

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ  
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ  
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ  
УНИВЕРСИТЕТ»

# **СТРУКТУРНОЕ РАСПОЗНАВАНИЕ ОБРАЗОВ**

Учебно-методическое пособие для вузов

Составитель  
Н.М. Новикова

Издательско-полиграфический центр  
Воронежского государственного университета  
2008

Утверждено научно-методическим советом факультета прикладной математики, информатики и механики 24 апреля 2008 г., протокол № 8

Пособие подготовлено на кафедре технической кибернетики и автоматического регулирования факультета прикладной математики, информатики и механики Воронежского государственного университета.

Рекомендуется для студентов 4 курса дневного отделения.

Для направления: 010200 – Математика. Прикладная математика  
(бакалавриат). ДС. 06

## **Введение**

Способность «распознавать» считается основным свойством человеческих существ, как, впрочем, и других живых организмов. Образ представляет собой описание объекта. В каждое мгновение нашего бодрствования мы совершаем акты распознавания. Мы опознаем окружающие нас объекты и в соответствии с этим перемещаемся и совершаем определенные действия. Мы можем заметить в толпе друга и понять, что он говорит, можем узнать голос знакомого, прочесть рукопись и идентифицировать отпечатки пальцев, можем отличить улыбку от злобной гримасы. Человеческое существо представляет собой очень сложную информационную систему – в определенной степени это определяется чрезвычайно развитыми у человека способностями распознавать образы.

В соответствии с характером распознаваемых образов акты распознавания можно разделить на два основных типа: распознавание конкретных объектов и распознавание абстрактных объектов. Мы распознаем символы, рисунки, музыку и предметы, нас окружающие. Процесс, включающий распознавание зрительных и слуховых образов, можно определить как «сенсорное» распознавание. Процессы этого типа обеспечивают идентификацию и классификацию пространственных и временных образов. Примерами пространственных образов служат символы, отпечатки пальцев, синоптические карты, физические объекты и рисунки. Примерами временных образов являются: речь, сигналы, электрокардиограммы, характеристики цели в радиолокации и временные ряды. С другой стороны, мы в состоянии с закрытыми ушами и глазами опознать старый довод или найти решение задачи. Подобные процессы обеспечивают распознавание абстрактных объектов и их можно определить как «понятийное» распознавание в отличие от зрительного или слухового распознавания.

### **Основные понятия теории распознавания образов**

Определение 1. Под классом образов понимается некоторая категория, определяемая рядом свойств, общих для всех ее элементов.

Определение 2. Образ – это описание любого элемента как представителя соответствующего класса образов.

В случае, когда множество образов разделяется на непересекающиеся классы, желательно использовать для отнесения этих образов к соответствующим классам какое-либо автоматическое устройство. Считывание и обработка погашенных банковских чеков являются примером задачи распознавания образов. Подобные задачи могут выполняться и

людьми; машина, однако, справляется с ними много быстрее. С другой стороны, некоторые задачи распознавания таковы, что человек едва ли в состоянии решать их. Примером задач такого рода служит выделение из множества морских сигналов и шумов тона подводной лодки посредством анализа подводных звуковых сигналов.

Очевидное, но совсем уж «бесхитростное» решение задачи распознавания заключается в применении к отдельным предъявленным образам ряда простых тестов для выделения признаков каждого класса. Совокупность этих тестов должна различать все допустимые образы из разных классов. Например, рассмотрим пять английских букв: COINS. Эти буквы можно классифицировать, применив тесты на наличие таких признаков, как замкнутая кривая, изгиб, двойной изгиб, вертикальный отрезок, короткий отрезок.

Если следовать такому интуитивному подходу, то построение автоматической системы распознавания образов может показаться довольно простой задачей. Не существует, однако, общей теории, позволяющей определить, какие из всего множества мыслимых тестов следует применить к предъявленным образам. Очень ограниченное количество или небрежный выбор тестов не дадут возможности получить характеристики предъявленных для распознавания образов, достаточные для отнесения их к соответствующим классам. Слишком много тестов, с другой стороны, необоснованно усложняют вычисления, осуществляемые в процессе дальнейшего анализа. Отсутствует какое-либо общее правило для получения неких ориентиров, способствующих определению набора таких тестов. Подобный подход чрезмерно зависит от опыта и технической интуиции разработчика и поэтому часто не дает удовлетворительного решения задач распознавания образов, встречающихся в практической деятельности.

### **Основные задачи, возникающие при разработке систем распознавания образов**

Задачи, возникающие при построении автоматической системы распознавания образов, можно обычно отнести к нескольким основным областям.

