее удобных методов аналитического представления формы является представление формы как функции комплексного переменного [31]. В этом случае контур (как непрерывная замкнутая кривая) задается своей координатой $z(l)$, где l – текущая длина. Комплексная функция $z(l)$ представима в виде ее действительной $x(l)$ и мнимой $y(l)$ компонент, т.е. $z(l) = x(l) + iy(l)$. Угол наклона касательной

$$\Phi(l) = \left| \frac{dy(l) / dl}{dx(l) / dl} \right|.$$

Кривизна есть действительная функция $k(l) = d\Phi(l)/dl$. Так как кривизна есть периодическая функция с периодом, то эту кривизну можно разложить в ряд Фурье. Коэффициенты ряда Фурье называются Фурье-описанием контура [31]. Усеченный ряд Фурье применяется для распознавания объектов по его форме.

2.5. ПРЕДСТАВЛЕНИЕ ИЗОБРАЖЕНИЙ

Начнем с иерархического представления изображений (частично изложенное в гл.1), согласно которому исходное изображение разбивается на части, которые в свою очередь также разбиваются на части, причем уровень разбиения определяется существом задачи [48]. Структурой данных в рассматриваемом случае является дерево: 0-уровень (корень дерева) – это исходное изображение; остальные уровни (вершины) компоненты (части) исходного изображения и т.д.

На рис. 2.6 представлены исходное изображение в виде замкнутых контуров и отображение этого изображения в древовидную структуру.

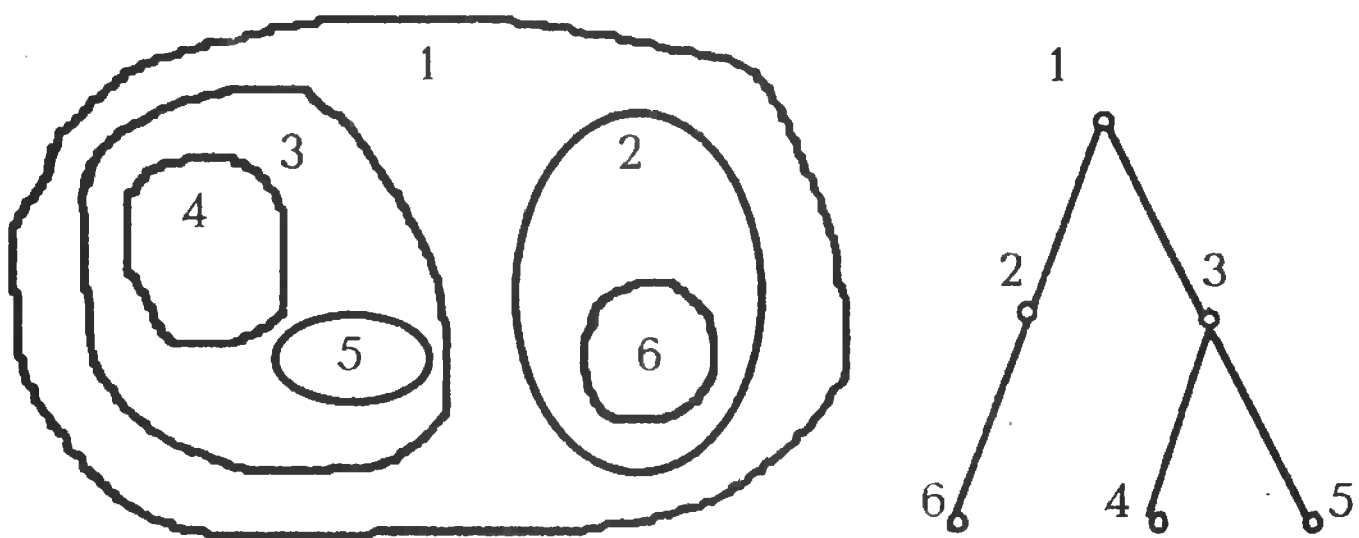


Рис. 2.6. Изображение контуров и отображение их в виде дерева

С помощью древовидных структур можно отразить и результат сегментации исходного изображения, что и проиллюстрировано на рис. 2.7.

Иерархическое представление используется и при описании сцен (см. гл.1). На рис. 2.8 изображено иерархическое представление рисунка.

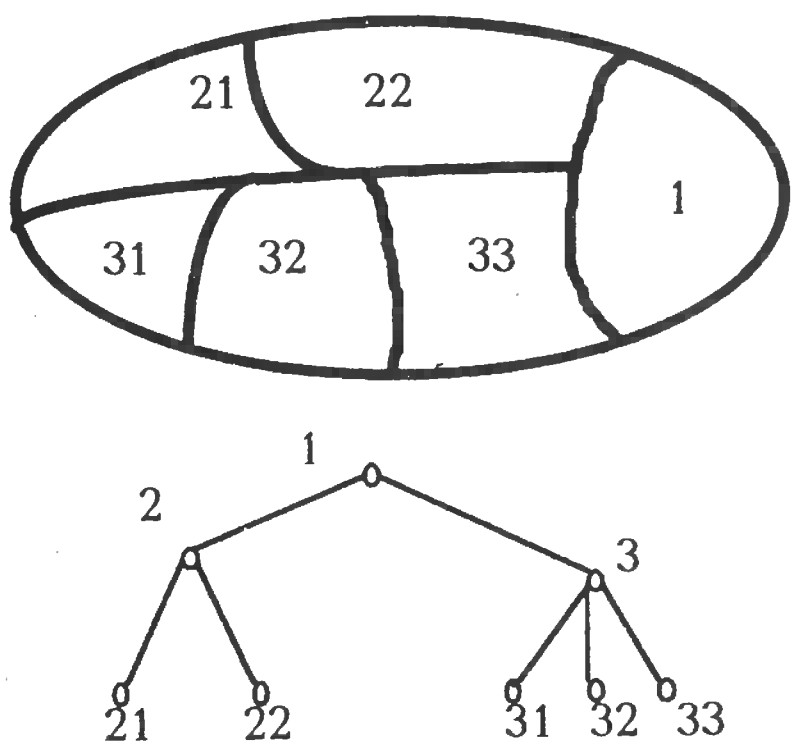
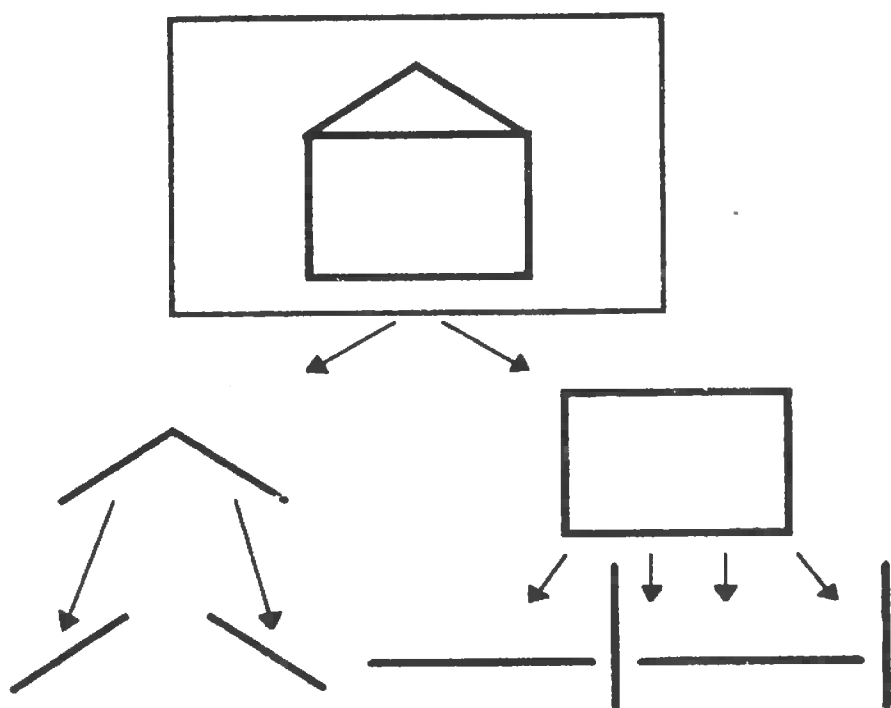


Рис. 2.7. Сегментированные области и их представление с помощью дерева

Заметим, что конечные элементы иерархического представления последнего рисунка – непроеизводные элементы, используемые для лингвистических методов распознавания образа (гл. 7).

Кроме древовидных структур используются и графовые представления данных (гл. 1), отображающие более сложные отношения (связи) между объектами, чем это возможно при древовидных представлениях. Представление изображений в виде графов необходимо и тогда, когда число отображаемых типов связей больше единицы. В этом случае имеем граф отношений.

В гл. 1 приведен пример, когда одна и та же сцена была представлена и древовидной, и графовой (граф отношений) структурами.



Кроме графа отношений изображение (сцена) может быть представлено в виде графа соседства [48]. На рис. 2.9 представлено сегментированное изображение и его представление в виде графа соседства.

Рис. 2.8. Иерархическое представление рисунка

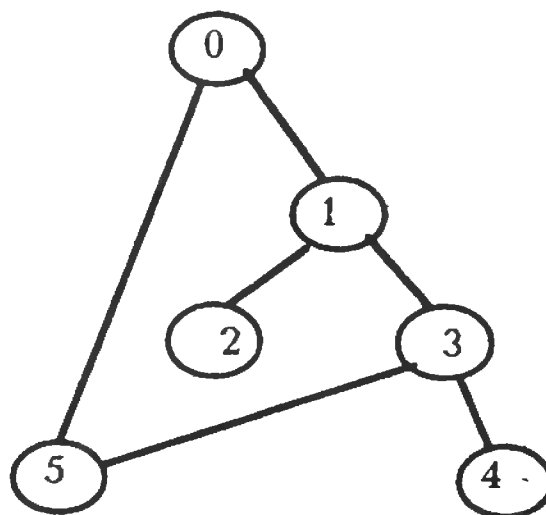
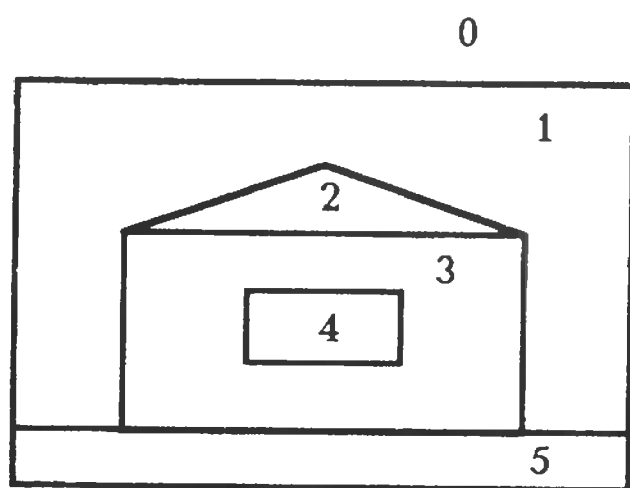


Рис. 2.9. Сегментированное изображение и его представление в виде графа соседства

И, наконец, скажем несколько слов о графовой структуре "область – сегмент – точка" [48]. Указанная структура представляет собой ориентированный граф, состоящий из трех слоев вершин. Слой O (области) состоит из вершин, представляющих области, ограниченные линейными сегментами. Второй слой C (сегменты) состоит из вершин, представляющих линейные сегменты, являющиеся границами областей. И, наконец, третий слой T (точки) представляет собой конечные точки линейных сегментов [48].

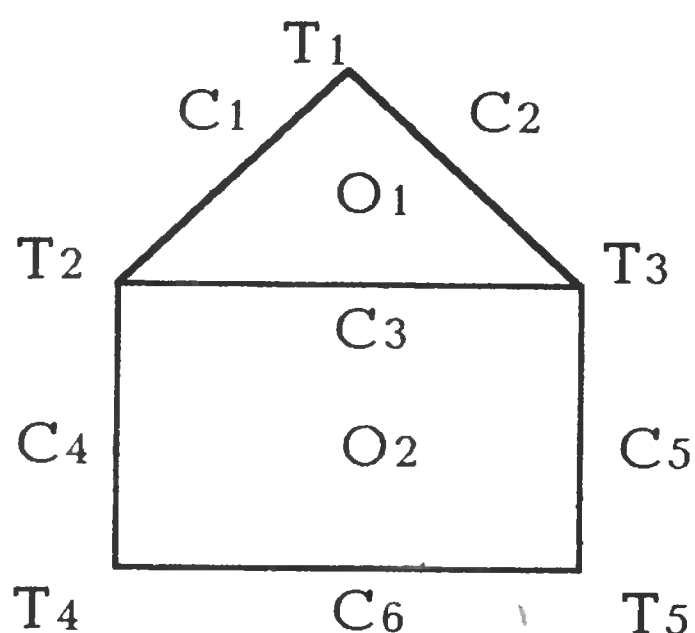


Рис. 2.10. Исходное сегментированное изображение

На рис. 2.10 представлено исходное сегментированное изображение $O\{O_1 \cup O_2\}$.

На рис. 2.11 представлена графовая структура "область – сегмент – точка" изображения $O\{O_1 \cup O_2\}$.

2.6. КОМБИНИРОВАННЫЕ ПРЕДСТАВЛЕНИЯ ДАННЫХ

Комбинированные представления данных позволяют объединить достоинства как позиционного, так и структурного представлений [48].

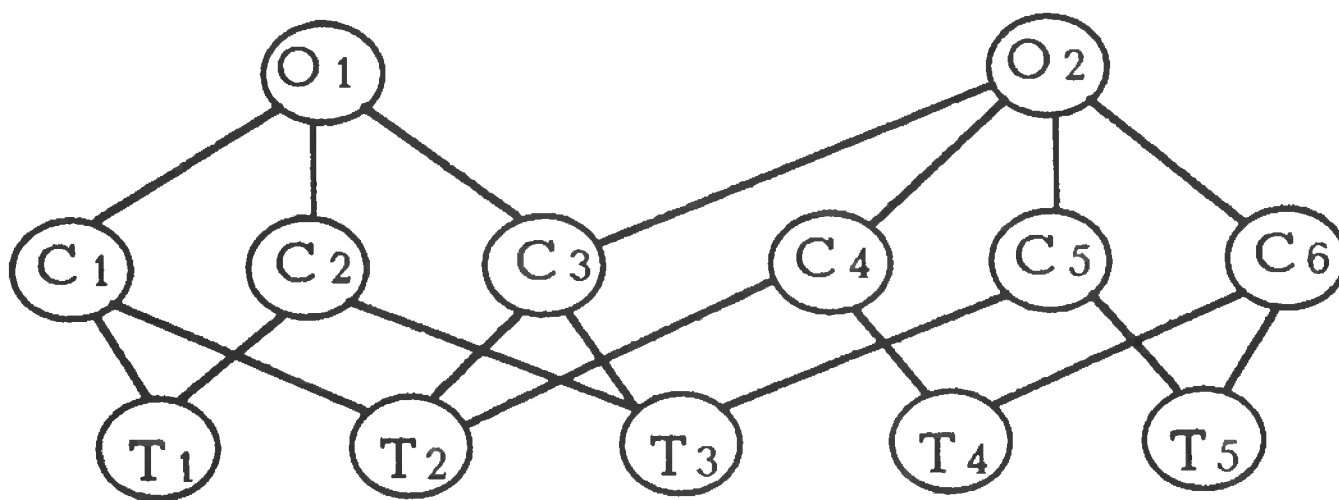


Рис. 2.11. Структура "область-сегмент-точка"

Эффективность представления изображения двумя способами заключается в том, что в зависимости от запроса пользователя обработку изображений целесообразно вести либо при его позиционном, либо при его структурном представлении [48].

2.7. РЕЛЯЦИОННАЯ МОДЕЛЬ ВИЗУАЛЬНОЙ БАЗЫ ДАННЫХ

При обработке и анализе изображений не последнюю (далеко не последнюю) роль играют проблемы экономного хранения большого количества изображений. Отсюда возникла необходимость выбора модели базы данных ("хранилища" большого количества данных).

Более подробно вопрос о различных базах данных, предназначенных для хранения (и поиска) больших массивов и их большого количества рассмотрен в [7].

Базу данных можно определить как унифицированную совокупность данных, которыми могут пользоваться большое количество независимых пользователей [7, 13, 20].

Три основных требования к базам данных (БД) [7, 13, 20]: 1) интеграция данных (все данные хранятся и накапливаются централизованно и их обновление (и уничтожение) также происходит централизованно; 2) максимально возможная независимость манипуляций с данными от самих данных; 3) по возможности (а в идеале и полностью) устранение избыточности в хранимых данных. Совокупность этих требований и привело к созданию собственно БД и к разработке одной управляющей программы для манипулирования данными — системы управления базой данных (СУБД).

Для задач хранения видеоданных чаще всего используются реляционные модели БД (еще существуют сетевые и иерархические модели [47, 49]). Сущность реляционной БД очень проста. Реляционная база данных представляет собой совокупность отношений (отсюда слово "реляционная"), содержащих всю необходимую информацию, которая должна храниться в БД [13, 25]. Под отношением здесь понимаются данные, представленные в табличной (очень удобной и наглядной) форме.

Для сопряжения понятий, встречающихся в разной литературе, приведем некоторые понятия, трактуемые "равноправным" образом [13]: 1) отношение, таблица, файл; 2) кортеж, строка, запись; 3) домен, атрибут, столбец и поле.

Ранее было показано, что структурное представление видеоданных (в древовидных и графовых моделях) представляют собой иерархическую структуру.

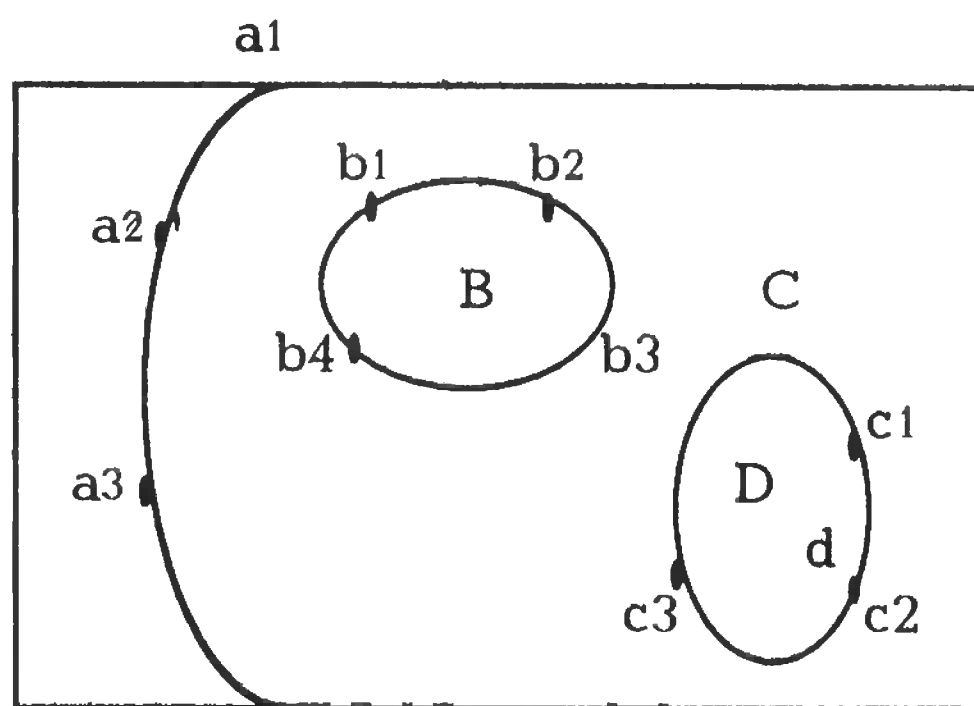


Рис. 2.12. Исходный рисунок

Наиболее полным является именно граф отношений, который в БД удобно представлять в табличном виде, откуда и следует предпочтительность представления видеоданных в реляционной БД. Приведем пример [49].

На рис. 2.13 представлен соответствующий граф отношений (на рисунке "ч" – часть, "с" – соединена с).

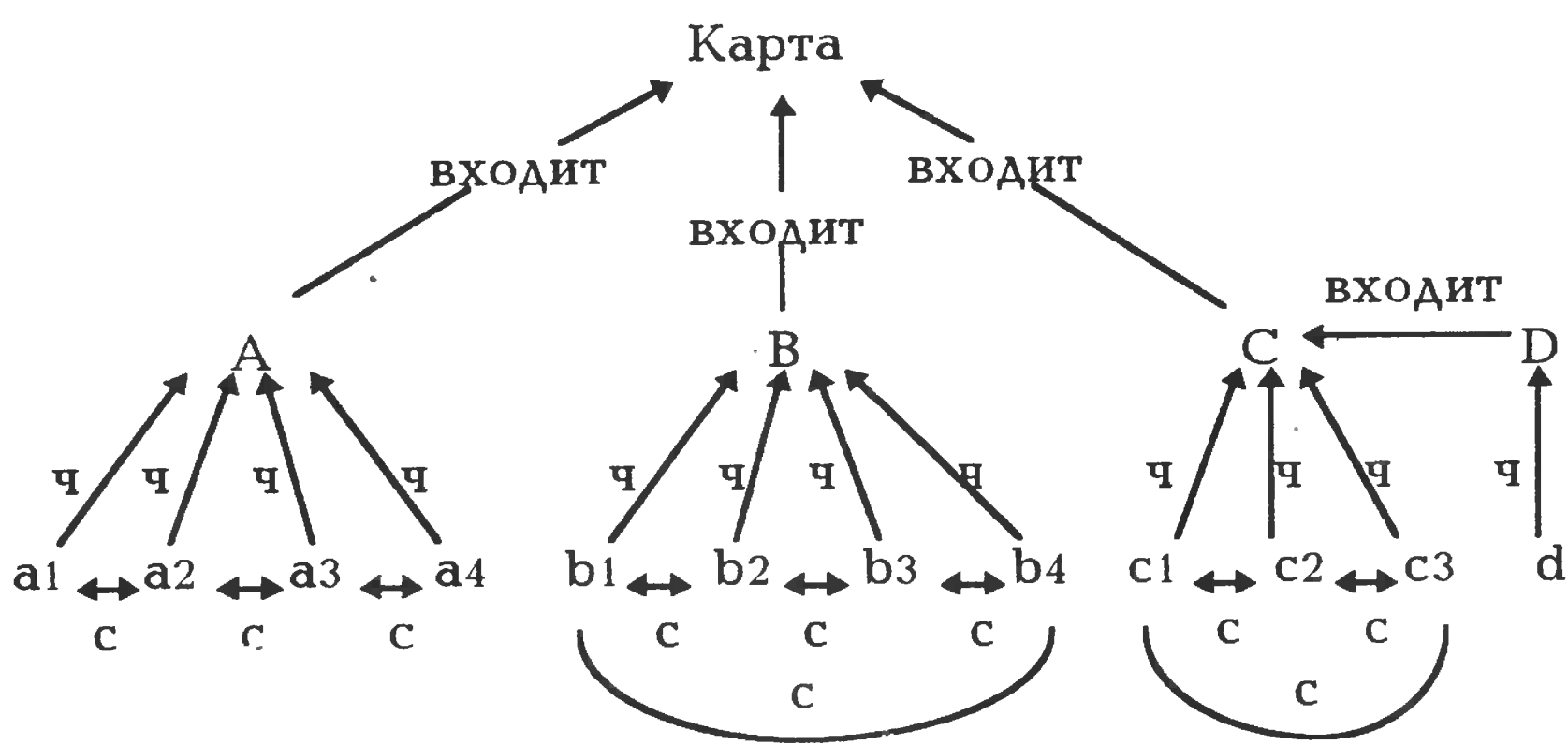


Рис. 2.13. Граф отношений

Входит		Часть		Соединение	
A	карта	a1	A	a1	a2
B	карта	a2	A	a2	a3
C	карта	a3	A	a3	a4
D	C	a4	A	b1	b2
		b1	B	b2	b3
		b2	B	b3	b4
		b3	B	b4	b1
		b4	B	c1	c2
		c1	C	c2	c3
		c2	C	c3	c1
		c3	C		
		d	D		

Рис. 2.14. Схема реляционной БД

3.1. СУЩНОСТЬ СИМВОЛЬНЫХ ИЗОБРАЖЕНИЙ

Определение пространственных соотношений является основой понимания изображений (см. гл.1), и позволяет отвечать на вопросы пользователя типа: "что находится юго-западнее дома с трубой?" или "найти все изображения, где справа от озера проходит дорога". Запрос может касаться части изображения, например, соответствует ли представляемая часть изображения какому-либо изображению, хранящемуся в памяти машины (или в визуальной базе данных).

В [47] для ответа на такие и подобные им запросы предложено использовать символьные изображения и их представления с помощью 2D-строк.

Сущность символьных изображений заключается в следующем [47]. После предварительной обработки и сегментации выделенные на изображении объекты классифицируются, причем (как один из вариантов) признаки, позволившие эту классификацию (распознавание) произвести, хранятся в памяти машины. Сами классифицированные объекты различаются с помощью меток (символов). Например, a -дом, b -озеро, $c_1 \dots c_m$ -дорога (вместе со своей траекторией). Затем на изображении (чаще всего с уменьшенной разрешающей способностью) размещаются указанные символы реальных объектов (символьные объекты), причем месторасположение этих символьных объектов выбирается, например, по геометрическим (или близким к ним) центрам. В результате и получаются символьные изображения, с помощью которых и определяются пространственные соотношения между символами сегментированных и классифицированных объектов, а значит и между самими объектами.

Роль символьных изображений этим не исчерпывается. Пусть запрос пользователя (например, найти все изображения,

хранящиеся в визуальной базе данных, в которых есть дома, расположенные вдоль реки) отработан, и все имеющиеся (символьные) изображения найдены. Предъявить пользователю их необходимо не в символьном виде, а близком к исходному. Методами машинной графики (не рассматриваемые в настоящей книге, см. [35]), по имеющимся в памяти признакам (а если надо, и по структурным описаниям) синтезируются реалистические изображения, которые и предъявляются пользователю.

3.2. ПРЕДСТАВЛЕНИЕ И СРАВНЕНИЕ СИМВОЛЬНЫХ ИЗОБРАЖЕНИЙ ПОСРЕДСТВОМ 2D-СТРОК

Представление символьных изображений посредством 2D-строк можно рассматривать как проекцию символьных изображений на направления осей OX и OY [47].

Пусть W – множество меток, или словарь; пустое множество $\{\}$ будет обозначать множество "нулевых" объектов. Введем три пространственных оператора $A = \{=, <, :\}$. Оператор "=" означает пространственное соотношение "в таком же самом положении, как". Оператор "<" означает левое–правое пространственное отношение в строке u (см. ниже) и нижнее–верхнее пространственное отношение в строке v (см. ниже). Оператор ":" означает отношение "в том же самом множестве, как". Указанные пространственные операторы используются для определения пространственных соотношений между символьными объектами на символьных изображениях.

1D-строка (над исходным множеством W) представляет собой запись $x_1x_2...x_n$ ($n \geq 0$), где все x_i ($i = 1, ..., n$) принадлежат множеству W .

2D-строка (также над исходным множеством W), записываемая как (u, v) , представляет собой запись вида

$$(x_1y_1x_2y_2...y_{n-1}x_n, x_{p(1)}z_1x_{p(2)}z_2...z_{n-1}x_{p(n)}),$$

где $x_1...x_n$ – 1D-строка (над множеством W); $p: \{1, ..., n\} \rightarrow \{1, ..., n\}$ –

некоторая перестановка; $y_1...y_{n-1}$ – 1D-строка (над множеством $A = \{=, <, :\}$, и $z_1...z_{n-1}$ – 1D-строка (также над множеством A).

Перейдем к примеру. Пусть символьное изображение имеет следующий вид (рис. 3.1):

d		
	b	c
a	a	

Рис. 3.1. Символьное изображение F

В рассматриваемом примере $W = \{a, b, c\}$. 2D-строка, описывающая символьное изображение F есть

$$(u, v) = (a = d < a = b < c, a = a < b = c < d) = \\ = (x_1 y_1 x_2 y_2 x_3 y_3 x_4 y_4 x_5 x_1 z_1 x_3 z_2 x_4 z_3 x_5 z_4 x_2),$$

где: $x_1 x_2 x_3 x_4 x_5 - adabc$; $x_1 x_3 x_4 x_5 x_2 - aabcd$; $p - 13452$

(перестановка диктуется расположением символьных объектов на символьном изображении);

$y_1 y_2 y_3 y_4$ есть $= < = <$ и $z_1 z_2 z_3 z_4$ есть $= < = <$.

Итак, заданному символьному изображению F соответствует 2D-строка (u, v) , и наоборот, причем левые-правые и нижние-верхние пространственные соотношения между символьными объектами сохраняются.

В [47] рассмотрены модификации 2D-строк, заключающиеся в том, что иногда некоторые пространственные операторы можно удалить (например операторы "=" и ":" могут быть опущены). В изложенном представлении символьного изображения F посредством 2D-строки учитывается только информация об относительном расположении; информация об абсолютном расположении теряется.

Итак, 2D-строка представляет собой еще один тип описания, но не исходных, а символьных изображений. Заметим, что структурное представление данных также можно интерпретировать как описание символьных изображений.

При сравнении изображений (в том числе и символьных) пользователя могут интересовать разные вопросы. Выделим три группы вопросов при сравнении символьных подизображений (изображений меньшего или равного размеров) с данным символьным изображением [47]:

1. Совпадает ли предъявленное подизображение с данным. Этот вопрос включает в себя как топологическую составляющую вопроса (конфигурацию расположения символьных обь-

ектов), так и метрическую (находятся ли символьные объекты на одинаковом расстоянии друг от друга).

2. Совпадает ли предъявленное подызоображение с данным в смысле конфигурации расположения (топологии) символьных объектов.

3. Имеются ли символьные объекты (расположенные на предъявляемом подызоображении) на данном символьном изображении (причем не важно, как именно расположенные).

В зависимости от утвердительных ответов на поставленные вопросы в [1] дана следующая классификация предъявляемых символьных подызоображений: 1) тип-0: если утвердительный ответ имеется лишь на третий вопрос; 2) тип-1: если утвердительный ответ дан на второй вопрос (при этом утвердительный ответ имеется и на третий вопрос); 3) тип-2: если утвердительный ответ дан на первый вопрос (при этом утвердительный ответ имеется и на остальные два вопроса).

Заметим, что в работе [60] дано несколько иное определение подызоображения типа-0, чем то, которое дано в [1] и используемое в этой книге.

На рис. 3.2 представлены четыре символьных изображения F , F_1 , F_2 и F_3 .

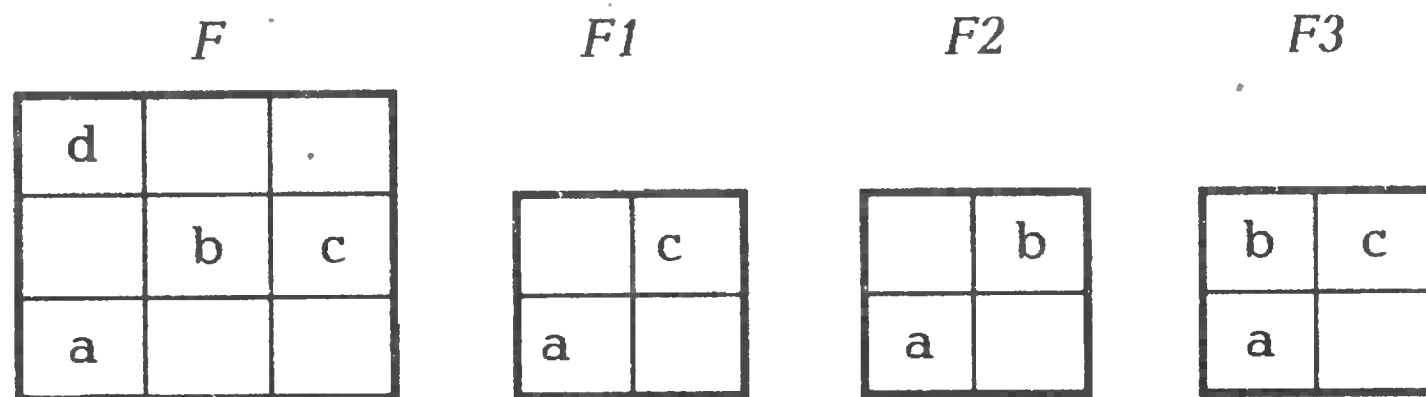


Рис. 3.2. Символьные изображения

Представление этих символьных изображений (с удаленными операторами "=" и ":") в виде 2D-строк имеют вид:

$$(u, v)_F = (ad < b < c, a < bc < d), \quad (u, v)_{F_1} = (a < b, a < b),$$

$$(u, v)_{F_2} = (a < c, a < c) \text{ и } (u, v)_{F_3} = (ab < c, a < bc).$$

Из данных определений следует, что F_3 – изображение тип-0; F_2 – изображение тип-1 (и тип-0); изображение F_1 является подызоображением тип-2 (а также и тип-1 и тип-0).

В [47] изложена формальная процедура сравнения символьных изображений с помощью $2D$ -строк (u,v) ; причем сравнение происходит отдельно для u -строки и для v -строки. Сравнение базируется на основе введенных в [47] рангов каждого символьного объекта строк u и v .

Имеется возможность представить символьные изображения несколько иначе, а именно в виде обобщенной $1D$ -строки G , в которой ранг каждого символьного объекта R равен его адресу. Например, символьное изображение F (рис. 3.1) может быть представлено в виде обобщенной $1D$ -строки G следующим образом:

$$G = (aR(1,1)dR(3,1)aR(1,2)bR(2,2),cR(2,3)).$$

Достоинством такого представления является его метричность (информация об абсолютном расположении символьных элементов не теряется), причем вопрос о выборе метрики решается в каждой задаче конкретно. Например, если расстояние определяется полетом самолетом, если от одного символьного объекта до другого можно добраться лишь по заданной сети дорог.

Трехмерные символьные изображения (полученные обработкой, сегментацией и классификацией трехмерной сцены) можно также представить либо в виде $3D$ -строки, либо в виде обобщенной $1D$ -строки вида G с трехадресной ссылкой.

3.3. НЕКОТОРЫЕ МЕТОДЫ ВЫЧИСЛИТЕЛЬНОЙ ГЕОМЕТРИИ В ЗАДАЧЕ ОПРЕДЕЛЕНИЯ ПРОСТРАНСТВЕННЫХ СООТНОШЕНИЙ

В вычислительной геометрии [30] под геометрическим поиском понимается установление соответствия между каким-либо заданным геометрическим объектом (прямоугольником, трапецией, сферой, гиперэллипсоидом и т.д.) и предъявляемым образцом (точкой, системой точек, и другими геометрическими объектами). Методы геометрического поиска успешно могут быть применены для задач определения пространственных соотношений. Перечислим некоторые из этих задач.

Итак, пусть дано символьное изображение F , содержащее N символьных объектов.

Сколько символьных объектов лежат внутри заданного прямоугольника (заданы координаты (адреса) четырех его вершин)? Задача решается просто. Необходимо проверить каждый из N символьных объектов на выполнение условия: удовлетворяют ли координаты (адреса) всех символьных объектов неравенствам, задающим прямоугольник.

Вместо запросов, в которых фигурирует прямоугольник (или другие, более сложные геометрические объекты) можно формулировать запросы, в которых фигурируют только точки (координаты, адреса) вершин прямоугольника. Это означает, что один запрос заменяется четырьмя (по числу точек, однозначно описывающих прямоугольник) и задача заключается в получении совместного решения для окончательного ответа.

Запросы, в которых фигурируют только точки, также имеют смысл для решения задач определения пространственных соотношений. Перечислим лишь две из них: 1) сколько символьных объектов (и какие) находятся юго-восточнее символьного объекта h ? 2) сколько символьных объектов (и какие) находятся севернее символьного объекта f и западнее символьного объекта d ?

3.4. МЕТОД ОРТОГОНАЛЬНЫХ СООТНОШЕНИЙ

Метод ортогональных соотношений, изложенный в [47] и описывающий пространственные соотношения, кроме традиционной обработки исходного изображения (предварительная обработка, сегментация и классификация) требует построения вокруг каждого выделенного и помеченного объекта описывающего прямоугольника. Построение описывающих прямоугольников (это простейшая фигура из плоско-пространственных геометрических объектов) фактически сводится к определению максимальных (x_{max}^i, y_{max}^i) и минимальных (x_{min}^i, y_{min}^i) координат каждого из выделенных объектов ($i = 1, \dots, N$), где N – число объектов на исходном изображении).

В результате на изображении будут объекты с описывающими прямоугольниками: 1) непересекающимися описывающими прямоугольниками; 2) частично-пересекающимися описывающими прямоугольниками; 3) полностью пересекающимися описывающими прямоугольниками.

Определение (с помощью имеющихся координат $x_{min}^i, y_{min}^i, x_{max}^i$ и $y_{max}^i, i = 1, \dots, N$), какой именно из трех случаев имеет место, не представляет затруднений.

Первый случай наиболее тривиален и он легко позволяет переходить к символьному изображению: достаточно найти геометрический центр прямоугольника, в котором и расположить уже символьные и помеченные ранее объекты.

Случай с частично-пересекающимися описывающими прямоугольниками более сложен. Один из возможных вариантов изображен на рис. 3.3, на котором через $a0$ и $b0$ условно изображены центры описывающих прямоугольников объектов a и b .

В этом случае поступают следующим образом. Рассматривают какой-либо объект (для конкретности b), который принимается за "точку обзора" [47], с координатами $b0$.

Пересечение описывающих прямоугольников определяет разбиение описывающего прямоугольника другого объекта (в данном случае a) на четыре прямоугольника a_1, a_2, a_3 и a_4 .

Определив геометрические центры прямоугольников a_2 и a_3 , мы получаем символьное изображение объекта a уже пространственно распределенным (до сих пор мы рассматривали только символьные объекты) вида $a = (a_2, a_3)$.

Построив таким образом символьное изображение, можем определять пространственные соотношения пространственно-распределенных символьных объектов.

Центр прямоугольника a_2 находится севернее объекта b , а центр прямоугольника a_3 находится западнее того же объекта b .

В других случаях кроме вариантов "севернее" и "западнее" могут использоваться и варианты "южнее" и "восточнее". Т.е. в рассматриваемом методе рассматриваются только четыре ортогональных направления (юг, запад, север и восток).

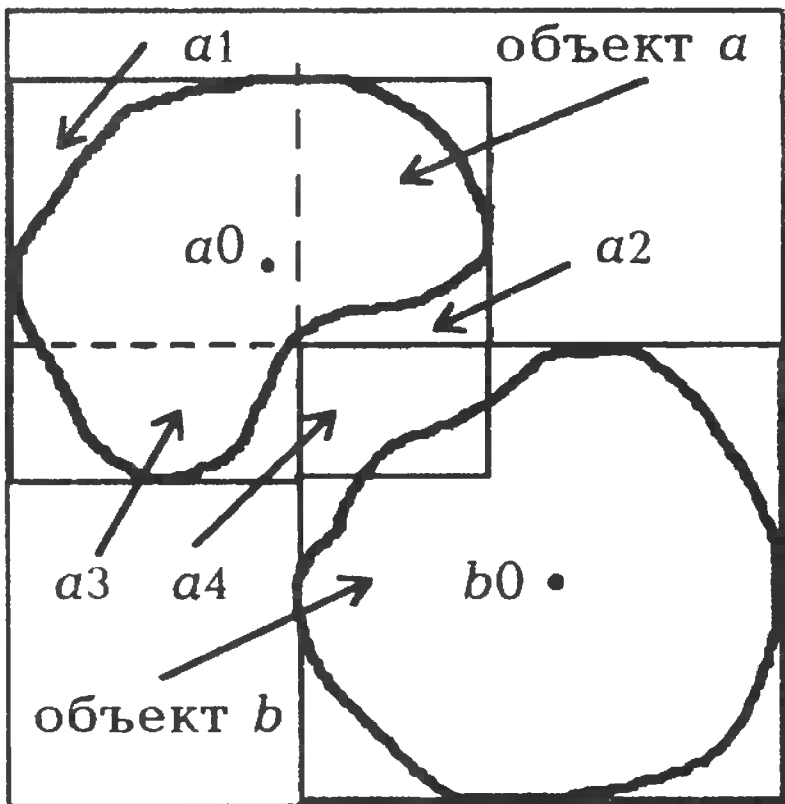


Рис. 3.3. Случай частично-пересекающихся описывающих прямоугольников

Откуда и название метода – метод ортогональных соотношений. Заметим, что в рассмотренном примере информация о том, что объект a расположен относительно объекта b и "северо-западнее" не теряется. Обратим внимание также на то, что соотношения "севернее", "южнее", "восточнее" и "западнее" имеют свои пространственные аналоги "выше", "ниже", и "следует за" [47].

И, наконец, третий случай (полностью пересекающиеся описывающие прямоугольники) отличается от рассмотренного тем, что объект (окружающий другой объект) виден из точки обзора (за которую принят геометрический центр "окруженного" прямоугольника) в более чем из двух ортогональных отношений (в трех или четырех при полном "окружении").

Если один объект окружает другой, например, с трех сторон (для конкретности он и южнее, и западнее, и севернее), то это однозначно означает, что этот объект находится и юго-западнее и северо-западнее. При окружении объекта со всех четырех ортогональных направлений вывод аналогичен.

В [47] изложено представление рассмотренных пространственных соотношений (окружение, частичное окружение) посредством $2D$ -строки. Указанные пространственные соотношения легко можно представить и посредством обобщенной $1D$ -строки G (см. п. 3.2).

Из изложенного ясно, как следует отвечать на более сложные запросы, например такие: находится ли река r (пространственно-символьный объект) между городом t и озером s и есть ли возможность перейти от города к озеру (и наоборот) через мост b ? Возможны также и ответы на запросы, имеющие метрический характер, например такие: находятся ли символьные объекты a и b вблизи друг друга (это означает, что между ними нет других символьных объектов).

Для расширения возможностей пространственного понимания изображений (с помощью символьных) должна учитываться и другая информация: 1) метрическая (представление символьных изображений посредством обобщенной $1D$ -строки G в принципе это позволяет); 2) форма объекта (учитывается, например, с помощью цепного или комплексно-значного кода (см. гл. 2)); 3) ориентация объектов (знание ориентации объектов позволяет чувствовать случаи, когда объекты имеют одинаковые проекции вдоль осей север-юг и запад-восток) и др.

Указанная информация должна быть выделена на этапах переработки информации, предшествующих пониманию изображений и храниться с помощью соответствующих структур данных в памяти машины.

3.5. СРАВНЕНИЕ СИМВОЛЬНЫХ ИЗОБРАЖЕНИЙ НА ПИРАМИДАЛЬНЫХ СТРУКТУРАХ

В гл. 2 для представления исходных изображений были описаны пирамидальные структуры, в целом приводящие к упрощению сегментации, поиска изображений в визуальной базе данных и т.д.

В [60] изложены два алгоритма, осуществляющие сравнение символьных изображений, представленных не посредством 2D-строк (см. п. 3.2), а в виде пирамидальной структуры. Там же изложена сущность пирамидального компьютера.

Предполагается, что исходное символьное изображение F загружено в пирамидальную структуру и находится на наименьшем уровне, который этому изображению соответствует. При размерности символьного изображения F $M \times N$ номер этого уровня определяется как $\log_{\max}(M, N)$ [60].

t	#
h	r

Рис. 3.4. Образец запроса в виде символьного изображения P

Пусть нам необходимо найти все изображения, на которых есть дом h , севернее которого находится дерево t , а восточнее – река r . Образец запроса представлен на рис. 3.4.

Продemonстрируем обработку данных запроса, отвечая на вопрос: является ли образец запроса P , представленный на рис. 3.4, подизображением типа-2 исходного изображения F , представленного на рис. 3.5.

	t	h	t
t	h	r	s
t	r	h	r

Рис.3.5. Исходное символьное изображение F

Алгоритм работает следующим образом [60]. На изображении P ищем элемент (символьный объект), с которого начинается просмотр. Так как в изображении четное число строк, то просмотр начинается с левого нижнего элемента h . При нечетном числе строк просмотр начинался бы с правого нижнего элемента символьного изображения P . Кроме выбранного элемента, на каждом шаге алгоритма выбирается еще и направление, определяемое как *запад* или *восток*. В данном случае это направление *восток*. При этом образуем такую же бинарную матрицу A_0 , которая имеет единичные элементы там, где на символьном изображении есть выбранный элемент h . Матрица A_0 имеет вид, показанный на рис. 3.6.

A0	0	0	1	0
	0	1	0	0
	1	0	1	0

A1	0	0	0	1
	0	0	1	0
	0	1	0	1

Рис. 3.6. Исходная матрица A_0 и матрица A_1

Так как выбрано направление *восток*, матрица A_1 получается из матрицы A_0 сдвигом на одну позицию на *восток* (см. рис.3.6).

На символьном изображении P вторым элементом является h и направление *север* (направления *восток* в данном случае не существует).

Если в символьном изображении F в тех позициях, в которых в матрице A_1 соответствуют единицы, есть элементы $г$, то в матрице A_2 на эти позиции "ставятся" единицы. Матрица A_3 образуется сдвигом на одну позицию на *север* единиц из матрицы A_2 (это выбранные направления от элемента $г$).