**Первая задача** связана с представлением исходных данных, полученных как результаты измерений для подлежащего распознаванию объекта. Это – проблема чувствительности. Каждая измеренная величина является некоторой характеристикой образа или объекта. Допустим, например, что образами являются буквенно-цифровые символы. В таком случае в датчике может быть успешно использована измерительная сетчатка. Если сетчатка состоит из  $n$  элементов, то результаты измерений можно представить в виде **вектора измерений** или **вектора образа**

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_n \end{pmatrix},$$

где каждый элемент  $x_i$  принимает, например, значение 1, если через  $i$ -ю ячейку сетчатки проходит изображение символа, и 0 в противном случае.

Векторы образов содержат всю информацию об образах, поддающуюся измерению. Процесс измерения, которому подвергаются объекты определенного класса образов, можно рассматривать как процесс кодирования, заключающийся в присвоении каждой характеристике образа символа из множества элементов алфавита  $\{x_i\}$ . Когда измерения приводят к информации, представленной действительными числами, часто оказывается полезным рассматривать векторы образов в качестве точек  $n$ -мерного пространства измерений. Соответствующий простой пример приведен на рис. В.1. для случая двух классов, обозначенных  $\omega_1$  и  $\omega_2$ . В этом примере предполагается, что классы  $\omega_1$  и  $\omega_2$  представляют соответственно группы футболистов-профессионалов и жокеев. Каждый "образ" характеризуется результатом двух измерений: ростом и весом. Векторы образов имеют, следовательно, вид  $x = (x_1, x_2)'$ , где параметр  $x_1$  – рост, а параметр  $x_2$  – вес. Каждый вектор образа можно считать точкой двумерного пространства. Как следует из рис. В.1, эти два класса образуют непересекающиеся множества, что объясняется характером измерявшихся параметров.

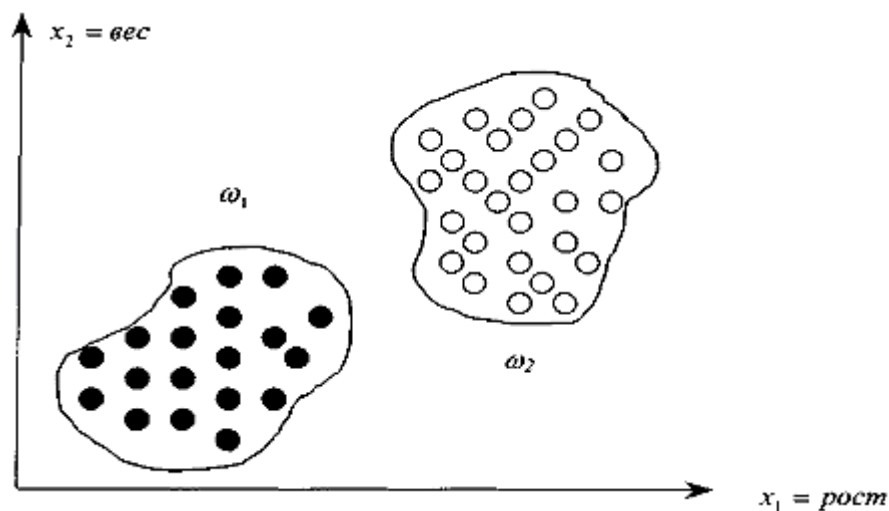


Рис. В.1 Два класса, образующие непересекающиеся множества

В практических ситуациях, однако, далеко не всегда удастся выбрать измеряемые параметры так, чтобы получить строго непересекаю-

щиеся множества. В частности, если в качестве критериев разбиения выбран рост и вес, может наблюдаться существенное пересечение классов, представляющих профессиональных футболистов и баскетболистов.

**Вторая задача** распознавания образов связана с выделением характерных признаков или свойств из полученных исходных данных и снижением размерности векторов образов. Эту задачу часто определяют как задачу предварительной обработки и выбора признаков.

Признаки классов образов представляют собой характерные свойства, общие для всех образов данного класса. Признаки, характеризующие различия между отдельными классами, можно интерпретировать как межклассовые признаки. Внутрикласовые признаки, общие для всех рассматриваемых классов, не несут полезной информации с точки зрения распознавания и могут не приниматься во внимание. Выбор признаков считается одной из важных задач, связанных с построением распознающих систем. Если результаты измерений позволяют получить полный набор различительных признаков для всех классов, собственно распознавание и классификация образов не вызовут особых затруднений. Автоматическое распознавание тогда сведется к процессу простого сопоставления или процедурам типа просмотра таблиц. В большинстве практических задач распознавания, однако, определение полного набора различительных признаков оказывается делом исключительно трудным, если вообще не невозможным. К счастью, из исходных данных обычно удастся извлечь некоторые из различительных признаков и использовать их для упрощения процесса автоматического распознавания образов. В частности, размерность векторов измерений можно снизить с помощью преобразований, обеспечивающих минимизацию потери информации.