A2	0	0	0	0
	0	0	1	0
	0	1	0	1

A3	0	0	1	0
	0	1	0	1
	0	0	0	0

Рис. 3.7. Вид матриц A2 и A3

Матрицы A2 и A3 имеют вид, показанный на рис. 3.7. При этом на каждом шаге алгоритма вновь получаемые матрицы A_i проверяются на условие наличия всех нулей. Если это условие на каком-либо шаге выполняется, то алгоритм останавливается с решением: вхождение j -го типа (в нашем случае тип-2) не обнаружено. Следующий элемент символьного изображения P для сравнения – пустой. Сравнения не происходит, однако направление следующего просмотра – *запад*. Следовательно матрица A4 получается из матрицы A3 сдвигом всех единиц на одну позицию на *запад*. В матрице A5 единицы имеются на тех позициях, где в матрице A4 есть единицы, а в символьном изображении F есть элементы t .

Матрицы A4 и A5 имеют вид, представленный на рис. 3.8.

A4	0	1	0	0
	1	0	1	0
	0	0	0	0

A5	0	1	0	0
	1	0	0	0
	0	0	0	0

Рис. 3.8. Вид матриц A4 и A5

Последующие уровни пирамиды образуются по схеме "ИЛИ" (единица "попадает" на следующий уровень, если таковые имеются хотя бы в одном из элементов предыдущего уровня). Сказанное означает, что верхний уровень пирамиды имеет значение "1". Следовательно, вхождение символьного изображения P тип-2 в исходное изображение F обнаружено.

Модификации изложенного алгоритма для определения вхождения других типов (тип-0 и тип-1) изложены в [60].

Сущность второго алгоритма, определяющего вхождение только тип-0 и тип-1 см. в [60].

Глава 4

АЛГОРИТМЫ ПРЕДВАРИТЕЛЬНОЙ ОБРАБОТКИ ВИДЕОДАНЫХ

В п. 1.1 были перечислены некоторые из задач, традиционно относящиеся к задачам предварительной обработки изображений. Из перечисленного множества задач в настоящей главе рассмотрены лишь различные задачи локальной фильтрации изображений, что, в первую очередь, обусловлено тем, что именно задачи фильтрации являются теми самыми "алгоритмами поддержки" (см. гл. 1), которые в наибольшей степени обеспечивают дальнейшую машинную обработку, имеющую конечной целью создание алгоритмов понимания изображений.

4.1. СУЩНОСТЬ ЛИНЕЙНОЙ ЛОКАЛЬНОЙ ФИЛЬТРАЦИИ ИЗОБРАЖЕНИЙ

Как указывалось ранее (см. п. 1.1), практически любое изображение зашумлено, а значит и возникает проблема борьбы с шумами. Одним из наиболее распространенных способов борьбы с шумами является фильтрация, которая с самого начала была применена к обработке одномерных сигналов. Не составляет исключения и борьба с шумами на зашумленных изображениях, но с особенностями, связанными с двумерностью видеоинформации. Указанная особенность приводит к тому, что вместо одномерного фильтра, предназначенного для фильтрации зашумленных сигналов одной переменной (обычно этой переменной является время), для фильтрации изображений чаще всего применяются двумерные фильтры, представляющие собой апертуры (маски) различной конфигурации на плоскости. Каждому элементу апертуры соответствует (присваивается исследователем) число, в дальнейшем называемое весовым множителем [31].

Фильтрация осуществляется перемещением слева-направо (или сверху-вниз) апертуры фильтра (маски) на один пиксел. При каждом положении апертуры производятся однотипные операции, а именно перемножение весовых множителей с со-

ответствующими значениями яркостей исходного изображения и суммированием полученных результатов. Полученное значение делится на заранее заданное число (нормирующий множитель), которое и присваивается центральному элементу апертуры (это и есть "выход" фильтра). Размеры апертуры берутся таким образом, чтобы центральный элемент апертуры определялся однозначно. Наиболее употребимые размеры апертуры 3×3 и 5×5 [5, 31].

Применяется также и так называемая разделимая фильтрация [8], суть которой заключается в том, что фильтрация осуществляется одномерным фильтром (фильтром с линейной апертурой). При этом обработка изображений осуществляется в два этапа. На первом этапе фильтрация осуществляется перемещением апертуры фильтра вдоль строк (или столбцов). На втором этапе процессу фильтрации подвергается уже отфильтрованное одномерным фильтром изображение: фильтрация осуществляется перемещением апертуры фильтра уже вдоль столбцов (или строк, если на первом этапе фильтрация осуществлялась перемещением апертуры фильтра вдоль столбцов).

Таким образом, получается отфильтрованное (линейным образом) изображение, причем термин *локальный* свидетельствует о том, что размеры апертуры по обеим осям меньше соответствующих размеров исходного (фильтруемого) изображения. В противном случае мы имеем уже глобальную фильтрацию, частично рассмотренную в гл. 6 (см. раздел "Кодирование с преобразованием"). Конкретизируем сказанное на примере фильтра, сглаживающего шум [31, 36]. Пусть мы имеем прямоугольную апертуру размером 3×3 , причем весовые и нормирующие множители имеют вид (через H обозначено множество весовых множителей, определяющих данный фильтр)

$$H = \frac{1}{10} \begin{vmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{vmatrix}.$$

Тогда выходное значение фильтра y определяется как

$$y = \frac{1}{10} (x_{i-1,j-1} + x_{i-1,j} + x_{i-1,j+1} + x_{i,j-1} + 2x_{i,j} + x_{i,j+1} + x_{i+1,j-1} + x_{i+1,j} + x_{i+1,j+1}),$$

где x_{ij} – элементы исходного (входного) изображения, "попав-

шие" в апертуру фильтра

$$X = \begin{vmatrix} x_{i-1, j-1} & x_{i-1, j} & x_{i-1, j+1} \\ x_{i, j-1} & x_{i, j} & x_{i, j+1} \\ x_{i+1, j-1} & x_{i+1, j} & x_{i+1, j+1} \end{vmatrix}.$$

Приведем еще два часто встречающихся вида весовых множителей шумоподавляющих фильтров [31]

$$H = \frac{1}{9} \begin{vmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{vmatrix}, \quad H = \frac{1}{16} \begin{vmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{vmatrix}.$$

Правый вид весовых множителей подчеркивает (большими весами) четырехсвязные элементы исходного изображения.

Если необходимо подчеркнуть восьмисвязные элементы исходного изображения, не являющиеся четырехсвязными, то можно предложить, например, следующее множество весовых множителей:

$$H = \frac{1}{15} \begin{vmatrix} 2 & 1 & 2 \\ 1 & 2 & 1 \\ 2 & 1 & 2 \end{vmatrix}.$$

Существуют другие принципы локальной фильтрации, когда не все элементы входного изображения, находящиеся в данный момент внутри апертуры (маски) учитываются при формировании выходного значения фильтра [5].

Например, сущность так называемой сигма-фильтрации заключается в том, что (яркости пикселей входного изображения распределены нормально, с известным среднеквадратичным значением σ и математическим ожиданием E_x) не учитываются (при формировании выходного значения фильтра) те значения элементов входного изображения, яркости которых лежат вне значений $E_x \pm 2\sigma$ [9]. Из теории вероятностей известно [9], что в указанном диапазоне находятся $\approx 95.5\%$ всевозможных значений яркости.

Рассмотренная выше фильтрация характеризовалась тем, что выходные значения фильтра y определялись только через

входные значения фильтра x . Такие фильтры называются *нерекурсивными* [31, 45, 54, 59].

При частотной интерпретации шумоподавляющие фильтры являются фильтрами нижних частот [5, 27, 31]. Фильтры, в которых выходные значения y определяются не только через входные значения x , но и через соответствующие выходные значения, называются *рекурсивными*.

В рекурсивной фильтрации будем сохранять те же значения весовых множителей.

Элементы входного изображения XU теперь изменятся и примут вид:

$$XU = \begin{vmatrix} x_{i-1, j-1} & x_{i-1, j} & x_{i-1, j+1} \\ x_{i, j-1} & x_{i, j} & x_{i, j+1} \\ x_{i+1, j-1} & x_{i+1, j} & x_{i+1, j+1} \end{vmatrix}.$$

Рекурсивным фильтром первого рода будем называть такой фильтр, у которого выход y формируется только перемножением весовых множителей H с элементами изображения XU ; нормирующий множитель при этом такой же, как и у нерекурсивного фильтра.

Рекурсивным фильтром второго рода будем называть такой фильтр, у которого выход y формируется как у рекурсивного фильтра первого рода плюс выход нерекурсивного фильтра, но с весовыми множителями вида

$$H = \begin{vmatrix} x_{i-1, j-1} & x_{i-1, j} & x_{i-1, j+1} \\ x_{i, j-1} & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix}.$$

Оба нормирующих множителя задаются эвристически. Заметим, что выход рекурсивных фильтров обоих родов зависит от траектории перемещения апертуры фильтра.

Рассмотрим динамику изменения весовых множителей на примере рекурсивного фильтра первого рода с линейной апертурой размером в три пиксела. Пусть одномерный (с линейной апертурой) низкочастотный, шумоподавляющий фильтр имеет весовые множители вида $H_1 = \frac{1}{3} [1, 1, 1]$.

При сдвиге апертуры фильтра на один пиксел вправо значения выхода фильтра $y_2 = \frac{1}{3}(y_1 + x_2 + x_3)$. Подставляя сюда значение $y_1 = \frac{1}{3}(x_0 + x_1 + x_2)$, можно получить, что весовые и нормирующие множители изменились и приняли вид $H_2 = \frac{1}{9}[1, 1, 4, 3]$. Повторяя проделанное, получим, что $H_3 = \frac{1}{2}[1, 1, 4, 12, 9]$ и $H_4 = \frac{1}{81}[1, 1, 4, 12, 36, 27]$.

Если исходные весовые и нормирующий множители низкочастотного фильтра имеют вид $H_1 = \frac{1}{4}[1, 2, 1]$, то аналогично предыдущему случаю можно получить, что $H_2 = \frac{1}{16}[1, 2, 9, 4]$,

$H_3 = \frac{1}{64}[1, 2, 9, 36, 16]$ и $H_4 = \frac{1}{25}[1, 2, 9, 36, 144, 64]$.

Из сказанного следует: 1) рекурсивные фильтры являются неоднородными (весовые и нормирующий множители зависят от местоположения апертуры); 2) рекурсивные фильтры обладают бесконечной памятью (фильтр каждый раз учитывает все предыдущие значения входного сигнала).

Сказанное означает, что без увеличения вычислительных затрат рекурсивный фильтр позволяет учитывать все входные значения фильтруемого изображения, что в ряде случаев позволяет надеяться на улучшение эффективности фильтра по сравнению с нерекурсивным фильтром.

Заметим, что и сигма-фильтрация может осуществляться в рекурсивной форме.

Фильтры, подчеркивающие перепады яркости и границы (в частотной интерпретации это высокочастотные фильтры) имеют три типовых набора весовых множителей [31]

$$H = \begin{vmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{vmatrix}, \quad H = \begin{vmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{vmatrix}, \quad H = \begin{vmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{vmatrix}.$$

Обратим внимание на то, что сумма весовых множителей приведенных масок равна единице [31].

Для подчеркивания перепадов используются следующие наборы весовых множителей (двумерное дискретное дифференцирование) [31].

$$\begin{array}{cccc}
 \text{север} & \text{северо-восток} & \text{восток} & \text{юго-восток} \\
 \mathbf{H} = \begin{vmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{vmatrix}, & \mathbf{H} = \begin{vmatrix} 1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & -1 & 1 \end{vmatrix}, & \mathbf{H} = \begin{vmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{vmatrix}, & \mathbf{H} = \begin{vmatrix} -1 & -1 & 1 \\ -1 & -2 & 1 \\ 1 & 1 & 1 \end{vmatrix},
 \end{array}$$

$$\begin{array}{cccc}
 \text{юг} & \text{юго-запад} & \text{запад} & \text{северо-запад} \\
 \mathbf{H} = \begin{vmatrix} -1 & -1 & -1 \\ 1 & -2 & 1 \\ 1 & 1 & 1 \end{vmatrix}, & \mathbf{H} = \begin{vmatrix} 1 & -1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & 1 \end{vmatrix}, & \mathbf{H} = \begin{vmatrix} 1 & 1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & -1 \end{vmatrix}, & \mathbf{H} = \begin{vmatrix} 1 & 1 & 1 \\ 1 & -2 & -1 \\ 1 & -1 & -1 \end{vmatrix}.
 \end{array}$$

Название географического направления говорит о направлении перепада, которое вызывает максимальный отклик фильтра.

Для подчеркивания перепадов без учета их ориентации используются следующие три вида весовых множителей (операторы Лапласа) [31]

$$\mathbf{H} = \begin{vmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{vmatrix}, \quad \mathbf{H} = \begin{vmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{vmatrix}, \quad \mathbf{H} = \begin{vmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{vmatrix}.$$

Сумма весовых множителей приведенных масок равна нулю, что говорит о том, что в областях изображения с постоянной яркостью операторы Лапласа дают нулевой отклик.

Сказанное выше относительно рекурсивной фильтрации справедливо, конечно, и для фильтров, подчеркивающих перепады и границы. Как показано в [5], операторы Лапласа реагируют на перепады яркости в виде ступенчатого перепада и на "крышеобразный" перепад.

Оператор Лапласа также выделяет на изображении изолированные точки, тонкие линии, их концы и острые углы объектов. Линия подчеркивается в 2 раза ярче, чем ступенчатый перепад, конец линии в 3 раза, а точка – в 4 раза [5]. Учитывая, что оператор Лапласа выделяет в основном не структурированные элементы (точки), то отсюда следует вывод, что оператор Лапласа очень чувствителен к шуму. Хотя подчеркивание перепадов с помощью оператора Лапласа происходит без учета их ориентации, оператор Лапласа не инвариантен к ориентации перепадов: например отклик оператора на наклонный перепад в диагональном направлении почти вдвое больше, чем в горизонтальном и вертикальном [5].

Для уменьшения чувствительности к ориентации перепадов в [5] предложен набор весовых коэффициентов вида

$$H = \begin{vmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{vmatrix}.$$

Для повышения помехоустойчивости в обработке изображения иногда совмещают процедуру низкочастотной фильтрации с применением оператора Лапласа; при этом весовые множители берутся пропорциональными значениям двумерной гауссовой кривой [5, 31].

И, наконец, отметим возможность применения весовых множителей, которые пропорциональны коэффициентам корреляции элементов исходного изображения [31]. Если моделью исходного изображения является марковский процесс, то весовые множители имеют вид [31]

$$H = \begin{vmatrix} \rho_c \rho_r & -\rho_c (1 + \rho_r^2) & \rho_c \rho_r \\ -\rho_r (1 + \rho_c^2) & (1 + \rho_c^2)(1 + \rho_c^2) & -\rho_r (1 + \rho_c^2) \\ \rho_c \rho_r & -\rho_c (1 + \rho_r^2) & \rho_c \rho_r \end{vmatrix},$$

где $\rho_c(\rho_r)$ – коэффициенты корреляции между соседними элементами по строке (столбцу).

При $\rho_c = \rho_r = 0$ все весовые множители равны 0 (кроме центрального, который равен 1) и корреляционная маска не изменяет исходного изображения. При $\rho_c = \rho_r = 1$ корреляционная маска сводится к оператору Лапласа с весовыми множителями вида

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

4.2. СУЩНОСТЬ НЕЛИНЕЙНОЙ ЛОКАЛЬНОЙ ФИЛЬТРАЦИИ ИЗОБРАЖЕНИЙ

Основные понятия теории линейной локальной фильтрации, изложенные в п. 4.1, справедливы и для нелинейной фильтрации. Имеется в виду понятие самого процесса фильтрации, понятие апертуры (маски), весовых множителей и понятия рекурсивных и нерекурсивных фильтров.

Основное отличие от линейной локальной фильтрации заключается в том, что выход нелинейного фильтра формируется нелинейным образом от данных исходного изображения, а для рекурсивной фильтрации нелинейным образом от данных уже отфильтрованного (на предыдущих шагах) изображения. Начнем изложение с медианного фильтра, сущность которого заключается в том, что выход фильтра y равен медиане данных исходного изображения, находящихся в апертуре [8, 31].

Медиана представляет собой центральный элемент в вариационном ряду, полученном из данных исходного изображения. В силу того, что для операции нахождения медианы не выполняется одна из аксиом линейности [31], медианный фильтр является нелинейным. Медианный фильтр не влияет на ступенчатые и линейно-изменяющиеся сигналы, что в ряде случаев является желательным свойством. При этом медианный фильтр подавляет импульсные выбросы исходного изображения (если длительность импульсного сигнала составляет менее половины длины линейной апертуры).

Если выходу фильтра присваивать не значение медианы данных, находящихся в апертуре, а значение любой r -й ($r = 1, \dots, N$), N – общее число элементов апертуры) порядковой статистики, то мы имеем $N-1$ фильтров, названных в [8] процентильными. Процентильные фильтры также являются нелинейными. Заметим, что N – нечетное число. Целесообразность введения процентильных фильтров заключается, в частности, в том, что в результате обработки исходных изображений процентильными фильтрами происходит либо уменьшение размеров объектов, находящихся на изображении ($r > \frac{N+1}{2}$), либо их увеличение ($r < \frac{N+1}{2}$; при $r = \frac{N+1}{2}$ имеем медианный фильтр). Конечно и для процентильных, и для медианных фильтров существует их рекурсивная версия, рассмотренная в п.4.1. В [31] для выделения перепадов предложили следующую аппроксимацию величины градиента

$$y_{ij} = ((x_{ij} - x_{i+1,j+1})^2 + (x_{i,j+1} - x_{i+1,j})^2)^S.$$

В последней формуле для вычисления выхода фильтра y_{ij} требуется выполнение таких трудоемких операций, как возведение в квадрат и извлечение квадратного корня (обычно эти операции реализуются табличным способом [5]). Поэтому последнюю формулу заменяют более простой вида [5, 31]

$$y_{ji} = |x_{ij} - x_{i+1,j+1}| + |x_{i,j+1} - x_{i+1,j}|.$$

Обе аппроксимации градиента обычно называют дифференциальными операторами [5]. Из последних двух формул видно, что применяется апертура размером 2×2 . Для уменьшения влияния к аддитивному шуму обычно увеличивают размеры апертуры [5]. Пусть мы имеем двумерную квадратную апертуру размером 3×3 , представленную выражением

$$\begin{vmatrix} A_0 & A_1 & A_2 \\ A_7 & y_{ij} & A_3 \\ A_6 & A_5 & A_4 \end{vmatrix},$$

где через A_0, \dots, A_7 обозначены входные значения фильтруемого изображения, а через y_{ij} – результат фильтрации. Собел [31] предложил следующую формулу для расчета величины y_{ij}

$$y_{ij} = (g^2 + f^2)^S, \quad \text{где}$$

$$g = (A_2 + 2A_3 + A_4) - (A_0 + 2A_7 + A_6), \quad f = (A_0 + 2A_1 + A_2) - (A_6 + 2A_5 + A_4).$$

Рекурсивная версия оператора Собела работает с апертурой вида

$$\begin{vmatrix} B_0 & B_1 & B_2 \\ B_7 & y_{ij} & A_3 \\ A_6 & A_5 & A_4 \end{vmatrix},$$

где через B_0, \dots, B_2 и B_7 условно обозначены выходные значения отфильтрованного изображения.

В [5] указан набор весовых множителей апертуры размером 5×5 (расширенный оператор Собела). Еще один нелинейный оператор для контрастирования перепадов предложил Кирш [31]. Выход фильтра y_{ij} вычисляется как

$$y_{ij} = \max \left\{ 1, \max_{i=0.7} |5Q_i - 5T_i| \right\},$$

$$Q_i = A_i + A_{i+1} + A_{i+2}, \quad T_i = A_{i+3} + A_{i+4} + A_{i+5} + A_{i+6} + A_{i+7}.$$

Необходимые для расчета величин y_{ij} параметры (g, f, Q_i, T_i) можно представить в виде масок со следующими весовыми

множителями. Для описанных операторов Собела и Кирша эти весовые множители соответственно имеют вид [5]

$$H = \begin{vmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{vmatrix}, \quad H = \begin{vmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{vmatrix}, \quad H = \begin{vmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{vmatrix}.$$

Из наиболее известных приведем еще весовые множители операторов Прэвитт 1 и Прэвитт 2 [5], которые соответственно имеют вид

$$H = \begin{vmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{vmatrix}, \quad H = \begin{vmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{vmatrix}.$$

Последовательно "поворачивая" весовые множители на 45° , получаем восемь различно ориентированных апертур [5].

Кроме дифференциальных операторов (аппроксимирующих величины градиента) существуют еще так называемые масочные операторы, аппроксимирующие величины градиентов как

$$y_{ij} = \max \{ |x_{ij} - x_{i+1,j+1}|, |x_{i,j+1} - x_{i+1,j}| \}.$$

В [5] была сделана попытка сформулировать разумный выбор весовых множителей. Этот выбор базируется на четырех предположениях.

1. Весовые множители, соответствующие вертикальному направлению перепада, должны быть с точностью до поворота на 90° эквивалентны весовым множителям, соответствующим горизонтальному направлению перепада. Это позволяет рассматривать только весовые множители, соответствующие вертикальному направлению перепада.
2. Весовые множители (обозначим их через A_0, \dots, A_7) должны быть симметричны относительно вертикальной оси, т.е.

$$A_0 = A_2, \quad A_7 = A_3 \quad \text{и} \quad A_6 = A_4.$$

3. Весовые множители, расположенные симметрично относительно горизонтали, должны иметь противоположные знаки, т.е.

$$A_0 = -A_6, \quad A_1 = -A_5, \quad A_2 = -A_4.$$

4. Отклик на однородные по яркости изображения должен быть равным нулю,

$$A_0 + A_1 + A_2 + A_3 + A_4 + A_5 + A_6 + A_7 + A_8 + A_9 = 0,$$

где через A_9 обозначен весовой множитель, соответствующий центральному элементу апертуры.

Объединение последних трех выражений дает только три свободных весовых множителя вида

$$\mathbf{H} = \begin{vmatrix} A_0 & A_1 & A_0 \\ A_7 & -2A_7 & A_7 \\ -A_0 & -A_1 & -A_0 \end{vmatrix}.$$

Для устранения неинтерпретируемого разумным образом вклада в выходное значение y_{ij} весового множителя A_7 , необходимо положить его равным нулю. В результате получаем следующий набор весовых множителей

$$\mathbf{H} = \begin{vmatrix} A_0 & A_1 & A_0 \\ 0 & 0 & 0 \\ -A_0 & -A_1 & -A_0 \end{vmatrix}.$$

Указанному набору весовых множителей удовлетворяют только операторы Собела и Прэвитт 1 [5]. В работе [5] приведена оценка работы различных операторов. Обобщающих выводов два: 1) операторы Собела и Прэвитт 1 эффективнее других операторов в смысле вероятности правильной классификации перепадов; 2) оценка величины градиента с помощью оператора Собела менее чувствительна к изменению ориентации перепадов по сравнению с оператором Прэвитт 1; однако оператор Прэвитт 1 лучше работает при горизонтальных и вертикальных перепадах.

Перейдем к фильтрации бинарных изображений, которая в силу того, что и выход бинарного фильтра должен быть бинарным, является нелинейной [46; 59]. Учитывая, что в рассматриваемом (бинарном) случае в апертуре фильтра находятся только нули и единицы, то представляется целесообразным следующее правило формирования выхода бинарного медианного фильтра (алгоритма нелинейной фильтрации): если в последовательности размером N (N – нечетное, общее число элементов апертуры) число единиц N_1 больше числа нулей, то центральному элементу апертуры (выходу фильтра) присваивается значение равное единице. В противном случае выходу фильтра присваивается значение равное нулю [46, 59].

Можно произвести следующее обобщение и для бинарного случая: центральному элементу апертуры присваивается значение равное единице, если число единиц в апертуре равно N_1 ($1 \leq N_1 \leq N$). Указанный класс фильтров будем называть процентильными бинарными фильтрами [46, 59].

Все сказанное в п. 4.1 относительно рекурсивной фильтрации справедливо и для бинарного случая.

4.3. НЕЛИНЕЙНАЯ ФИЛЬТРАЦИЯ КОМПЛЕКСНО-ЗНАЧНЫХ КОДОВ КОНТУРОВ ВЫДЕЛЯЕМЫХ ОБЪЕКТОВ

Рассмотрим алгоритм фильтрации кодов контуров, т.е. фильтрацию в признаковом пространстве [53, 58]. В контурах выделяемых объектов, как правило, содержится вся необходимая информация для их надежного распознавания по контурным признакам, как указывалось в гл. 2. При цепном кодировании контуров по Фримену каждому из восьми направлений элементарных векторов (элементов кода) присваивается номер от 0 до 7, а при комплексно-значном кодировании по Фурману каждому элементу кода (элементарному вектору) ставится в соответствие комплексное число. При наличии разного рода помех искажаются как сами изображения, так и их контуры вместе с их цепными и комплексно-значными кодами, что приводит к уменьшению вероятностной надежности процедур сегментации и соответственно распознавания. Для уменьшения влияния помех, а следовательно и увеличения вероятностной надежности распознавания, обычно применяют различного рода фильтрацию (см п. 4.1 и 4.2), в том числе и неоднократную, что приводит к значительным вычислительным и временным затратам. В настоящем пункте изложен метод фильтрации, требующий значительно меньших вычислительных и временных затрат, так как вместо традиционной попиксельной обработки всего изображения, что и рассматривалось ранее в этой главе, обработке подлежат лишь элементы комплексно-значного кода выделяемых объектов, которые формируются в процессе получения изображения [53, 58].

В табл. 2.1 были представлены значения всех элементов цепного кода Фримена и комплексно-значного кода и соответствующие ему значения мнимой (Im) и действительной (Re) компонент ($I = -1$). Фильтровать можно как сам комплексно-значный

код, состоящий из восьми элементов, так и его мнимые (Im) и действительные (Re) компоненты, состоящие каждая из трех элементов тринарной последовательности $(-1, 0, 1)$ [53, 58].

Начнем рассмотрение с покомпонентной фильтрации (фильтрации тринарных последовательностей). Учитывая, что каждая из компонент комплексно-значного кода представляет собой одномерную последовательность, то для их фильтрации необходим фильтр с линейной, а не пространственной апертурой.

Изложим сущность нелинейной фильтрации на примере фильтра с линейной апертурой, состоящей из трех элементов ($m=3$). Введем следующее нелинейное правило принятия решения (фильтрации), представленное в табл. 4.1. Выход фильтра, формируемый с помощью правила (алгоритма), представленного табл. 4.1, представляет собой медиану тринарной последовательности [32, 59]. Поэтому предложенный фильтр будем именовать медианным. Общее число входных тринарных последовательностей как нетрудно видеть, равно 3^m .

Таблица 4.1.

Входная последовательность	Выход фильтра	Входная последовательность	Выход фильтра	Входная последовательность	Выход фильтра
1 1 1	1	0 1 1	1	-1 1 1	1
1 1 0	1	0 1 0	0	-1 1 0	0
1 1 -1	1	0 1 -1	0	-1 1 -1	-1
1 0 1	1	0 0 1	0	-1 0 1	0
1 0 0	0	0 0 0	0	-1 0 0	0
1 0 -1	0	0 0 -1	0	-1 0 -1	-1
1 -1 1	1	0 -1 1	0	-1 -1 1	-1
1 -1 0	0	0 -1 0	0	-1 -1 0	-1
1 -1 -1	-1	0 -1 -1	-1	-1 -1 -1	-1

Рассмотрим теперь сущность процесса нелинейной фильтрации самого комплексно-значного кода. Аналогично предыдущему случаю, сформулируем медианное правило принятия решения для фильтра с линейной апертурой, состоящей из трех элементов ($m=3$). Если в трех смежных элементах входного комплексно-значного кода все три элемента кода равны между собой, то выход фильтра равен этому элементу кода. Если в трех смежных элементах входного кода только два элемента кода

равны между собой, то и выход фильтра равен этим, равным друг другу элементам.

И, наконец, если все три элемента входного кода между собой различны, то выходу медианного фильтра присваивается то значение элемента комплексно-значного кода, которое находится между двумя остальными значениями, причем все три элемента входного комплексно-значного кода лежат на дуге меньшей длины (все элементы комплексно-значного кода лежат на окружности единичного радиуса). В рассматриваемом случае общее число входных элементов последовательностей равно $8^m = 8^3 = 512$. Обобщение обоих правил принятия решения на случай $m = 2k + 1$, при $k > 1$ очевидно. Заметим, что изложенные принципы нелинейной (медианной) фильтрации не изменяют длины комплексно-значных контуров.

Рассмотрим фильтрацию шумового выброса на горизонтальном участке контура размером в один пиксел. Комплексно-значный код зашумленного таким образом горизонтального участка контура имеет вид (см. табл. 2.1) $\dots 1, 1, 1, 1+i, 1-i, 1, 1, 1 \dots$. Мнимая компонента указанного комплексно-значного кода имеет вид (см. табл. 2.1) $\dots 0, 0, 0, 1, -1, 0, 0, 0 \dots$. Фильтруя мнимую компоненту по правилу, представленному табл. 4.1, получим, что отфильтрованная мнимая компонента имеет вид $\dots 0, 0, 0, 0, 0, 0, 0, 0 \dots$. Аналогично можно получить, что зашумленная действительная компонента $\dots 1, 1, 1, 1, 1, 1, 1, \dots$ преобразуется (отфильтровывается) к виду $\dots 1, 1, 1, 1, 1, 1, 1 \dots$. Таким образом, шумовой выброс, размером в один пиксел отфильтровался нелинейным медианным фильтром с линейной апертурой размером $m=3$. Для фильтрации шумовых выбросов с размерами большими, чем один пиксел, необходимо брать фильтры, имеющие больший (чем 3) размер апертуры; при этом правила принятия решения, базирующиеся на принципах, представленных в табл. 4.1, легко выписываются.

Единственным параметром алгоритма, подлежащим выбору, является размер апертуры m . Его выбор трудно формализовать. Дело в том, что "размер" помехи, зашумляющей контур, зависит от ее (помехи) спектра частот, а частотные свойства таких помех к настоящему времени совершенно не исследованы. Рекомендации по выбору величины m могут быть следующими. При "размере" помехи равном m , апертура должна быть $m = m+2$. При этом отфильтровываются помехи меньшего "размера", а помехи большего "размера" выделяются.

5.1. ОСНОВНЫЕ ПОНЯТИЯ

Существует несколько, не сильно отличающихся между собой определений процедур сегментации. Наиболее распространенным является следующее определение: это представление исходного изображения на области (их количество заранее неизвестно), каждая из которых удовлетворяет некоторому критерию схожести [6]. Следовательно, одной из основных целей сегментации является удаление не используемой в дальнейшем (при описании, распознавании и анализе изображений) "лишней" видеоинформации. Упомянутый критерий схожести диктует разделение методов и алгоритмов сегментации на две большие группы: по критерию однородности и по критерию неоднородности [6]. К первой группе относятся методы разметки внутренних точек областей, методы пороговой обработки изображений, а также методы наращивания областей. Ко второй группе методов относятся методы выделения границ областей.

Широкое распространение первой группы методов обусловлено их высокой вычислительной эффективностью с одной стороны, и, с другой стороны, тем фактом, что часто характерным признаком наличия объекта на изображении является перепад яркости. Вторая группа методов применима в том случае, когда изображение содержит большое количество различных типов областей, и представление исходного изображения на однородные области становится практически невозможным. Исследуя (и используя) характер "локальных" неоднородностей на границе выделяемых (сегментируемых) областей, выделяют точки, предположительно являющиеся границами этих областей, с дальнейшим восстановлением связности границ. В результате появляется так называемый контурный препарат исходного изображения, однозначно разбивающий его (изображение) на сегментированные

Эффективность сегментации (иногда оцениваемая по числу ошибочно сегментированных пикселей) во многом зависит от выбранных исследователем признаков. Под признаками изображения обычно понимаются его простейшие отличительные особенности (количественные признаки) или свойства (качественные признаки).

Яркостные и гистограммные признаки являются наиболее важными и их значения положены в основу упомянутой первой группы методов сегментации. Контурные признаки, составляющие основу второй группы методов сегментации, характеризуются тем, что на границе между объектом и фоном (или между объектами) существуют изменения (локальные неоднородности) яркости.

5.2. МЕТОДЫ ПОРОГОВОЙ ОБРАБОТКИ ИЗОБРАЖЕНИЙ

5.2.1. Обработка изображений с постоянным порогом

Исторически первым методом, в котором использовалась гистограмма яркостей для пороговой обработки, является метод мод [6], заключающийся в том, что постоянный (глобальный, для всего изображения) порог находится в районе "впадины" гистограммы всего изображения, в свою очередь находящейся между двух мод, каждая из которых отражает распределение яркостей соответственно фона и выделяемого (или группы однородных по яркости) объекта (объектов).

К сожалению, в реальных изображениях все сложнее ("впадина" не глубокая, и к тому же она широкая и плоская) и поэтому модальная структура общей гистограммы выражена недостаточно. Для устранения указанных недостатков, в частности, производят аппроксимацию гистограммы смесью нормальных распределений [6]. Ясно, что для этого требуются большие вычислительные мощности, причем смесь нормальных распределений плохо аппроксимирует реальные моды [6].

Используют также подход, в основе которого лежит эвристически определяемая мера качества сегментированного изображения: это либо дискриминантный, либо энтропийный критерии [6]. В [6] в основу определения порога положен принцип неизменности моментов распределения после сегментации (по сравнению с моментами исходного) изображения.

В [6] рассмотрена возможность применения теории нечетких множеств. Также для устранения неярко выраженной модальности гистограммы всего изображения используют гистограмму подходящей совокупности точек. Такими совокупностями являются множество граничных и внутренних точек однородных по яркости областей. Поскольку число указанных точек невелико, то они соответствуют межмодальной впадине глобальной гистограммы. Если эти точки могут быть идентифицированы и они будут удалены при построении гистограммы, то впадина станет глубже и, следовательно, порог будет определен более точно [6].

Идентификация указанных точек может быть осуществлена с помощью операторов Лапласа или других операторов, аппроксимирующих величину градиента (см. гл.4).

Выявление указанных точек может быть осуществлено с помощью одной из структур представления видеоданных [6] – дерева квадрантов (см. гл. 2). Как указывалось ранее (гл. 2), в упомянутой структуре осуществляется итерационное (иерархическое) разбиение исходного изображения на квадранты. Прекращается дальнейшее разбиение на квадранты, если дисперсия яркостей, входящих в него пикселей, меньше заданной. Так как граничные точки дают большой вклад в дисперсию, то содержащие эти точки квадранты будут делиться до своего минимального размера (вплоть до пиксела). Множество точек таких квадрантов и принадлежит границе объект–фон [6]. В [6, 33] для целей сегментации рассматривается алгоритм на пирамидальных структурах (см. гл. 2).

Множество граничных точек можно и не удалять для построения глобальной гистограммы: можно строить гистограммы точек с большим значением градиента [6]. Если граница (разделяющая объект и фон) оказывается размытой, то гистограмма будет унимодальной, и порог может быть выбран либо по среднему значению полученной гистограммы, либо по ее медиане. В результате получилось два порога: один полученный без учета градиентных точек, а другой – с помощью градиентных точек. Значит можно производить классификацию объект – фон (т.е. производить сегментацию) в двумерном признаковом пространстве яркость – градиент [6].

Таким образом, совместный учет двух (трех и более) характеристик исходного изображения (яркость, градиент,

лапласиан и т.д.) может быть весьма полезен при процедуре сегментации.

Известен [6] метод преобразования гистограммы признаков второго порядка [6, 31], который определяется так: $P_{l,\alpha}(i,j)$ – относительная частота совместного появления уровней яркостей (i,j) , отстоящих друг от друга на расстоянии l в направлении α .

Пусть $P_{1,\alpha}(i,j)$ – относительная частота совместного появления уровней яркости (i,j) в пределах апертуры размером 3×3 . Ясно, что возможны четыре направления – $(0^\circ, 45^\circ, 90^\circ$ и $135^\circ)$. Элементы этих четырех гистограмм определяются в рассматриваемом случае выражениями [6]:

$$P_{1,0} = \frac{1}{2M(N-1)} \sum_{x=1}^N \sum_{y=1}^{M-1} O_{1,0}(x, y, i, j),$$

$$P_{1,45} = \frac{1}{2(M-1)(N-1)} \sum_{x=2}^N \sum_{y=1}^{M-1} Q_{1,45}(x, y, i, j),$$

$$P_{1,90} = \frac{1}{2N(M-1)} \sum_{x=1}^N \sum_{y=2}^M Q_{1,90}(x, y, i, j) \text{ и}$$

$$P_{1,135} = \frac{1}{2(M-1)(N-1)} \sum_{x=2}^N \sum_{y=2}^M Q_{1,135}(x, y, i, j),$$

где $Q_{1,\alpha}$ – логические выражения, принимающие значения либо ноль, либо единица по формулам:

$$Q_{1,0}(x, y, i, j) = [S(x, y) = i] \cap [S(x, y+1) = j],$$

$$Q_{1,45}(x, y, i, j) = [S(x, y) = i] \cap [S(x-1, y+1) = j],$$

$$Q_{1,90}(x, y, i, j) = [S(x, y) = i] \cap [S(x-1, y) = j],$$

$$Q_{1,135}(x, y, i, j) = [S(x, y) = i] \cap [S(x-1, y-1) = j].$$

Усредняя (по четырем направлениям) полученные матрицы, получим $P = \frac{1}{4} (P_{1,0} + P_{1,45} + P_{1,90} + P_{1,135})$. Рассмотрим диагональ полученной матрицы P , а также ближайшие к ней диагонали (в смысле расстояния $d(i,j) = |i-j|$:

$$d=0: \quad V_0(i) = P(i, i), \quad i = 1, \dots, 2^L;$$

$$d=1: \quad V_1(i) = [P(i-1, i) + P(i, i+1)]/2, \quad i = 1, \dots, 2^L-1;$$

$$d=2: \quad V_2(i) = P(i-1, i+1), \quad i = 2, \dots, 2^L-1;$$

$$d=3: \quad V_3(i) = [P(i-2, i+1) + P(i-1, i+2)]/2, \quad i = 3, \dots, 2^L-2;$$

$$d=4: \quad V_4(i) = P(i-2, i+2), \quad i = 3, \dots, 2^L-2.$$

Смысл введенных функций (гистограмм) $V_k(i)$ ($i=0, \dots, 3$) заключается в том, что хотя бы на одной из них присутствует пик (отсутствующий на обычной глобальной гистограмме), соответствующей объекту, площадь которого мала по отношению к площади фона. В [6] предложена комбинированная гистограмма $V_1 + V_2 + V_3$, имеющая более сглаженный вид.

Итак, изложенный метод сегментации состоит в следующем [6]: 1) для заданного числа уровней исходного изображения 2^L вычисляется матрица P совместного появления уровней яркости соседних пикселей исходного изображения; 2) получают сглаженную гистограмму вида $V_1 + V_2 + V_3$; 3) определяют порог по сглаженной гистограмме следующим образом:

находят два наибольших локальных максимума таких, что второй пик составляет не менее $k\%$ от главного (глобального пика), и при этом два пика отстоят друг от друга на расстоянии не меньше d (числа k и d определяются эвристически);

находят абсолютный минимум между двумя пиками, который и является порогом.

Глобальный порог можно определять и с помощью локальных признаков [6]. В этом случае для каждого уровня яркости $i = 1, \dots, 2^L$ вычисляется среднее значение градиента

$$g_i = \frac{1}{R_i} \sum_{(x,y) \in T_i} \left[\left(\frac{\partial S}{\partial x} \right)^2 + \left(\frac{\partial S}{\partial y} \right)^2 \right]^{1/2},$$

где T_i – множество точек, имеющих яркость i ; n_i – количество этих точек. В качестве порога берется то значение уровня яркости, для которого величина g_i максимальна. Так как этот уровень соответствует точкам с максимальным градиентом, этот уровень должен находиться на границе объекта с фоном.

Другой алгоритм (также градиентный, но итерационный) изложен в [6], в начале которого с помощью произвольного по-

ничной точки вычисляется средний уровень яркости в апертуре размером $1 \times m$, центрированной в граничной точке и расположенной перпендикулярно контуру границы. Для реальных изображений, имеющих размытые границы, такой средний уровень яркости является неплохим приближением к рациональному порогу. Итерационное вычисление таким образом среднего уровня (аппроксимирующего порог) в конечном счете дает стабильное решение [6]. Если же исходное изображение содержит более чем два класса однородных по яркости объектов, они могут сегментированы выбором другого начального порога; при этом также получается стабильное решение [6].

В [6] предложено следующее правило определения порога: наиболее рациональным является тот порог, который позволяет выделять больше высококонтрастных и меньше низкоконтрастных перепадов яркости. Указанное правило продиктовано желанием уменьшить потери истинных границ и уменьшить появление ложных границ. Количественным соотношением предложенного правила выбора порога является средний контраст перепадов яркости, выделяемых выбранным порогом. Чем больше выделяется высококонтрастных перепадов и чем меньше при этом выделяется низкоконтрастных перепадов, тем больше величина среднего контраста. Порог, соответствующий максимальному среднему контрасту, в [6] считается оптимальным.

Итак, пусть имеем две смежные точки $u_1 = (x_1, y_1)$ и $u_2 = (x_2, y_2)$, соответственно имеющие яркости $S(u_1)$ и $S(u_2)$. Пусть для конкретности $S(u_1) \leq S(u_2)$. Если порог c выделяет эти перепады, то он находится между $S(u_1)$ и $S(u_2)$ ($S(u_1) \leq c \leq S(u_2)$). Число перепадов, которые выделились с помощью порога c , равно $N(c) = \sum Q(u_1, u_2, c)$, где $Q(u_1, u_2, c) = 1$, если $S(u_1) \leq c \leq S(u_2)$ и равно нулю в противном случае.

Определим контраст между двумя точками u_1 и u_2 как $B(u_1, u_2) = |S(u_1) - S(u_2)|$.

Полный контраст при этом подсчитывается как $E(c) = \sum G(u_1, u_2, c)$, по всем перепадам, где $G(u_1, u_2, c) = B(u_1, u_2)$, если $S(u_1) \leq c \leq S(u_2)$ и равно нулю в противном случае. Средний контраст рассчитывается как $E(c)/N(c)$ (при $N(c) = 0$ средний контраст приравнивается к 0).

Описанный метод можно назвать методом равномерного контраста, так как разные значения порога c , удовлетворяю-

щие условию $S(u_1) \leq c \leq S(u_2)$, дают одинаковые значения контраста. Чтобы отдать предпочтение порогам, расположенным посредине отрезка $(S(u_1), S(u_2))$, понятие относительности контраста изменяется на значение [6]. В [6] изложен способ определения глобального порога, использующий непосредственно гистограммные признаки второго порядка.

Аналогично тому, как описывалось ранее, строится относительная частота совместного появления уровней яркости (i, j) , но только в горизонтальном $P_{1,0}$ и вертикальном $P_{1,90}$ направлениях. Аналогично формируется усредненная матрица $P = P_{1,0} + P_{1,90}$. В отличие от рассмотренного ранее случая матрица P используется не для модификации гистограмм яркости, а непосредственно для получения порога.

Произвольный порог c разбивает точки исходного изображения на два класса h_0 и h_1 . Указанному разбиению соответствуют четыре блока в матрице P (A_1, A_2, A_3 и A_4).

Вычислим величины A_i ($i = 1, \dots, 4$) как $A_1 = \sum_{i=1}^c \sum_{j=1}^c P(i, j)$,

$$A_2 = \sum_{i=1}^c \sum_{j=c+1}^N P(i, j), \quad A_3 = \sum_{i=c+1}^M \sum_{j=1}^c P(i, j) \text{ и } A_4 = \sum_{i=c+1}^M \sum_{j=i+1}^N P(i, j).$$

$$P_0 :=$$

	$\frac{h_0}{1, 2, \dots, c}$	$\frac{h_1}{c+1, \dots, N}$
$\frac{h_0}{1, 2, \dots, c}$	A_1	A_2
$\frac{h_1}{c+1, \dots, M}$	A_3	A_4

Величины A_i ($i = 1, \dots, 4$) представляют собой общее число переходов внутри h_0 , внутри h_1 , из h_0 в h_1 и из h_1 в h_0 .

Для определения порога рассчитывается величина

$$P(c) = \left(\frac{A_2 + A_3}{A_1 + A_2 + A_3 + A_4} \right),$$

минимальному значению которой и соответствует оптимальный порог [6].

Следуя [6], введем величину

$$e(x,y) = \max\{|S(x,y+1)-S(x,y-1)|, |S(x+1,y)-S(x-1,y)|\} -$$

максимальное из значений аппроксимаций производных в горизонтальном и вертикальном направлениях в точке (x,y) .

Образуем величины

$$G_1 = \sum_{x=1}^M \sum_{y=1}^N |S(x,y)e(x,y)| \text{ и } G_2 = \sum_{i=1}^M \sum_{j=1}^N |e(x,y)|.$$

Отношение указанных величин равно среднеарифметическому значений равномерной яркости объекта F_0 и фона F_1 , т.е. $G_1/G_0 = (F_0 + F_1)/2$. Полученная величина и берется в качестве порога.

В [6] изложены методы, обобщающие все изложенное выше.

5.2.2. Обработка изображений с переменным порогом

Необходимость в переменном пороге обусловлена нередко имеющейся неравномерностью освещения в пределах всего изображения. Обычно при этом отношения яркостей (объект/фон) сохраняются, но объекты и фон при этом в одних частях изображения могут быть темнее или светлее, чем в других. Если указанная неравномерность известна заранее, то она может быть компенсирована на этапе предварительной обработки (коррекция уровней яркости). После компенсации можно применять методы сегментации с постоянным порогом (п.5.2.1).

Если неравномерность освещения заранее не известна, то все изображение разбивают на блоки, в пределах которого выбирают порог. Если в каком-либо блоке имеется объект и фон, то гистограмма яркостей этого блока должна быть бимодальной, и ее впадина определяет локальный порог. Если же гистограмма яркостей унимодальна (в блоке содержится только объект либо только фон), то локальный порог определяется интерполяцией (к примеру, линейной) локальных порогов (по унимодальной гистограмме затруднительно определить локальный порог).

Изложенная идея реализована в [6] для скользящих блоков размером 7×7 пикселей. При каждом положении апертуры оценивается дисперсия яркостей находящихся в апертуре 49 пикселей: если дисперсия превышает некоторый заданный порог, то строится гистограмма яркостей (которая, как правило, бимодальна), по которой и определяется локальный порог.

В [6] полученная гистограмма (для более точной установки порога) аппроксимируется смесью двух нормальных распределений, минимум которой и соответствует локальному порогу. При этом в тех блоках, где гистограмма унимодальна, порог определяется линейной интерполяцией выбранных локальных порогов. Другой алгоритм определения переменного порога изложен в [6]. В квадратной апертуре размером 5×5 с центральным элементом, имеющим яркость S_{ij} выбирается 16 (из 25) пикселей с яркостями $S_1 \dots S_{16}$. Выбор этих пикселей показан на рис. 5.1.

Изложены два правила выбора порогов в точке (i,j) : применение того или иного зависит от значения максимальной яркости S_{max} . При $S_{max} < S_c$ (S_c – эмпирически определяемая константа) считается, что точка (i,j) принадлежит фону, если в апертуре есть точка с яркостью a_k ($k = 1, \dots, 4$) такой, что $S_{ij} - a_k < c_0$ (c_0 – предварительно заданная величина).

	S_1	S_2	S_3	
S_4	S_5		S_6	S_7
S_8		S_{ij}		S_9
S_{10}	S_{11}		S_{12}	S_{13}
	S_{14}	S_{15}	S_{16}	

Рис. 5.1. Схема выбора шестнадцати пикселей.

Если же $S_{max} \geq S_c$, то считается, что точка (i,j) также принадлежит фону, если по крайней мере для одной точки a_k в апертуре справедливо неравенство $S_{ij} < a_k / c_1$, где c_1 – также предварительно заданная величина. Ясно, что качество сегментации во многом зависит от выбора величин c , c_0 и c_1 [6].

Изложенному алгоритму присущ недостаток – он неадаптивен, так как в алгоритме не предусмотрен выбор указанных величин (c , c_0 и c_1) в процессе обработки изображения.

К применению методов сегментации с переменным порогом приводит задача выделения небольших по площади объектов, разбросанных по фону, который составляет большую часть изображения. В этом случае яркости пикселей, принадлежащих к объекту, не дают значительного вклада в пик гистограммы. Поэтому переходят к построению гистограмм нескольких, но

небольших фрагментов изображения; полученные гистограммы проверяют на бимодальность и устанавливают пороги так, как указано выше [6]. Для случая бинарных изображений вопросы сегментации областей сгущения рассмотрены в [51]. В [40] рассмотрены алгоритмы (и их математические модели) селекции по площади бинарных изображений.

5.3. НАРАЩИВАНИЕ ОБЛАСТЕЙ КАК МЕТОД СЕГМЕНТАЦИИ ИЗОБРАЖЕНИЙ

Наращивание областей есть процесс присоединения соседних точек к уже выделенным областям, если присоединяемые точки удовлетворяют выбранному критерию однородности [6].

Методы наращивания характеризуются следующими параметрами: способ выбора начальных точек; принцип однородности; способ просмотра пикселей изображения.

Одним из методов наращивания изображений является метод центроидного связывания [6], основанный на следующем принципе однородности. Если $|S_{ij} - c_m| < c$, то рассматриваемая точка принадлежит анализируемой области; c_m – текущее среднее значение яркости пикселей рассматриваемой области; c – некоторый порог, задаваемый эвристически. Процедура наращивания областей происходит до тех пор, пока не останется точки, удовлетворяющей упомянутому неравенству. Затем процедура повторяется, но с другой начальной точки.

К недостаткам метода центроидного связывания можно отнести следующие: зависимость результатов сегментации от выбора начальных точек; появление большого количества областей малой площади; появление незамкнутых контурных линий. В [6] рассмотрен алгоритм, предусматривающий наращивание одновременно нескольких областей. Вначале смежные точки объединяются в "атомарные" группы по принципу наличия одинаковой яркости. Границы "атомарных" областей при этом представляют элементарные линейные сегменты длиной l . Для каждого сегмента вычисляется модуль градиента яркости h . Если $h < c_h$ и при этом длина сегмента l удовлетворяет неравенству $(l/\min(P_i, P_j)) > c_l$ (c_h и c_l – эвристически устанавливаемые пороги, P_i, P_j – периметры смежных "атомарных" областей), то рассматриваемый участок границы стирается. Описанный алгоритм предотвращает слияние смежных

областей примерно одинаковой площади, при этом допускает поглощение меньших областей большими. Оставшаяся часть общей границы длиной l_0 удаляется при $l/l_0 > c_0$ (c_0 – эвристически устанавливаемая величина). Изложенный алгоритм обеспечивает хорошую сегментацию простых изображений с малым числом объектов и без текстуры.

Другие подходы к наращиванию областей изложены в [6]. Известен также алгоритм слияния–расщепления на пирамидальных структурах (см. гл. 2), в которых просмотр пикселей изображения организован более экономно [5, 6, 27].

Процедура сегментации производится следующим образом [5, 27]. Начиная с некоторого уровня l пирамиды производится анализ текущего блока размером 2×2 . Если все элементы этого блока удовлетворяют критерию однородности, то происходит переход на следующий $l+1$ – й уровень пирамиды (процесс слияния). Если критерий однородности не выполняется, то происходит процесс расщепления, т.е. переход на более низкий $l-1$ – й уровень пирамиды, и дальнейший анализ производится уже на $l-1$ – ом уровне пирамиды.

В результате первого этапа работы алгоритма появляется совокупность однородных квадратных блоков максимальной площади; при этом окончательной сегментации не происходит. На втором этапе (этапе группировки) выполняется слияние смежных блоков (не обязательно одного размера) на основе сходства их свойств [6, 27].

5.4. МЕТОДЫ ВЫДЕЛЕНИЯ ГРАНИЦ ОБЛАСТЕЙ

5.4.1. Метод выделения граничных точек

Основой для выделения граничных точек является анализ скорости изменения функции яркости.

Некоторые из алгоритмов (аппроксимирующие производные и градиенты) рассмотрены в предыдущей главе как алгоритмы контрастирования (подчеркивания) перепадов яркости. Там же рассмотрены алгоритмы (операторы Лапласа и другие) использующие аппроксимацию вторых производных функций яркости.

5.4.2. Методы формирования границ областей

Рассмотрим сначала методы "уточнения" границ, которые делятся на две категории [6]: итерационные и неитерационные. Итерационные методы в основном пригодны для бинарных изображений, полученных после выделения граничных точек.

Задачей уточнения является получение границы толщиной в один пиксел, причем полученная граница должна проходить посередине потенциальной границы. Известно несколько алгоритмов уточнения бинарных изображений [27]. Часть из этих алгоритмов подавляет потенциальные точки (при этом сохраняется связность контура) [6]; другая часть основывается на прослеживании линии [6]. Преимущество итерационных методов заключается в высоком качестве контурных препаратов, что достигается за счет больших вычислительных затрат.

Неитерационные алгоритмы работают быстрее, но для их работы требуется векторное поле признаков неоднородности яркости S_{ij} . Указанное поле получается в результате контрастирования перепадов и имеет вид

$$\bar{S}_{ij} = \{S_{ij}^1, S_{ij}^2, \dots, S_{ij}^M\}, \quad (i = \overline{1, M}, j = \overline{1, N}), \quad (i = 1, \dots, M, j = 1, \dots, N).$$

Размерность m вектора признаков неоднородности равна числу оцениваемых параметров принятой модели идеального перепада [6]. Компонентами указанного поля, как правило, являются: исходное изображение, поле градиента, поле вторых производных и т.д.

Представляется целесообразным использовать в качестве компоненты поле локальных оценок корреляционных моментов (учитывающих пространственные статистические связи), а также поле корреляционных моментов уровней (или битовых срезов) исходного изображения, учитывающих межуровневые (межбитовые) статистические связи. Ясно, что граница объект-фон (или объект-объект) будет подчеркнута полями корреляционных моментов, что в совокупности с данными поля градиентов улучшит процесс формирования границ областей. В [6] изложен итерационный алгоритм уточнения границ, суть которого заключается в последовательном подавлении потенциальных граничных точек, не удовлетворяющих условию тонкой линии. Проверка указанного условия производится сравнением бинарного изображения (размером 3×3) с набором из восьми апертур, представленных на рис. 5.2.

Весовому множителю представленных апертур, равному единице, должна соответствовать граничная точка. Хотя бы в одном из пикселей, обозначенных как a , также должна быть граничная точка. Изменение центрального пиксела S_{ij} не происходит, если конфигурация единичных весовых множителей совпадает с одной из апертур, представленных на рис. 5.2. Среднее количество итераций алгоритма равно четырем [6].

В алгоритме, изложенном в [6], требуется всего лишь две итерации, но требуется дополнительная память для хранения бинарной карты направлений локальных перепадов S_{ij} (0 соответствует горизонтальному направлению, а 1 вертикальному).

a	0	a
1	S_{ij}	1
1	1	1

a	0	0
1	S_{ij}	0
1	1	a

1	1	a
1	S_{ij}	0
1	1	a

1	1	a
1	S_{ij}	0
a	0	0
1	1	1
1	S_{ij}	1
a	0	a
a	1	1
0	S_{ij}	1
0	0	a
0	1	1
0	S_{ij}	1
a	1	1
a	0	a
0	S_{ij}	1
a	1	1

Рис. 5.2. Набор апертур итерационного алгоритма утончения границ

На первом этапе горизонтального утончения, для каждой строки i ($i=1, \dots, M$) на изображении S_{ij} ищутся все связанные одномерные фрагменты B_h длиной более одного пиксела, состоящие из таких точек (i, j) , что $S_{ij}=1$ и $S_{i+1, j}=1$. Для всех точек $(i, j) \in B_h$ $S_{ij}=0$, кроме центральной точки B_h , в которой сохраняется единичное значение $S_{ij}=1$. Второй этап осуществляет аналогично уже вертикальное утончение. При этом отыскиваются вертикальные одномерные фрагменты B_v , содержащие точки (i, j) такие, что $S_{ij}=1$ и $S_{i, j+1}=1$.

Перейдем к методам удаления разрывов, потребность которых возникает часто при получении контурного препарата из-за наличия помех, непостоянства величины перепада, неверного выбора порога и т.д.[6]

Известно несколько алгоритмов удаления разрывов [6]. Они, как правило, основываются на анализе геометрической конфигурации отдельных сегментов. Более эффективными являются алгоритмы, учитывающие значения яркости вблизи выделенных границ. Важными характеристиками алгоритмов удаления разрывов (алгоритмов связывания) является мера "соединимости" произвольной пары сегментов и правило принятия решения о соединении тех сегментов, которые максимизируют указанную меру "соединимости".

Рассмотрим алгоритм, изложенный в [6], в котором мера соединимости M определяется как $M = M_1 \cdot M_2 \cdot M_3 \cdot M_4$, где M_1 и M_2 – геометрические параметры (угол, расстояние), M_3 и M_4 – яркостные параметры (близость значений уровней яркости и яркость объекта). Ясно, что соединяемые сегменты должны иметь схожую ориентацию, т.е. эти сегменты должны образовывать тупой угол в диапазоне $[-\pi + \alpha, \pi - \alpha]$, где $0 \leq \alpha \leq 25^\circ$ [6]. Предпочтение отдается тем сегментам, которые образуют наименьший угол ϕ_0 , указанный в диапазоне (указанный угол измеряется со стороны, где находится яркий объект). Таким образом, $M_1 = 1 - (\phi - \phi_0) / (2\pi - \phi_0)$, если $\phi_0 \leq \phi \leq 2\pi - \phi_0$, или равно нулю в противном случае.

Расстояние между двумя соединяемыми сегментами не должно значительно превышать длины соединяемых сегментов. Для количественного выражения указанного условия введем отношения расстояний между концевыми точками a_0 и a_1 , b_0 , b_1 двух сегментов A и B (a_0 и b_0 наиболее близко расположенные точки)

$$R_1 = \frac{\overline{a_0 b_0}}{\overline{a_0 b_1}}, \quad R_2 = \frac{\overline{a_0 b_0}}{\overline{a_1 b_0}}, \quad R_3 = \frac{\overline{a_0 b_0}}{\overline{a_1 b_1}}.$$

Введем величину $r = (R_1 + R_2 + R_3) / 3$, и пусть r_0 – наименьшее из всех величин r среди всех сегментов B , претендующих быть соединенными с A . Тогда $M_2 = 1 - (r - r_0) / (1 - r_0)$.

Если сегменты A и B принадлежат границе одного и того же объекта, то значения уровней яркостей внутренних точек (вблизи сегментов A и B) должны быть близки. Количественным выражением этой близости может быть величина $M_3 = d_1 + d_2 + d_3$, $0 \leq M_3 \leq 3$, где $D_i = 1 - f_i / D$, если $f_i \leq D$, или равно нулю в противном случае,

$f_1 = |S_A - S_B|$, $f_2 = |S_A - S_R|$, $f_3 = |S_B - S_R|$; S_A , S_B , S_R – средние значения уровней яркости, измеренные в полосе шириной четыре пиксела на более светлой стороне сегментов A и B . D – величина порога, равная $\approx(15\ldots 25)\%$ от диапазона уровней яркостей обрабатываемого изображения. Параметр M_4 можно включать (а можно и не включать в общем случае) в выражение для меры "соединимости" затем, чтобы обеспечить связность контуров в чем-либо важных для исследователя объектов.

В [6] рассматривается задача выделения ярких и протяженных дорог и зданий. В указанном случае средняя яркость S_0 оценивается в полосе шириной в четыре пиксела на более светлой стороне 5% самых длинных сегментов.

$$\text{Величина } Q_A = \begin{cases} 1 & 0 \leq |S_A - S_D| \leq D, \\ 2 - |S_A - S_0|/D & D \leq |S_A - S_D| \leq 2D, \\ 0 & |S_A - S_D| > 2D, \end{cases}$$

количественно определяет принадлежность сегмента границы объекта, и [6] $M = Q_A Q_B$, где Q_B определяется аналогично величине Q_A . Величину M (равную $M_1 \cdot M_2 \cdot M_3 \cdot M_4$) можно определить и иначе, например вводя другие параметры, учитывающие какие-либо другие априорные сведения.

5.5. КОНТУРНАЯ СЕГМЕНТАЦИЯ

Обычно под контурной сегментацией понимают процедуры получения (прослеживания) контуров изображений объектов и кодирование контуров тем или иным способом [14, 31]. Под прослеживанием контуров понимается последовательное (от пиксела к пикселу) вычерчивание границы между объектом и фоном. Прослеживание контуров можно осуществлять разными алгоритмами, из которых наиболее известных два: алгоритм "жука" и алгоритм Розенфельда [14]. По всей видимости алгоритм Розенфельда более эффективен, так как этот алгоритм не приводит к заикливанию при любых конфигурациях контуров (если элемент изображения объекта имеет толщину в один пиксел, а длину более трех, то при применении алгоритма "жука" происходит заикливание и оконтуривание обрывается).

В [26] предложен оригинальный алгоритм выделения контуров бинарных изображений, сущность которого заключается в формировании двух упорядоченных массивов, совокупность которых и позволяет аппроксимировать контур.

В [29] исследовалась применимость использования f – статистики для выделения контуров протяженных объектов на текстурных изображениях. Изображение делится на сегменты (блоки), в каждом из которых оценивается дисперсия D_i уровней яркости всех пикселей, входящих в блок.

Статистика $f = D_i/D$, где D – оценка дисперсии уровней яркостей на участке изображения с заведомо отсутствующим изображением объекта. Если в исследуемый i -й блок попадает край объекта, то для этого участка элемент статистики f возрастает, что и является признаком наличия границы (и, следовательно, и контура).

Исследования дали неплохие результаты. Заметим, что все изложенные алгоритмы требовали полностью сформированного изображения.

В [55] предложен алгоритм формирования контура в процессе формирования самого изображения.

Алгоритм формирования цепного кода заключается в следующем. Берется апертура размером 3×3 таким образом (алгоритм работает в темпе поступления информации), что ее правый нижний элемент совпадает с последним правым пикселем нижней (текущей) строки. Указанная апертура перемещается вместе с правым пикселем нижней (текущей) строки. При каждом фиксированном положении апертуры информация во всех ее девяти ($3 \times 3 = 9$) пикселях анализируется. Если все девять пикселей апертуры заполнены единицами, то апертура находится внутри объекта и формирование кода контура не производится. Когда центральный элемент апертуры попадает на точку контура, в этой точке появляется возможность связывания отдельных элементов контура в цепочки (в виде кода Фримена).

Если все элементы, окружающие центральный элемент апертуры, имеют нулевые значения, то центральный элемент апертуры считается изолированной точкой (не принадлежащей контуру) и кодирование, естественно, не происходит.

Таким образом, совокупность всех возможных значений информации в восьми элементах апертуры (центральный элемент исключается) составляет $2^8 = 256$ вариантов, и осуществить свя-

звания отдельных элементов кода контура в цепочку следует из того, что в апертуре имеется область (ненулевые элементы), в которой расположены определенные элементы контура. Для получения замкнутого контура объекта отдельные цепочки, составляющие контур, соединяются между собой. Для любого из 256 вариантов расположения информации вокруг центрального элемента апертуры в процессе работы алгоритма осуществляется один из трех возможных вариантов: завязка нового контура; прослеживание контура; замыкание начатых контуров. Рассмотрим процесс завязки нового контура. Если текущая контурная точка расположена в центральном элементе апертуры, то в апертуре имеются еще две точки контура, которые по отношению к текущей точке при цепном описании контура являются предыдущей и последующей.

Если обе эти контурные точки располагаются в нижней части апертуры, т.е. там, где находятся необработанные элементы изображения, то завязывается новая контурная цепочка. Новая цепочка завязывается на верхних и нижних границах областей и полостей. Каждой новой цепочке присваивается свой номер, и этим номером характеризуются все элементы контура, входящие в эту цепочку.

От точки завязки цепочку можно проследить в обе стороны, поэтому в дальнейшем два конца цепочки прослеживаются как две отдельные цепочки, отмеченные одним номером. Различие этих цепочек между собой описано ранее. Перейдем к процессу прослеживания контура.

Все прослеживаемые цепочки делятся на два типа по виду элементарных векторов, входящих в них. Правые цепочки состоят из векторов 0, 1, 2, 3 (код Фримена) и названы так потому, что при обходе контура по часовой стрелке располагаются на правых и верхних границах объектов.

Левые цепочки состоят из векторов 4 – 7 и располагаются на левых и нижних границах объектов. Различие между цепочками заключается в том, что направление векторов правых цепочек совпадает с направлением сканирования, а направления векторов левых цепочек противоположны ему.

Это приводит к тому, что элементы правых цепочек выделяются последовательно друг за другом, в том же порядке, в каком они входят в цепной код контура, а элементы левых цепочек выделяются также последовательно друг за другом, но в цепной код контура входят в обратном порядке.

Три контурные точки в апертуре размером 3×3 определяют два вектора. Ситуация прослеживания точки возникает в случае, когда один из них уже включен в некоторую цепочку. Номер этой цепочки присваивается второму вектору, и таким образом он включается в цепочку.

В результате выполнения операций завязки и прослеживания образуется последовательность контуров сцены. Описание отдельного элемента контура включает в себя номер элементарного вектора и номер цепочки, к которой он принадлежит. Замыкание начатых контуров происходит следующим образом. Если в центре апертуры находится текущая контурная точка, а две другие контурные точки располагаются в верхней части апертуры, т.е. в области обработанных элементов изображения, то правая и левая цепочки замыкаются. Замыкание так же как и завязка цепочек, может происходить на верхних и нижних границах объектов и полостей. В момент замыкания запоминается номер левой цепочки, замкнувшейся в данной правой цепочке. Такой указатель замыканий формируется для каждой правой цепочки и позволяет в дальнейшем связывать их в замкнутый контур.

Рассмотрим связывание цепочек в замкнутый контур. Цепной код замкнутого контура формируется из отдельных цепочек следующим образом. По номеру цепочки из общей последовательности выбираются элементы соответствующей левой цепочки. Порядок их следования меняется на противоположный, и они присоединяются к цепному коду контура. Затем к цепному коду присоединяются элементы правой цепочки с таким же номером (порядок их следования не меняется). После этого по указателю замыканий определяется номер следующей левой цепочки. Эти действия повторяются до появления в указателе замыкания номера первой цепочки контура, что означает замыкание контура. При таком порядке связывания цепочек образуется цепной код, соответствующий направлению обхода контуров объектов по часовой стрелке, а контуров полостей – против часовой стрелки.

Таким образом описанный алгоритм позволяет формировать цепной код бинарных изображений в процессе формирования изображения. В зависимости от положения и размеров объекта, выигрыш во времени по сравнению с алгоритмом Розенфельда составляет от 1 до M ($M = N$ – размер изображения) [55].

6.1. ОСНОВНЫЕ ПОНЯТИЯ

Под сжатием видеоданных понимается такое их представление, которое требует памяти для хранения сжатых видеоданных не более исходной. Если употребляют термин кодирование видеоданных, то понимают под этим кодирование с целью последующего их сжатия. Возможность сжатия видеоданных обусловлена наличием в них большой избыточности, которую подразделяют на статистическую, психовизуальную и семантическую [11, 18]. Поскольку мы занимаемся машинной обработкой видеоданных, то вопросы устранения психовизуальной информации рассматривать не будем. Из сказанного ясно, что уже начиная с этапа предварительной обработки и кончая этапами распознавания образов и анализа сцен происходит то или иное сжатие видеоданных (устранение "лишней" информации из видеоданных).

К сжатию видеоданных можно отнести некоторые виды позиционного и структурного представления видеоданных (см. гл. 2), а также представление исходных изображений в виде символьных изображений (см. гл. 3). При этом происходит устранение статистической и семантической избыточности. Сказанное подтверждает не раз высказанную в этой книге мысль, что различные этапы переработки видеоданных тесно связаны между собой. Несмотря на сказанное, собственно проблема сжатия видеоданных с целью уменьшения необходимого объема памяти для хранения сжатых видеоданных существует сама по себе. Методов сжатия видеоданных разработано очень много. Мы же, учитывая цель и ограниченный объем книги, изложим лишь основные из них. Наиболее полно методы сжатия видеоданных изложены в монографии [31] и в обзорах [19, 23, 24]. Ясно, что кодирование должно быть таким, чтобы была сохранена информация, используемая на дальнейших этапах переработки изображений.

Наиболее простой метод сжатия видеоданных – это осуществление лишь черезстрочной или черезстолбцовой записи исходных данных в память машины, с последующей (при восстановлении исходного изображения) линейной интерполяцией яркостей пропущенных при записи строк или столбцов.

Все методы кодирования можно укрупненно подразделить (причем приводимая классификация далеко не единственная) на кодирование с импульсно-кодовой модуляцией (п. 6.2), статистическое кодирование (п. 6.3), кодирование с предсказанием (не рассматривается в настоящей книге), кодирование серий (п. 6.4), кодирование с преобразованием (п. 6.5), и ряд других методов (п. 6.6 и 6.7) [3].

6.2. ИМПУЛЬСНО-КОДОВОЕ КОДИРОВАНИЕ

Импульсно-кодовое кодирование есть представление видеоданных дискретизированных во времени (базируясь на известной теореме Котельникова) и по амплитуде (число уровней квантования обычно от 2^6 до 2^8).

Из сказанного ясно, что импульсно-кодовое кодирование само по себе не устраняет никакой избыточности. Уменьшения объема данных можно добиться, в частности, уменьшением числа уровней квантования: при этом возможно появление (из-за увеличивающихся скачков уровней яркости) ложных контуров, что нежелательно, так как контуры играют заметную роль во многих процессах переработки видеоданных.

Для борьбы с возникновением ложных контуров применяют так называемую модуляцию Робертса [31], заключающуюся в том, что перед операцией квантования к видеоинформации добавляют псевдошум (распределенный равномерно), который вычитают уже на выходе квантователя. При этом достигается значительный эффект: число уровней квантования, при котором практически не возникают ложные контура, можно снизить до $8=2^3$ [31], так как при модуляции Робертса резкие скачки уровней яркости трансформируются в менее резкие скачки. В [31] для борьбы с ложными контурами был предложен метод (квантование с улучшенной передачей градиций яркости), сущность которого заключается в том, что при квантовании текущего пиксела (яркость которого детерминированно связана с яркостью предшествующего пиксела) осуществляется малое возмущение яркости.

Теми же авторами предложен еще один метод борьбы с ложными контурами (метод с грубой – тонкой шкалой квантования), который предусматривает равномерное квантование яркостей каждого пиксела на $2^6=64$ уровня. При этом три старших разряда указанного шестиразрядного двоичного слова содержат грубую оценку уровня яркости каждого пиксела; три младших разряда содержат более тонкую информацию.

Если изменение уровней яркостей таково, что его достаточно закодировать тремя старшими разрядами, так и поступают. Если же изменение уровней яркости не приводит к изменению трех старших разрядов, то указанное изменение кодируется тремя младшими разрядами.

6.3. СТАТИСТИЧЕСКИЕ МЕТОДЫ КОДИРОВАНИЯ И СЖАТИЯ

Изображения обладают большой статистической избыточностью; об этом говорит тот факт, что на изображении есть много пикселей, для которых смежные с ним имеют одинаковый, или примерно одинаковый уровень яркости [11].

Этот же факт (большую статистическую избыточность) подтверждают и статистические данные о распределении яркостей пикселей и оценке энтропии реальных изображений.

Количественно избыточность в данном случае определяется как общее число битов, которые необходимо затратить, чтобы хранить данное изображение, за вычетом энтропии всего изображения [31].

При размерах изображения $M \times N$, равных 256×256 , и если каждый пиксел имеет $L=2^8=256$ уровней, то общее число всевозможных изображений равно $Q=10^{158000}$ [31].

Пусть известно (каким-либо образом измерено или оценено) распределение вероятностей p_i появления каждого из Q изображений $\left(\sum_{i=1}^Q p_i = 1 \right)$.

Энтропия ансамбля изображений (которая, как известно, равна минимальному среднему количеству бит, затрачиваемых на кодирование всего ансамбля) равна $H = - \sum_{i=1}^Q p_i \log p_i$.

Количество бит, требуемых на кодирование (и хранение) каждого изображения равно $(256 \times 256) \times 8$; энтропия H значительно ниже указанного числа [31], откуда и следует, что импульсно-кодовое кодирование обладает низкой эффективностью. Статистические методы кодирования и направлены на повышение эффективности кодирования и сжатия видеоданных. Возникает вопрос: существует ли метод кодирования, позволяющий так закодировать изображение, что количество бит, затрачиваемых на его хранение, было равно величине энтропии H ?

Теоретически такой метод существует. Практически же этого добиться нельзя по причине весьма большого числа Q , так как каждому из этих чисел (вероятности распределения которых p_i известны) в так называемой кодовой книге, составлен свой оптимальный код (код Хоффмана). Так вот, перебор этого громадного количества Q и его кодирование практически неосуществимо. Поэтому переходят к поэлементному кодированию одного, данного изображения на основе построенной (оцененной) гистограммы распределения яркостей этого данного изображения.

Кодирование будет эффективным, если уровню яркостей с большей вероятностью будет соответствовать малое (короткое) кодовое слово и наоборот. Для указанного кодирования существуют эффективные коды, наиболее известны из которых код Шеннона-Фано и код Хоффмана [31]. Последний всегда реализует наивысшую достижимую эффективность [31].

Заметим, что эффективность статистического кодирования существенно зависит от того, насколько "правильно" построена (оценена) гистограмма. На указанное обстоятельство необходимо обращать особое внимание.

Как уже указывалось, для реальных изображений характерна большая статистическая избыточность, т.е. между яркостями смежных пикселей существует весьма сильная вероятностная зависимость, что в свою очередь предполагает при статистическом кодировании учитывать условные вероятности вида $P(i/j)$, где условно через j обозначена яркость предыдущего, а через i яркость текущего пиксела. Кодирующая система поступает следующим образом: она обращается в кодовой книге к j -му уровню яркости предшествующего пиксела и выбирает из нее i -й уровень яркости текущего пиксела и соответствующий этому уровню яркости код.

Ясно, что при этом надо хранить в два раза больше кодовых слов, правда при этом повышается эффективность кодирования [31]. Процедуру кодирования можно усложнить и перейти к более полному учету вероятностных связей, т.е. учитывать условные вероятности вида $p(i/j,k)$, $p(i/j,k,l)$ и т.д.

Другой подход к кодированию с учетом яркости предшествующего пиксела заключается в кодировании разности уровней яркости соседних пикселов; уровень яркости первого пиксела кодируется независимо. При этом необходимо учитывать, что вероятность появления разностей с большой относительной величиной относительно мала.

6.4. КОДИРОВАНИЕ СЕРИЙ

Кодирование серий – это сравнительно простой метод кодирования. Серией называется последовательность пикселов, чьи уровни яркостей незначительно отличаются друг от друга. Поэтому целесообразно в памяти машины хранить какую-либо характеристику серии, например, адрес последнего элемента серии и уровень возникающего при этом (на следующем пикселе) скачка яркости. Можно хранить длину серии. Выбор этой или иной характеристики определяется конкретным изображением. Например, если серии длинные, то кодирование ее длины требует много битов; возможно при этом кодировать адрес последнего элемента серии (адрес конца серии) и т.д. Кодирование длин серии наиболее выгодно при бинарных изображениях. В этом случае достаточно указать уровень яркости начального элемента строки (или столбца) и далее кодировать адреса мест изменения уровней. Выбор, что же кодировать (длину серии или ее конец) в каждом случае решается индивидуально.

Кодирование серий дает хорошие результаты для факсимильных изображений. Кодирование же серий на дактилоскопических отпечатках (точнее их бинарных копий) положительного эффекта практически не дает. Каждое полутоновое изображение с числом уровней 2^L имеет L бинарных срезов, которые можно рассматривать как бинарные изображения и осуществлять кодирование серий каждого среза.

Эксперименты показали, что для ряда байтовых осмысленных и бессмысленных изображений шестой и седьмой срезы являлись негативными друг относительно друга.

Отсюда следует, что один из срезов (например, шестой) можно не хранить (и, конечно, не кодировать); восстановить же его можно (негативным отображением) из седьмого среза. Для тех же изображений первые и вторые битовые срезы состояли из одних единиц: еще один резерв для сжатия видеоданных. Можно также подсчитать коэффициенты корреляции (как нормированное число совпавших единиц) между всеми восьмью срезами, и из тех срезов, которые имеют весьма высокие коэффициенты корреляции, хранить один срез; второй срез при восстановлении изображения можно сгенерировать статистическими методами.

Заметим также, что преобразованием, приводящим к построению срединных осей, можно обеспечить эффективное кодирование бинарных изображений.

6.5. КОДИРОВАНИЕ С ПРЕОБРАЗОВАНИЕМ

Основная цель преобразования – переход от статистически связанных уровней яркости исходного изображения к относительно независимым коэффициентам. Сказанное эквивалентно устранению статистической избыточности исходного изображения [24]. Большинство известных преобразований являются линейными и унитарными. Поясним последнее свойство. Пусть матрица прямого преобразования есть A . Тогда матрица обратного преобразования есть A^{-1} . Унитарное свойство эквивалентно равенству $A^{-1} = A^T$, где матрица A^T является комплексно-сопряженной к транспонированной матрице A^T . Итак, унитарное свойство говорит о том, что обратное преобразование выполняется достаточно легко.

В вычислительном плане сложно произвести двумерное преобразование всего изображения, поэтому исходное изображение разбивается на сегменты малого размера (блоки), над которыми и осуществляется двумерное преобразование с коэффициентами менее статистически зависимыми, чем уровни яркостей исходного изображения [24].

Вследствие наличия статистических связей (статистической избыточности) между уровнями яркостей исходного изображения большая часть энергии преобразованного изображения концентрируется в относительно небольшом количестве коэффициентов преобразованного изображения.

Тогда сжатие достигается отбрасыванием небольших по величине коэффициентов (пороговый отбор коэффициентов), чем и достигается сравнительная эффективность сжатия кодирования с преобразованием.

Попробуем потребовать от преобразования, чтобы коэффициенты преобразования были статистически независимыми. Для выполнения этого условия по крайней мере необходимо знать статистические характеристики исходного изображения порядка выше второго, что затруднительно. Поэтому потребуем от коэффициентов преобразованного изображения их некоррелированности, так как для этого достаточно знать статистические характеристики исходного изображения порядка не выше второго.

Составим из сегмента T -компонентный вектор x и будем искать такое линейное преобразование (в виде матрицы A размером $T \times T$), которое обеспечивает получение T -компонентного вектора y с некоррелированными коэффициентами [24] $y = Ax$.

Так как сжатие заключается в отбрасывании некоторых коэффициентов, то необходимо, чтобы была минимизирована среднеквадратическая ошибка, возникающая из-за такого сжатия при восстановлении вектора x . Желательно также, чтобы энергия преобразованного изображения была сосредоточена по возможности в меньшем количестве коэффициентов.

Такое преобразование существует, и оно называется преобразованием Карунена-Лоэва [24].

Оптимальное (по перечисленным выше свойствам) преобразование можно осуществить, используя ковариацию вектора x $K_x = E\{(x - Ex)(x - Ex)^T\}$, где E – символ математического ожидания; верхний индекс T означает транспонирование.

Строки матрицы A являются нормированными собственными векторами матрицы K_x , т.е. являются решениями уравнения, $K_x x = \lambda x$. Коэффициенты преобразованного вектора y ($y = Ax$) имеют при этом ковариационную матрицу вида

$$K_y = \begin{vmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \dots & \dots & \dots & 0 \\ 0 & 0 & 0 & \lambda_T \end{vmatrix},$$

где $\lambda_1, \lambda_2, \dots, \lambda_T$ – собственные значения матрицы K_x .

Последнее и свидетельствует о некоррелированности преобразованного вектора x . Так как K_x – ковариационная матрица с неотрицательными собственными значениями ($\lambda_i \geq 0$, $i = 1, \dots, T$), то если эти собственные значения расположить по убыванию, то максимум энергии преобразованного вектора x сконцентрируется в первых K коэффициентах ($K < T$).

При хранении только K коэффициентов (соответствующим наибольшим собственным значениям) восстановление исходного значения (сегмента) произойдет с погрешностью, средне-квадратическое значение которой равно $\sum_{i=k+1}^T l_i$ [24].

Указанное значение погрешности мало, так как в нее входят собственные значения минимальной величины.

Применение указанного оптимального преобразования на практике затруднительно, т.к. чаще всего ковариационная функция исходного изображения нестационарна. Поэтому приходится брать различные ковариационные функции для различных сегментов исходного изображения. Можно взять одну, усредненную ковариационную функцию, но это приводит к потере эффективности [24].

Поэтому были предложены различные субоптимальные преобразования. Отказ от оптимальности заключается в том, что коэффициенты преобразования остаются все же коррелированными, а отбрасывание части коэффициентов (сжатия видеоданных) приводит к большей величине среднеквадратической ошибки [24].

Эти субоптимальные преобразования следующие [24]:
дискретное преобразование Фурье:

$$A = ||a_{ij}||, \quad a_{ij} = \frac{1}{\sqrt{T}} \exp[-2\pi\sqrt{-1}(ij)];$$

дискретное косинусное преобразование:

$$a_{ij} = \frac{2c(i)}{\sqrt{T}} \cos\left[\frac{(2j+1)i\pi}{2T}\right], \quad \text{где } c(i) = \begin{cases} 1/\sqrt{2}, & \text{при } i=1, \\ 1, & \text{при } i=2, \dots, T, \\ 1, & \text{иначе.} \end{cases}$$

симметрическое преобразование Адамара порядка $T=2^q$:

$$a_{ij} = \frac{1}{\sqrt{T}} (-1)^{b(i, j)}, \quad \text{где } b(i, j) = \sum_{l=0}^{q-1} i_l j_l.$$

В последней формуле i_l и j_l – это состояния разрядов двоичного представления чисел i и j . Из изложенного, например, следует, что матрица Адамара A размером 4×4 имеет вид

$$A = \begin{vmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{vmatrix}.$$

Кроме перечисленных, известны и другие преобразования, в частности наклонные, обладающие базисными векторами, являющиеся кусочно-линейными [24, 31]. В отличие от преобразования Карунена-Лозва наклонное преобразование не зависит от статистики исходного изображения и легко реализуется.

Дискретное преобразование Фурье может быть реализовано с помощью быстрого преобразования Фурье [8]. Преобразование Адамара реализуется наиболее просто (для его выполнения нужны лишь сложения).

Несмотря на неоптимальность изложенных преобразований, они обеспечивают высокую концентрацию энергии. Базисы указанных преобразований соответствуют как наиболее вероятным, так и наименее вероятным сегментам.

Поясним сказанное на примере преобразования Адамара с сегментом размера 4×4 : первой из базисных функций является квадрат (размером 4×4) и весь состоит из единиц (наиболее вероятный сегмент); последняя (шестнадцатая) из базисных функций также имеет размеры 4×4 , но имеет вид шахматной доски – наиболее хаотическая структура (наименее вероятный сегмент) [24].

Коэффициенты, получаемые в результате таких преобразований, хотя и коррелированы (в отличие от преобразования Карунена-Лозва), но и "более независимы", чем элементы исходного изображения. Возникает вопрос: насколько далеки указанные преобразования от оптимального? Для ответа на этот вопрос необходимо знать как статистические характеристики исходного изображения, так и способ оценки качества сжатого изображения.

В [24] ответ на этот вопрос получен для стационарного марковского исходного изображения с экспоненциальной

корреляционной функцией, разделяющейся на два сомножителя (по горизонтальному и вертикальному направлениям). Для указанных исходных данных обобщенный вывод свидетельствует о том, что наиболее близко к оптимальному дискретное косинусное преобразование. Сделанный вывод основывается, в частности, на том, что базисные функции дискретного косинусного преобразования и преобразования Карунена-Лоэва весьма схожи [23].

Все сказанное относится к двумерным сегментам. Однако возможна (а иногда и нужна) обработка и трехмерных сегментов, в которых роль третьего измерения играет время (совместная обработка последовательности кадров видеоданных).

В [24] описана схема обработки изображений в реальном масштабе времени с использованием блоков (трехмерных сегментов) размером $4 \times 4 \times 4$, а также моделирование на блоках размером $16 \times 16 \times 16$. Так как между соседними по времени пикселями существует статистическая связь, то кодирование с учетом последовательности кадров (трехмерные сегменты) дает значительное повышение эффективности [24].

Перейдем к сжатию, т.е. к отбрасыванию части коэффициентов преобразованного изображения. Самым простым методом (он уже упоминался) отбрасывания коэффициентов является пороговый: отбрасываются все коэффициенты, величина которых меньше заданного значения порога. Другой метод (его можно назвать спектральным) отбрасывания коэффициентов заключается в том, что из всего множества коэффициентов преобразования отбрасываются те, которые принадлежат "нужному" спектру. Например, оставляются коэффициенты (а остальные отбрасываются), составляющие низкочастотную область двумерного спектра (преобразования), или, наоборот, высокочастотную область и т.д.

Альтернативный метод отбора коэффициентов заключается в том, что вначале оцениваются дисперсии коэффициентов (по выборке, состоящей из "типовых" изображений), а затем отбрасываются все коэффициенты, дисперсия которых ниже определенного значения. Такой метод называется зональной фильтрацией. Естественно, что в зависимости от вида преобразований эффективность сжатия меняется.

В [24] приводятся данные для преобразования Адамара с сегментом 4×4 . Отбрасывание коэффициентов 11, 12..16 приводило лишь к незначительному искажению изображений. Наибольшая часть энергии была сосредоточена в первом

коэффициенте, дисперсия которого максимальна (дисперсия второго коэффициента составляет не более 10% от дисперсии этого первого коэффициента). Так что даже если отбросить все коэффициенты кроме первого, изображение удастся восстановить с достаточным качеством.

6.6. ИЕРАРХИЧЕСКИЙ МЕТОД КОДИРОВАНИЯ

В [25] изложены достаточно простые и эффективные методы кодирования без потерь как для многоуровневых, так и для бинарных изображений. Сущность метода заключается в том, что пользователю выдается иерархическая структура последовательности изображений такая, что каждой паре блоков, имеющих размеры от минимального до всего изображения с помощью специальной таблицы перекодирования ставится в соответствие величина, близкая к среднему значению яркости указанной пары блоков.

Во время сеанса с пользователем вначале извлекается первичное значение (полученное последним), а затем (в случае необходимости, определяемой пользователем) извлекаются величины, обеспечивающие все более и более детализированное изображение; так происходит до тех пор, пока пользователь не остановит этот показ иерархических изображений (пользователь "поймет" все изображение) либо пока не будет известно последнее (т.е. исходное) изображение. Сжатие (иногда значительное) достигается тем, что пользователь часто удовлетворяется промежуточным результатом, не доводя просмотр до последнего (т.е. исходного) изображения.

Таким образом, этот метод обладает достоинствами кодирования с преобразованием (см. п. 6.5), но не нуждается в обработке коэффициентов преобразования и не дает ошибок, вызванных отбором (пороговым, спектральным или зональным) коэффициентов.

Кодирование бинарных изображений также связано с иерархической структурным представлением видеоданных [27], но основная цель – сжатие видеоданных – достигается выявлением блоков по возможности большего размера, которые можно объявить сплошными черными и белыми и исключить из дальнейшей процедуры.

6.7. УСЕЧЕННОЕ БЛОЧНОЕ КОДИРОВАНИЕ

Суть усеченного блочного кодирования заключается в том [22], что все изображение (размером $M \times N$) разбивается на непересекающиеся блоки, каждый размером $m \times n$ пикселей. Описываемый метод подстраивает параметры квантования не под некоторую усредненную характеристику (например, глобального порога), а под локальные особенности каждого блока.

Это и позволяет сохранить мелкие детали изображений, что особенно важно в задачах выделения и распознавания (классификации) объектов. Метод также не приводит к размыванию границ. Каждый из указанных блоков обрабатывается независимо, поэтому кратко изложим суть обработки одного блока, заключающуюся в вычислении порога и двух уровней квантования, после которого и происходит квантование блока на два уровня (светлый и темный).

Порог c вычисляется как среднее значение всего блока; верхний a и нижний b уровни квантования вычисляются из условия сохранения (приблизленно) первых двух выборочных моментов, что приводит к следующим расчетным формулам [22]

$$a = c + d[(k-q)/q]^S, \quad b = c - d[q(k-q)]^S,$$

где d – оценка среднеквадратичного значения по данным всех $k = m \times n$ элементов блока; q – число элементов блока, превышающих порог. Фактически мы переходим к бинарным (двухуровневым) блокам.

Восстановление закодированного изображения происходит в обратном порядке: вместо нулей записывается нижний уровень квантования, вместо единиц – верхний.

Описанный выше способ определения верхнего и нижнего уровней квантования по критерию сохранения первых двух выборочных моментов не является единственным. В [21] проанализирован ряд других критериев.

6.8. ЗАКЛЮЧЕНИЕ

В [23] рассмотрены методы кодирования с учетом движения объектов, причем рассмотрены рекурсивные алгоритмы [22] и алгоритмы оценивания двумерной корреляционной функции

[23]. Там же рассмотрены обобщения кодирования с преобразованием в виде адаптивного кодирования, т.е. квантования и кодирования коэффициентов преобразования с управлением на основе содержания исходного изображения, а также проведено исследование других преобразований. В п. 6.5 рассмотрено кодирование с преобразованием при разбиении исходного изображения на непересекающиеся сегменты (блоки), в результате не учитываются статистические связи за пределами межблочных границ. В [22,25] осуществлено кодирование перекрывающимися сегментами, что приводит к увеличению числа блоков, но заметность границ блоков снижается.

В [19] рассмотрены методы (названные методами второго поколения), основанные на контурно-текстурных описаниях, суть которых заключается в расчленении исходного изображения на текстурные области, окруженные контурами так, чтобы эти контуры по возможности соответствовали контурам объектов на исходном изображении. Контурная и текстурная информация кодируется по отдельности. Контуры выделяются методами наращивания областей (см. п. 5.3).

Глава 7

ПРИНЦИПЫ ВЫДЕЛЕНИЯ ПРИЗНАКОВ И РАСПОЗНАВАНИЕ ОБРАЗОВ

7.1. ОСНОВНЫЕ ПОНЯТИЯ

Термин "распознавание образов" (как и многие термины в обработке и анализе изображений) является несколько расплывчатым. В этой книге под распознаванием образов понимаются совокупность методов и алгоритмов, позволяющих по данным ряда измерений отличительных характеристик (признаков) объекта, предъявляемого к распознаванию, вынести решение о его принадлежности к одному из N ($N > 1$) классов.

Из сказанного сразу вытекают две проблемы распознавания образов:

- 1) проблема определения и извлечения (измерения) признаков;
- 2) проблема принятия решения о принадлежности предъявленного объекта (в виде вектора уже измеренных признаков) к одному из N классов.

В распознавании образов, (точнее, во второй его проблеме – принятие решения) существует два понятия: идентификация и классификация, которые не имеют однозначного толкования.

Мы придерживаемся толкования, данного в [33]. Итак, под идентификацией объекта понимается такое вынесение решения об объекте, которое присваивает объекту однозначное название.

Под классификацией объекта понимается такое вынесение решения о нем, которое указывает лишь на принадлежность объекта некоторому классу объектов. В тех случаях, когда каждый класс содержит только один объект, понятия идентификации и классификации равнозначны.

* Проблемы обучения и связанные с ней вопросы в настоящей книге не рассматриваются.

7.2. ПРИНЦИПЫ ВЫДЕЛЕНИЯ ПРИЗНАКОВ

Основой вопрос при определении признаков заключается в том, чтобы определить какие (и сколько) признаков необходимо выделить для надежной классификации объектов. Общей теории формализованного определения таких признаков к настоящему времени не существует. В основном необходимо базироваться на опыте и интуиции (основных принципах научного познания вообще).

При определении признаков следует руководствоваться одним наиболее устойчивым принципом: обладает ли исследуемый (а в дальнейшем и классифицируемый) объект некоторой регулярностью, и если да, то эту регулярность необходимо положить в основу формирования признаков. Причем необходимо учитывать, чтобы эта регулярность была присуща как всем объектам, принадлежащим N классам, так и всем N классам. Например, при различении (классификации) монет достоинством в две и пятнадцать копеек регулярности две: размер и масса монет. Однако масса позволяет более надежно различать эти две монеты, так как массы различаются больше, чем размеры.

Приведенный пример характерен тем, что взятые признаки (масса и размер) хорошо поддаются определению и легко интерпретируются в терминах самих предъявляемых образов (монет) [12]. Пример не относится к обсуждаемой в этой книге проблеме обработки и анализа изображений и дан лишь для иллюстрации общего принципа подхода к выбору признаков.

Другой пример (имеющий отношение уже к изображениям) связан с классификацией (распознаванием) автомобилей (на грузовые и легковые) [33]. В этом случае характерными регулярностями будут длина и высота, измерив которые, легко различить легковые автомобили и грузовые.

В более сложных ситуациях возникают и более сложные признаки, не так непосредственно интерпретируемые, как рассмотренные выше.

При обработке (предварительной и сегментации) изображений обычно выделяются пространственно распределенные объекты с такими характеристиками, как форма, яркость, текстура, цвет (при обработке цветных изображений). Сам результат обработки и диктует выбор признаков: контурные признаки, признаки формы, аппроксимация посредством мо-

ментов, текстурные признаки, яркостные признаки и т.д. Еще более сложный случай имеет место тогда, когда признаки явно не визуализируемы; тогда их нужно искать (их можно и не найти) в более скрытых регулярностях (закономерностях), например в отыскании различных статистических связей (а они всегда есть, даже при детерминированных объектах) между уровнями значений яркости выделенных (или выделяемых) объектов. Этими связями, в частности, могут быть (см. гл.5) оценки коэффициентов корреляции не только по пространственным координатам, но и по различным уровням полутонового изображения (включая битовые срезы).

Учитывая сказанное, а также то, что (на это обращалось внимание и ранее, см. гл. 1) изображения обычно зашумлены; следовательно подвержены шумовым воздействиям и выделенные (сегментированные) объекты. И уж, конечно, оказываются зашумленными (также и в результате их измерения) и измеренные признаки, что дает полное право отнести их к случайным величинам.

Следовательно, для процедур вынесения решений (классификации) целесообразно использовать статистические методы классификации (см. п. 7.3). В некоторых методах выделения признаков (разложение в ряд естественной линии, аппроксимация посредством моментов) их число либо счетно, либо просто много. Поэтому возникают две задачи: 1) задача уменьшения количества признаков; 2) задача упорядочивания признаков.

Для уменьшения количества признаков (когда их количество менее, чем счетно) используют известные в статистике принципы уменьшения признакового пространства [28].

Можно, в принципе, уменьшить размерность признакового пространства до одного, просто проектируя многомерные данные на прямую [14]. Однако этот метод вряд ли даст удовлетворительные результаты, так как при таком проектировании все данные могут (и это часто происходит) перемещаться и разделение на классы становится просто невозможно.

Можно, однако, вращать эту прямую; при этом спроектированные данные могут дать удовлетворительные результаты, что является скорее исключением, чем правилом [14].

* Заметим, что сегментацию (гл. 5) можно определить как разбиение изображения на части, поддающиеся единому описанию в пространстве выбранных признаков.

Сказанное является целью классического дискриминантного анализа [14]. Поэтому обычно увеличивают размерность (от одномерного) признакового пространства, на которое проектируется исходное многомерное пространство.

В [28] изложены шесть основных принципов снижения размерности признакового пространства.

Вторая задача (задача упорядочивания признаков) возникает, в частности, тогда, когда необходимо из полного набора признаков ранжировать их по степени их информативности. В [41] рассмотрены теоретико-информационный подход упорядочивания признаков, и упорядочивание признаков с помощью разложения Карунена-Лоэва.

В статистических методах классификации довольно часто используют предположение о нормальном распределении вектора признаков, чем и вызвано применение мощного аппарата многомерного статистического анализа. На наш взгляд признаки, извлеченные из изображений, вряд ли распределены нормально.

Сказанное следует хотя бы из того, что исходное изображение дискретно, и формируемые признаки либо сами дискретны (например, при кодировании контуров), либо формулы их получения (например, аппроксимация посредством моментов) не дают оснований (скажем, в виде центральной предельной теоремы теории вероятностей) считать их нормально распределенными. Это усложняет проблему принятия решений с помощью статистических методов, так как последние подробно исследованы в предположении о нормальном расположении признаков.

Наконец, как указывалось ранее (гл. 3), при обработке запросов по определению пространственных соотношений (пониманию изображений) результатом запроса может быть необходимость предъявления синтезированного (методом машинной графики) изображения пользователю.

Указанный синтез (по данным символьного изображения) может быть произведен только при наличии признаков сегментированных и классифицированных объектов, имевшихся на исходном изображении. Отсюда следует, что необходимо хранить и все признаки всех выделенных объектов (если конечно в памяти машины не хранятся исходные или обработанные изображения).

7.3. СТАТИСТИЧЕСКИЕ МЕТОДЫ КЛАССИФИКАЦИИ

Как указывалось выше, статистические методы принятия решений (точнее, методы статистической теории принятия решений) применяются при классификации тех объектов, у которых признаки представляют собой случайные величины. Основой для применения классической статистической теории принятия решений является знание законов распределения вероятностей вектора признаков, что не всегда обеспечивается.

Процедура принятия решений в самом общем виде заключается в следующем. Каждый из N классов представлен в Q -мерном (Q – размерность вектора признаков) пространстве, называемом признаковым пространством. Каждый объект в этом случае задается точкой Q -мерного пространства. Считается (и на этом базируются практически все статистические методы классификации и методы кластеризации), что все объекты одного класса группируются в Q -мерном признаковом пространстве компактно (так называемая гипотеза компактности [4]); поэтому расстояние (введенное тем или иным образом) между объектами одного класса меньше, чем расстояние между объектами различных классов. Группировка объектов одного класса происходит вокруг центра, который определяется, чаще всего, математическим ожиданием вектора признаков. Увеличение интенсивности шумов приводит к уменьшению компактности (или, как иногда говорят [33], к увеличению характеристического объема), которое может привести к увеличению расстояний между объектами одного класса по сравнению с расстояниями между объектами разных классов, что приводит к появлению ошибок классификации.

Собственно основные усилия специалистов, занимающихся классификацией методами статистической теории принятия решений и были направлены с одной стороны на получение методов вычисления этих ошибок (в виде вероятностей перепутывания классов), а с другой стороны – на получение таких методов, разбивающих Q -мерное пространство признаков на N непересекающихся подмножеств, которые указанные ошибки позволяют либо уменьшить, либо вообще минимизировать [33].

Наиболее полно эта задача решена с применением байесовского правила принятия решения при распределении вероят-

ностей признаков, описываемых многомерным нормальным законом распределения вероятностей [14, 28]. Заметим, что в этом случае требуется знание априорных вероятностей p_i ($i = 1, \dots, M$) появления классов, которые (особенно при обработке изображений) в большинстве случаев неизвестны. Поэтому были предложены методы, требующие меньших априорных сведений о классах: в первую очередь разрабатывались методы, не требующие знания априорных вероятностей p_i ($i = 1, \dots, M$).

В предположении нормальности распределения признаков и зная ковариационные матрицы признаков, в [33] изложены две группы методов: разделение с помощью квадратичных функций (если ковариационные матрицы всех классов различны); разделение с помощью линейных функций (если ковариационные матрицы всех классов одинаковы).

Последняя группа методов известна как линейная разделяющая функция Фишера [33].

Следующие две группы методов не требуют знания не только законов распределения вероятностей признаков, но и знания ковариационных матриц. Это [33]: правило ближайшего соседа; иерархическое разделение. В случае применения правила "ближайшего соседа" объект считается принадлежащим к тому классу, к которому принадлежит его ближайший (по выбранному расстоянию) "сосед" или кто-либо из ближайших соседей.

При иерархическом разделении используется древовидная структура следующим образом [14, 33]: вначале в каждой вершине дерева исследуется один из Q признаков, и в зависимости от его значения выбирается следующая ветвь. Последовательно применяя это правило, окончательная классификация происходит в нижней вершине дерева [33]. Более подробно указанная процедура, и ее модификации рассмотрены в [14].

7.4. ПРИНЦИПЫ КЛАСТЕРНОГО АНАЛИЗА

Для кластерного анализа также не существует точного определения. Одно из определений следующее [15, 33]: кластерным анализом называют анализ неклассифицированных объектов, проводимый с целью выделения классов чем-то схожих между собой (кластеров). Задачу кластеризации можно трактовать как и задачу сегментации, но не исходного изображения, а данных, представленных в виде точек (это могут быть и признаки).

В кластерном анализе не существует однозначного количественного критерия (типа вероятности перенутывания в статистических методах классификации), характеризующего процесс кластеризации в целом [33]. Известны сотни алгоритмов кластеризации [15]. В [33] изложены три разновидности кластерного анализа: 1) методы основанные на отыскании максимумов (мод) распределения вероятностей источника, порождающего данные, подлежащие кластеризации. Каждый максимум и соответствует одному из кластеров; 2) методы, в которых в качестве критерия используется отношение внутрикластерной дисперсии и межкластерной дисперсии; 3) методы, использующие иерархические схемы кластеризации.

Вторая группа методов подробно изложена в [15] а первая и третья в [14, 33]. Изложим один из простейших методов кластеризации [14], суть которого заключается в том, что по данным Q -признаков вычисляется матрица (размерностью $Q \times Q$) всевозможных расстояний d_{ij} ($i, j = 1, \dots, M$) между всеми признаками. Полученная матрица является симметричной с нулевыми диагональными элементами.

Исходя из предположения о справедливости гипотезы компактности, матрица расстояний отреагирует на эту гипотезу таким образом, что данные, относящиеся к одному из кластеров, будут иметь сравнительно небольшие взаимные расстояния, и наоборот. Выбирая (эвристически) некоторый порог c , получим бинарную матрицу, образованную следующим образом

$$d_{ij} = \begin{cases} 1, & \text{если } d_{ij} \leq c, \\ 0, & \text{если } d_{ij} > c. \end{cases}$$

Полученная бинарная матрица проявит кластеры следующим образом: к одному кластеру относятся те данные, которым в бинарной матрице соответствуют одинаковые строки. Таким образом и произойдет автоматическая кластеризация исходных данных на заранее неизвестное число кластеров.

В случае "сильно" зашумленных исходных данных в бинарной матрице могут не появиться абсолютно одинаковые строки (либо их число будет относительно невелико).

В этом случае необходима корректировка решающего правила, заключающаяся в том, что строки считаются одинаковыми (т.е. соответствующие этим строкам данные относятся к одному кластеру), если число совпадающих единиц равно заданному числу (это число также выбирается эвристически).

7.5. НЕКОТОРЫЕ МЕТОДЫ ВЫЧИСЛИТЕЛЬНОЙ ГЕОМЕТРИИ В ЗАДАЧАХ КЛАСТЕРНОГО АНАЛИЗА

Многие задачи кластерного анализа можно трактовать [30] как задачи нахождения ближайших точек. Перечислим некоторые из этих задач, для которых в [30] изложены алгоритмы их решения с оценкой вычислительной сложности.

Эти задачи следующие. На плоскости заданы N точек. Найти две из них, расстояние между которыми наименьшее. На плоскости заданы N точек. Найти ближайшего соседа для каждой из N точек. На плоскости даны N точек. Построить дерево, вершинами которого будут все N точек, и суммарная длина всех ребер минимальна.

Перечисленные задачи обобщаются на случай Q -мерного пространства ($Q > 2$).

Для задач кластеризации характерна следующая задача, известная в вычислительной геометрии как задача построения диаграммы Вороного [30]. Эта задача и алгоритм ее решения следующие [30]. На плоскости задано множество A , содержащее N точек. Требуется для каждой точки a_i ($i = 1, \dots, N$) множества A построить locus (геометрическое множество) точек (x, y) на плоскости, для которых расстояние до a_i меньше, чем до любой другой точки множества A .

Определим структуру такого разбиения плоскости. Для двух точек a_i и a_j ($i, j = 1, \dots, N$) множество точек, более близких к a_i , чем к a_j , есть полуплоскость, определяемая прямой, перпендикулярной отрезку $a_i a_j$ и делящей его пополам, и содержащая точку a_i . Обозначим эту полуплоскость как $H(a_i, a_j)$. Множество точек, более близких к a_i , чем к любой другой точке [которое обозначим как $V(i)$], получается в результате пересечения $N-1$ полуплоскостей. Это множество является выпуклой многогранной областью, имеющей не более $N-1$ сторон. Полученная указанным способом область $V(i)$ и называется многоугольником Вороного, соответствующая точке a_i ($i = 1, \dots, N$). Получаемые таким образом N областей образуют разбиение плоскости, представляющую некоторую сеть, которую и называют диаграммой Вороного [30]. Диаграмму Вороного можно определить для пространства любой размерности.

7.6. АЛГОРИТМ КЛАССИФИКАЦИИ В СЛУЧАЕ ОПИСАНИЯ ОБЪЕКТОВ КОДОМ ФРИМЕНА

В гл. 2 указывалось, что кодирование контуров объектов с помощью кода Фримена позволяет компактно описывать контур в виде последовательности цифр, каждая из которых может принимать восемь значений (от 0 до 7). По этим данным можно составить эталон объекта следующим образом. Подсчитывается относительная частота появления в закодированном контуре каждого направления в виде n_i / N , где n_i – число элементов контура, имеющих i -е направление ($i = 1, \dots, 7$), N – общее число элементов, описывающих весь контур ($\sum_{i=0}^7 n_i = N$). Эталон объекта представляет собой восьмимерный вектор вида

$$\left| \frac{n_0}{N}, \frac{n_1}{N}, \dots, \frac{n_7}{N} \right|.$$

При классификации предъявленного объекта аналогично составляется описанный вектор, который и сравнивается со всеми эталонными векторами. Решение принимается в пользу того класса, в котором расстояние между предъявленным вектором и векторами всех эталонов минимально [44].

7.7. СИНТАКСИЧЕСКИЕ МЕТОДЫ В РАСПОЗНАВАНИИ ОБРАЗОВ

Синтаксические методы (или лингвистически-структурные) применяются тогда, когда признаки могут быть представлены неприводимыми элементами (подобразами) и их связями (отношениями). Выбор неприводимых элементов, так же как и выбор признаков при статистических методах классификации (см. п. 7.2) не всегда очевиден и прост. И здесь на помощь должны приходить опыт и интуиция, так как общей теории, позволяющей формализовать процедуры выбора неприводимых элементов, в настоящее время нет. Основные принципы, на которых базируется выбор неприводимых элементов, заключаются в следующем [33, 38]: 1) каждый из выбранных неприводимых элементов не должен давать представление об анализируемом объекте в целом; 2) неприводимые элементы и

их комбинации должны позволять производить простые описания анализируемых объектов. Для описания объекта через его непроеизводные элементы (и их отношения) обычно используют специальные языки [33, 38]. Заметим, что уже разработаны специализированные языки описания изображений [36, 42]. Подробно синтаксические методы изложены в [38]. Мы излагаем материал, ориентируясь на работу [33]. Правила указанных специальных языков, позволяющих составлять образы из непроеизводных элементов, называются грамматикой.

При этом составленный образ называется предложением соответствующей грамматики. Для распознавания объекта, представленного предложением, необходимо в первую очередь выделить (сегментировать) непроеизводные элементы, что само по себе представляет нетривиальную задачу переработки видеоданных. После того, как непроеизводные элементы будут выделены, необходимо провести синтаксический анализ имеющегося предложения. Предложение начинается с некоторого начального символа S , принадлежащего множеству вспомогательных символов V_A . Имеется также набор правил R , позволяющих преобразовать начальный символ S в некоторую цепочку символов, например $S \rightarrow aA$, (это означает что символ S заменен символами aA). В указанную цепочку могут входить как символы из множества V_A , так и символы, принадлежащие множеству непроеизводных (терминальных) элементов V_P [33]. Предложение считается построенным, если оно состоит только из непроеизводных элементов. Рассмотрим пример с $V_P = \{a, b, c, d\}$, $V_A = \{S, A, B, C, D\}$ и правилом $R = \{S \rightarrow aA, S \rightarrow bB, S \rightarrow cC, S \rightarrow dD, A \rightarrow b, B \rightarrow c, C \rightarrow d, D \rightarrow a\}$. Если непроеизводные элементы $\{a, b, c, d\}$ имеют вид, изображенный на рис. 7.1,

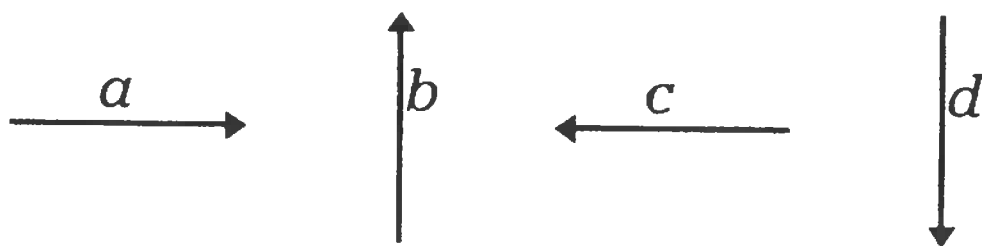


Рис. 7.1. Нпроеизводные элементы a, b, c, d

то следующие четыре предложения (знак \rightarrow обозначает вывод) $S \rightarrow aA \rightarrow ab$, $S \rightarrow bB \rightarrow bc$, $S \rightarrow cC \rightarrow cd$ и $S \rightarrow dD \rightarrow da$ описывают четыре прямых угла (рис. 7.2). Применяя грамматики с более сложными правилами, можно породить более сложные образы.

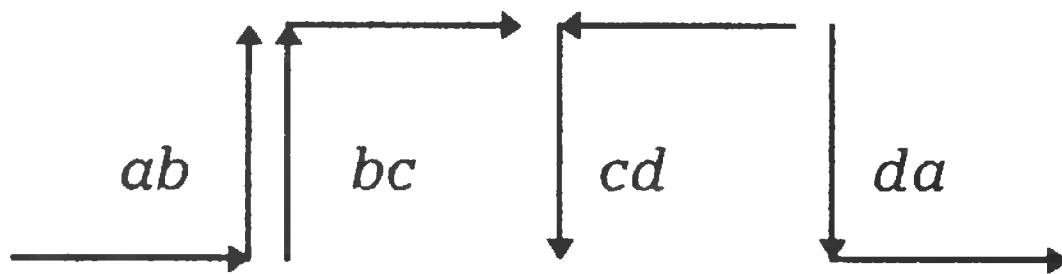


Рис. 7.2. Четыре образа прямого угла

Рассмотрим еще более сложный пример: $V_A = \{S, A, B, C, D\}$, $V_P = \{a, b, c\}$. Множество правил R следующее: $R = \{S \rightarrow aSBC, S \rightarrow abC, CB \rightarrow CD, CD \rightarrow BD, BD \rightarrow BC, bB \rightarrow bb, C \rightarrow c\}$. Например, предложение abc получается как $S \rightarrow abC \rightarrow abc$. Нетрудно видеть, что данная грамматика порождает предложения типа $\{a^n b^n c^n \mid n \geq 1\}$. Если непроеизводные $\{a, b, c\}$ элементы имеют вид, изображенный на рис. 7.3, то данная грамматика описывает прямоугольные треугольники разных размеров.

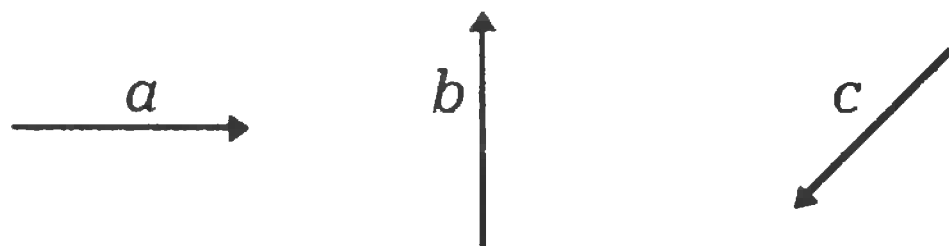


Рис. 7.3. Непроизводные элементы a, b, c

Для переработки зашумленных изображений используются стохастические (вероятностные) грамматики с заданными вероятностями p_{ij} замены некоторой строки a_i некоторой строкой b_j . В [42] показано, что математическими моделями одношагового вида являются конечные дискретные поглощающие цепи Маркова. Многошаговые выводы и его характеристики можно получить, используя методы, изложенные в [46]. Распознавание образов, заданных в виде предложения, означает определить, является ли предъявленное предложение синтаксически правильным, а значит и определить, принадлежит ли предложенный объект к объектам, задаваемым данной грамматикой.

Если предъявленный образ не определяется с помощью данной грамматики, то этот образ из классификации либо исключается, либо используется другая грамматика.

Приведем синтаксический разбор первого из двух изложенных выше примеров. Непроизводный элемент a (из предложения ab) можно получить как с помощью правила $D \rightarrow a$, так и с помощью правила $S \rightarrow aA$; элемент D порождается правилом $S \rightarrow dD$, а значит не приводит к получению предложения ab . Так

как $S \rightarrow aA$ в сочетании с $A \rightarrow b$ действительно позволяет получить предложение ab , то это предложение допустимо. Предложения ac и ad отклоняются.

7.8. СРАВНЕНИЕ ДВУХ ПОДХОДОВ

Многие методы предварительной обработки и сегментации изображений полезны в обоих подходах. Это связано с тем, что выбор конкретного алгоритма зависит скорее от вида выделяемых объектов и степени зашумленности изображений, чем от применяемого в дальнейшем подхода [42].

Задачи отбора и выделения признаков при статистическом подходе схожи с задачами выделения непроеизводных элементов. Различие заключается в том, что непроеизводные элементы составляют подобразы (кирпичики, из которых состоит образ), а признаками могут быть любое количество (вплоть до счетного) характеристик объекта. На этом уровне нет различий в двух подходах, так как сами непроеизводные элементы не содержат информацию об анализируемом объекте в целом [42]. Сказанное позволяет сделать вывод, что в практических задачах оба подхода дополняют друг друга.

Если информация о структуре объекта не нужна и задача состоит только в классификации, а не в описании анализируемого объекта, то целесообразнее применять статистический подход (статистические методы классификации). Но если информация о структуре объекта нужна и требуется его описание, то здесь предпочтение следует отдать синтаксическому подходу. При этом надо помнить, что простые непроеизводные элементы не всегда легко выделить, особенно из зашумленных изображений.

Многие конкретные задачи распознавания занимают промежуточное положение между указанными двумя случаями. В целом, при анализе изображений (см. гл. 1), в которых последние представлены в виде древовидных и графовых структур, лингвистические методы очень полезны, хотя процедуры классификации (предшествующие процедурам анализа изображений) могут быть выполнены с помощью статистических методов.

Как указывалось в начале книги, системы анализа (понимания) изображений являются очередным этапом для разработки визуальных информационных систем (заметим, что процедуры сегментации и классификации являются, в свою очередь, основой разработки как визуальных информационных систем, так и систем понимания изображений). Поэтому, на наш взгляд, основные тенденции заключаются в разработке все более совершенных визуальных информационных систем, для чего необходимо решить следующие группы научно-технических проблем.

1. Проблема совершенствования алгоритмов определения пространственных соотношений, включая алгоритмы работы с трехмерными объектами. Перспективными здесь являются обобщения представлений с помощью 2D-строк и обобщенной строки G (см. гл. 3) на представления с помощью 3D-строки и трехадресной строки G .

2. Проблема создания интегрированных визуальных баз данных, в котором хранятся как сами исходные изображения (либо их признаки, либо символьные изображения), так и их различные описания, включая структурные и позиционные представления. По-видимому, проблема устранения или хотя бы сокращения избыточности здесь будет одной из главенствующих.

3. Проблема разработки визуальных баз знаний, включая совершенствование алгоритмов сравнения структурных представлений данных, а также разработку логических правил (алгоритмов) получения при этом новых знаний.

4. Проблема рационального выбора (оптимизации) структур представления видеоданных для конкретной предметной области.

Ко второй и третьей проблемам примыкают следующие две.

5. Проблема разработки и совершенствования языков описания видеоданных, языков манипулирования данными и языков запросов пользователя.

6. Проблема создания интеллектуального интерфейса.

Кроме указанных проблем, на наш взгляд существуют следующие проблемы уже традиционных процедур (этапов) переработки видеоданных.

7. Проблема устойчивости (робастности) правил принятия решений к неточным знаниям законов распределения вероятностей извлеченных (измеренных) признаков.

8. Проблема уменьшения признакового пространства, а также проблема создания методики расчета вероятностной эффективности процедур классификации.

9. Проблема создания методик формализованного отбора признаков (и их числа) и непроеизводных элементов.

10. Совершенствование стохастических грамматик (и их математических моделей) и алгоритмов лингвистического распознавания образов при работе с зашумленными изображениями.

11. Проблема создания алгоритмов фильтрации в различных признаковых пространствах.

12. Проблема создания различных рекурсивных линейных и нелинейных локальных фильтров и их математических моделей.

13. Проблема создания алгоритмов переработки сжатых изображений, в частности совершенствование алгоритмов сегментации и извлечения некоторых признаков из сжатых изображений.

14. Проблема создания алгоритмов локальной обработки изображений, ориентированных более или менее на исходную пространственную структуру обрабатываемых изображений.

И, наконец, отметим последнюю (в перечне, но не в жизни) проблему (на которую неоднократно указывалось в [11, 18, 37]) – это проблема разработки общей теории изображений.

Приложение

ПРОСТРАНСТВЕННЫЕ МАТРИЦЫ В ЦИФРОВОЙ ОБРАБОТКЕ ИЗОБРАЖЕНИЙ

Как известно, аппарат обычных (плоских) матриц $A = \| a_{ij} \|$ ($i = 1, \dots, m_1, j = 1, \dots, m_2$) успешно применяется в цифровой обработке изображений [31].

Возможно, увеличение числа индексов у элементов a_{ij} на большее, чем 2, число: при этом приходим к понятию пространственной матрицы [37].

Следуя [37], будем называть пространственной l -мерной матрицей совокупность элементов a_{i_1, i_2, \dots, i_l} ($i_1 = 1, \dots, m_1, i_2 = 1, \dots, m_2, \dots, i_l = 1, \dots, m_l$), расположенные в точках l -мерного пространства, определяемых целочисленными координатами i_1, i_2, \dots, i_l .

В дальнейшем (если это не оговорено специально) будем предполагать, что $m_1 = m_2 = \dots = m_l = m$. Получаемую при этом пространственную матрицу будем называть l -мерной матрицей m -го порядка. При $l = 3$ имеем кубическую матрицу второго ($m = 2$) и третьего ($m = 3$) порядка, для которых единичные матрицы могут быть представлены в виде двух и трех граней соответственно [52]

$$I = \left| \begin{array}{cc|cc} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{array} \right|, \quad I = \left| \begin{array}{ccc|ccc|ccc} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{array} \right|. \quad (\text{П. 1})$$

Традиционно преобразование одного изображения X в другое Y (и обратно) в цифровой обработке осуществляется с помощью двумерных унитарных преобразований (см. п. 6.5).

Однако осуществлять линейные преобразования изображений (не только двумерных) можно и с помощью пространственных матриц, причем для преобразования двумерных изображений достаточно кубической матрицы с помощью соотношения

$$Y = XB. \quad (\text{П. 2})$$

Свойства преобразованного изображения $Y = \parallel y_{ik} \parallel$ целиком определяются свойствами матрицы B ($X = \parallel x_{ij} \parallel$).

Пусть существует ($m_1 = m_2 = m_3 = m$) обратная кубическая матрица B^{-1} такая, что $BB^{-1} = I$ (см. (П. 1)). Тогда $YB^{-1} = XBB^{-1}$ и обратное преобразование

$$X = YB^{-1}. \quad (\text{П. 3})$$

Для нахождения обратной матрицы B^{-1} необходимо знать операцию умножения кубических матриц, т.к. обратная матрица определяется из системы линейных алгебраических уравнений, получаемых из равенства $BB^{-1} = I$.

Не всякое определение произведения дает обратную матрицу. Например, если определить произведение двух кубических матриц B и C как $D = BC$ [52]

$$d_{ijk} = \sum_{v=1}^m b_{ijv} c_{jvk}, \quad (\text{П. 4})$$

то обратной матрицы не существует [52].

Следовательно, необходимо дать другое (другие) определение правила умножения. Например, правило умножения вида

$$d_{ijk} = \sum_{v=1}^m b_{ijv} c_{ivk}, \quad (\text{П. 5})$$

позволяет определить обратную матрицу B^{-1} , а следовательно и существует обратное преобразование вида (П. 3).

Следовательно, вопрос о выборе правила умножения решается каждый раз конкретно (от выбранного правила зависит не только само существование обратной матрицы, но и свойства преобразованного изображения Y).

Пространственные матрицы (в отличие от двумерных унитарных преобразований) позволяют линейным образом преобразовать многомерные изображения. Потребность в четырехмерной матрице возникает, например, при совместной обработке нескольких плоских изображений (это либо смежные кадры при межкадровой обработке, либо изображения одной и той же сцены, полученные в разных спектральных диапазонах, либо цветные изображения и т.д.).

При этом обработка (совместная) может осуществляться, например, по правилу $Y = XB$

$$y_{ijk} = \sum_{v=1}^m x_{ivk} b_{ivjk} . \quad (\text{П. 6})$$

Естественно, правило (П. 6) не единственное. Если из нескольких плоских изображений необходимо скомплексировать одно, то правило (П. 6) необходимо скорректировать еще одним дополнительным усреднением вида

$$y_{ij} = \sum_{k=1}^m \sum_{v=1}^m x_{ivk} b_{ivjk} . \quad (\text{П. 7})$$

Дополнительным усреднением (П. 7) можно прийти к вектору признаков y_i ($i = 1, \dots, m$)

$$y_j = \sum_{i=1}^m \sum_{k=1}^m \sum_{v=1}^m x_{ivk} b_{ivjk} . \quad (\text{П. 8})$$

по которому в дальнейшем можно производить распознавание изображений. В цифровой обработке двумерных плоских изображений различные локальные фильтры (см. гл. 4) представляют в виде набора весовых (иногда и нормирующих) множителей – плоских матриц. Аналогично можно поступать и при обработке многомерных изображений. Например, при обработке трех изображений кубическую матрицу третьего порядка Φ (набор весовых множителей) можно представить в виде набора трех весовых множителей $H1$, $H2$ и $H3$ (см. (П. 1))

$$\Phi = [H1 \mid H2 \mid H3] , \quad (\text{П. 9})$$

где все три набора весовых множителя могут представлять различные фильтры (скажем $H1$ – низкочастотный фильтр, $H2$ – оператор Лапласа, а $H3$ – медианный фильтр).

Введя четырехмерную матрицу (см. П. 6), можно из трех, обработанных оператором (П. 9) изображений, скомплексировать одно, обладающее требуемыми (или наилучшими из возможных) свойствами. Фильтр (П. 9) позволяет осуществлять и корреляционную обработку систем из трех изображений для задачи обнаружения объектов, наиболее согласованных с фильтром Φ . Для этого необходимо просуммировать все отклики, даваемые фильтром Φ при его сканировании (свертке) по исходному изображению, и вынести решение об обнаружении по максимальному отклику.

Заметим, что фильтр типа (П. 9) позволяет формировать комбинированные пирамидальные структуры, а также представления с помощью дерева квадрантов многомерных изображений.

Пространственные матрицы являются хорошим аппаратом для представления символьных изображений посредством K -строк ($K \geq 3$), особенно если в последние вносить метрические свойства.

Также, на наш взгляд, пространственные матрицы могут быть полезны при кодировании (по Фримену, Фурману) многомерных контуров и в структурном представлении многомерных изображений.

Список литературы

1. **Александров В.В., Горский Н.Д.** Алгоритмы и программы структурного метода обработки данных, Л.-д.: Наука, 1983, 208 стр.
2. **Александров В.В., Горский Н.Д.** Представление и обработка изображений: рекурсивный подход, Л.-д.: Наука, 1985, 190 стр.
3. **Анисимов Б.В., Курганов В.Д., Злобин В.К.** Распознавание и цифровая обработка изображений, М.: Высшая школа, 1983, 296 стр.
4. **Аркадьев А.Г., Браверман Э.М.** Обучение машин классификации объектов. М.:Наука, 1971, 192 стр.
5. **Бакут П.А., Колмогоров Г.С.** Сегментация изображений: методы выделения границ областей, Зарубежная радиоэлектроника, 1987, N10.
6. **Бакут П.А., Колмогоров Г.С., Варновицкий И.Э.** Сегментация изображений: методы пороговой обработки, Зарубежная радиоэлектроника, 1987, N10.
7. **Бойко В.В., Савинков В.М.** Проектирование баз данных информационных систем, М.: Финансы и статистика, 1989.
8. **Быстрые алгоритмы в обработке изображений**, под ред. Хуанга Т.С., М.: Радио и связь, 1984, 221 стр.
9. **Вентцель Е.С.** Теория вероятностей, М.: Физматгиз, 1969.
10. **Вопросы статистической теории распознавания**, под ред. Варского Б.В., М.: Сов.радио, 1967, 400 стр.
11. **Горелик А.Л., Гуревич И.Б., Скрипкин В.А.** Современное состояние проблемы распознавания, М.: Радио и связь, 1985.
12. **Горелик А.Л., Скрипкин В.А.** Методы распознавания, М.: Высшая школа, 1984, 222 стр.
13. **Джексон Г.** Проектирование реляционных баз данных с использованием микроЭВМ, М.:Мир, 1991.
14. **Дуда Р., Харт П.** Распознавание образов и анализ сцен, М.: Мир, 1976, 511 стр.
15. **Дюран Б., Одед П.** Кластерный анализ, М.: Статистика, 1977, 128 стр.
16. **Загоруйко Н.Г.** Методы распознавания и их применение, М.:Сов.радио, 1972, 206 стр.
17. **Искусственный интеллект**, справочник, Книга 1, Системы общения и экспертные системы, под ред. Попова Э.В., М.: Радио и связь, 1990, 462 стр.
18. **Искусственный интеллект**, Справочник, Книга 2, Модели и методы, под ред. Поспелова Л.А., М.:Радио и связь, 1990, 304 стр.
19. **Кунт М., Икономопулос А., Кошер М.** "Методы кодирования изображений второго поколения"// ТИИЭР, 1985, т.73, № 4.
20. **Куправа Т.В.** Создание и программирование баз данных средствами СУБД dBase4 Plus, Fox Base Plus, Clipper, М.:Мир, 1991, 112 стр.

21. **Латышев В.В.** "Оптимизация усеченного блочного кодирования" в сб. научных трудов МАИ.: Вопросы передачи, распределения и обработки информации в задачах испытаний летательных аппаратов, М.:изд-во МАИ, 1982.
22. **Митчелл О.Р., Делл Э.** "Усеченное блочное кодирование многоуровневой информации"// ТИИЭР, т.68, №7, 1980.
23. **Мусман Х.Г., Пирш П., Граллерт Х.Й.** "Достижения в области кодирования изображений"// ТИИЭР, 1985, т.73, № 4.
24. **Нетравали А.Н., Лимб Дж.О.** "Кодирование изображений: обзор"// ТИИЭР, 1980, т.68, № 3.
25. **Ноултон К.** "Простые эффективные методы кодирования без потерь для передачи многоуровневых и двухуровневых изображений с постепенным воспроизведением"// ТИИЭР, т.68, № 7, 1980.
26. **Опыр Н.В.** "Выделение контуров бинарного изображения" Отбор и обработка информации: Республиканский межведомственный сб. науч. трудов., Физико-механический институт АН УССР, 1988, вып.2.
27. **Павлидис Т.** Алгоритмы машинной графики и обработка изображений, М.:Радио и связь, 1986, 400 стр.
28. **Патрик Э.** Основы теории распознавания образов, М.:Сов. радио, 1980, 408 стр.
29. **Потапов А.А., Галкина Т.В., Орлова Т.К., Хлявич Я.А.** "Метод выделения контуров протяженных детерминированных объектов в стохастических полях"// Радиотехника и электроника АН СССР, 1991, №11.
30. **Препарата Ф., Шеймос М.** Вычислительная геометрия, М.:Мир, 1989.
31. **Претт У.** Цифровая обработка изображений, в двух книгах, М.:Мир , 1982, 790 стр.
32. **Пушкин А.В., Яншин В.В.** "Кореляционные функции выходных сигналов медианных и процентильных фильтров бинарных изображений".- В сб.: Вопросы радиозлектроники, серия "Общие вопросы радиозлектроники", вып.12, 1991.
33. **Распознавание образов: состояние и перспективы**, К.Верхаген, К.Дейн, Ф.Грун и др., М.:Радио и связь, 1985, 103 стр.
34. **Робертс Л.** "Автоматическое восприятие трехмерных объектов". В сб.: Интегральные роботы, М.:Мир , 1973, т.1.
35. **Роджерс Д.** Алгоритмические основы машинной графики, М.:Мир, 1989, 504 стр.
36. **Розенфельд А.** Распознавание и обработка изображений, М.:Мир, 1972, 232 стр.
37. **Соколов Н.П.** Пространственные матрицы и их приложения, М.: Гостехиздат, 1960, 300 стр.
38. **Ту Дж., Гонзалес Р.** Принципы распознавания образов, М.:Мир, 1978.
39. **Фомин Я.А., Тарловский Г.Р.** Статистическая теория распознавания образов, М.:Радио и связь, 1986, 264 стр.
40. **Фор А.** Восприятие и распознавание образов, М.: Машиностроение, 1989, 272 стр.
41. **Фу К.** Последовательные методы в распознавании образов и обучении машин, М.:Наука, 1971, 256 стр.

42. **Фу К.** Структурные методы в распознавании образов, М.:Наука,1977, 319с.
43. **Фукунага К.** Введение в статистическую теорию распознавания образов, М.:Наука, 1979, 368 стр.
44. **Фурман Я.А.** "К вопросу о распознавании изображений с сильной вариабельностью формы". В сб.: Автоматизация анализа и распознавания изображений., Вып.2 - АН Латвийской ССР, Рига, Зинатне, 1980.
45. **Фурман Я.А., Юрьев А.Н., Яншин В.В.** Цифровые методы обработки и распознавания бинарных изображений, Изд-во Красноярского университета, Красноярск, 1992, 246 стр.
46. **Фурман Я.А., Яншин В.В.** Многошаговые процедуры принятия решений, изд-во Красноярского Университета, Красноярск, 1989, 292 стр.
47. **Чен Ш.К.** Принципы проектирования систем визуальной информации, М.:Мир , 1993.
48. **Чукин Ю.В.** Структуры данных для представления изображений, Зарубежная радиоэлектроника, 1983, № 8.
49. **Чукин Ю.В.** Графические информационные системы, Зарубежная радиоэлектроника, 1985, № 10.
50. **Яншин В.В.** "Алгоритмы селекции по площади бинарных изображений и их математические модели"// Радиотехника и электроника АН СССР, т.36, 1991, №11.
51. **Яншин В.В.** "Выделение подозрительных областей сцены по признаку наличия сгущений " // Автоматика, 1990, № 5.
52. **Яншин В.В.** "Многосвязные цепи Маркова как пространственные матрицы"// Доклады АН СССР, т.318, № 5,1991.
53. **Яншин В.В.** "Нелинейная фильтрация комплекснозначных кодов контуров выделяемых объектов"// Исследование Земли из космоса, РАН, 1992, № 4.
54. **Яншин В.В.** "Рекурсивная нелинейная фильтрация бинарных изображений"// Исследование Земли из космоса, РАН, 1993, № 5.
55. **Яншин В.В., Мальгин Ю.Ю.** "Контурное кодирование бинарных изображений в процессе формирования сцены". В сб.: Вопросы радиоэлектроники, серия "Общие вопросы радиоэлектроники", вып.9, 1991.
56. **Ярославский Л.П.** Введение в цифровую обработку изображений, М.:Сов.радио, 1969, 312 стр.
57. **Image analysis: Principles and practice / Publ. J.LoebI. Short Run press** London, 1985, p. 250.
58. **Furman Ya.A., Yanshin V.V.** "Spectral Analysis and Filtering of Closed Polygonal Contours of Images"//Pattern Recognition and Image Analysis,Vol.2, № 3,pp.306-318.
59. **Yanshin V.V.** "Mathematical Models for Nonlinear Mediann and Percentil Filtering of Binary Images, Vol.2, № 4,pp.417-422.
60. **Tortora G., Costagliola G., Arndt T., Chang S.K.** "Pyramidal Algorithms for Iconic Indexing"// Computer Vision, Graphics and Image Processing, 1990 ,v.52, № 1.

ББК 32.81
Я66
УДК 681.3

*Издание выпущено в счет дотации, выделенной
Комитетом РФ по печати*

Яншин В.В.

Я66 Анализ и обработка изображений: принципы и алгоритмы.
– М.: Машиностроение. 1994. – 112с.: ил.

ISBN 5-217-02625-1

В настоящее время вопросами переработки изображений занимается большое количество предприятий. Часть задач решается уже ставшими традиционными методами предварительной переработки изображений. Другая часть задач требует для своего решения нетрадиционной методологии и алгоритмов: это задачи, ориентированные на анализ сцен (понимание изображений). Излагаются основные модели (принципы) и алгоритмы построения систем понимания изображений, под которыми понимаются системы определения пространственных соотношений. Также излагаются традиционные методы переработки изображений, которые рассматриваются как алгоритмы, поддерживающие процедуры понимания изображений. Рассмотрены также различные структуры видеоданных и методы и алгоритмы сжатия изображений.

Для студентов, аспирантов, инженеров и научных работников как специализирующихся в области обработки и анализа изображений, так и использующих в своей деятельности методы и алгоритмы обработки и анализа изображений.

Я $\frac{1402070000 - 033}{038(01) - 94}$ Без объявл.

ББК 32.81

ISBN 5-217-02625-1

© Яншин В.В., 1994.

Учебное издание

Яншин Валерий Владимирович

**АНАЛИЗ И ОБРАБОТКА ИЗОБРАЖЕНИЙ:
ПРИНЦИПЫ И АЛГОРИТМЫ**

Редактор *Г.Н.Сидорова*

Технический редактор *И.В.Мегведев*

Художественный редактор *В.В.Лебедев*

ИБ № 7614

Лицензия ЛР № 080003 от 15.08.91

Подписано в печать с оригинал-макета 26.12.94. Формат 60×88 1/16.

Гарнитура "Балтика". Бумага офсетная. Печать офсетная.

Усл. печ. л. 6,86. Усл. кр.-отт. 7,11. Уч.-изд. л. 6,02. Тираж 2 000 экз. Заказ № 13

Ордена Трудового Красного Знамени издательство "Машиностроение"
107076, Москва, Стромьинский пер., 4.

Отпечатано в типографии № 9 Комитета Российской Федерации по печати.
109033, Москва, Волочаевская ул., 40 с оригинал-макета, изготовленного на персональных ЭВМ.