**Третья задача**, связанная с построением систем распознавания образов, состоит в отыскании оптимальных решающих процедур, необходимых при идентификации и классификации. После того, как данные, собранные о подлежащих распознаванию образах, представлены точками и векторами измерений в пространстве образов, предоставим выяснить, какому классу образов эти данные соответствуют.

Одним из решений задачи распознавания образов является структурный или синтаксический метод.

### **Синтаксический подход к распознаванию образов**

Большинство различных математических методов решения задач распознавания образов распадается на две группы, одну из которых можно трактовать с позиций теории решений (дискриминантный подход), а другую – в рамках синтаксического (или структурного) подхода. В первом подходе объекты характеризуются наборами чисел – результатов некоторого множества измерений, характеризующих объекты, называемые признаками. Распознавание образов (отнесение каждого объекта к некоторому

классу) обычно проводят при помощи разбиения пространства признаков на области. Развитие исследований по распознаванию образов за последнее десятилетие было большей частью связано с дискриминантным подходом и его применениями. Структурный подход применяется к задачам распознавания образов, в которых важна информация, описывающая структуру каждого объекта. А от процедуры распознавания требуется, чтобы она давала возможность не только отнести объект к определенному классу (классифицировать его), но и описать те стороны объекта, которые исключают его отнесение к другому классу. Типичным примером таких задач служит распознавание изображений или, говоря шире, анализ сцен. Рассматриваемые в этом классе задач объекты обычно сложны, и число требуемых признаков часто велико. Это делает привлекательней идею описания сложного объекта в виде иерархической структуры более простых подобразов.

**Пример 1.** Изображение F на рис. 1 можно описать иерархическими структурами, показанными на рис. 2.

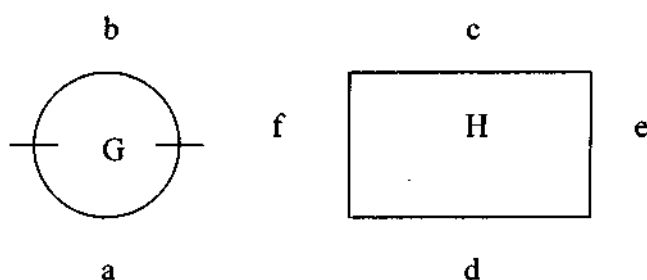


Рис. 1



Рис. 2

Для того чтобы представить иерархическую (древовидную) структурную информацию, содержащуюся в каждом образе, т. е. описывать образ при помощи более простых подобразов, а каждый подобраз снова описывать еще более простыми подобразами и т. д., был предложен синтаксический, или структурный, подход. Этот подход основан на аналогии между структурой образов (иерархической или древовидной) и синтаксисом

языков. В рамках синтаксического подхода считается, что образы строятся из соединенных различными способами подобразов, так же как фразы и предложения строятся путем соединения слов, а слова соединяются из букв. Очевидно, что такой подход полезен только в том случае, когда распознать выбранные простейшие подобразы, называемые непроизводными элементами, легче, чем сами образы.

Определение 3. «Язык», который обеспечивает структурное описание образов в терминах множества непроизводных элементов и операций композиции этих элементов, называют «языком описания образов».

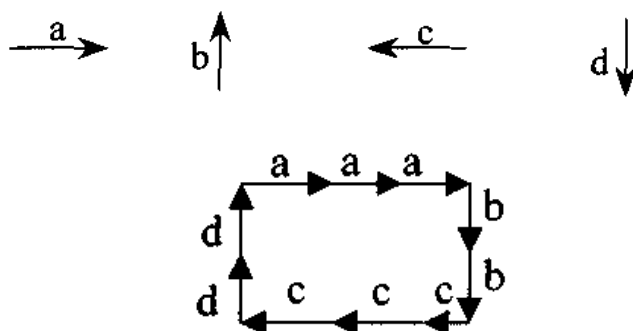
Правила композиции непроизводных элементов обычно задают при помощи грамматики языка описания образов.

Процесс распознавания осуществляется после идентификации в объекте непроизводных элементов и составлении описания объекта. Распознавание состоит в синтаксическом анализе, или грамматическом разборе, «предложения», описывающего данный объект. Эта процедура устанавливает, является ли это предложение синтаксически (или грамматически) правильным по отношению к заданной грамматике. Параллельно синтаксический анализ дает некоторое структурное описание предложения (обычно в виде древовидной структуры).

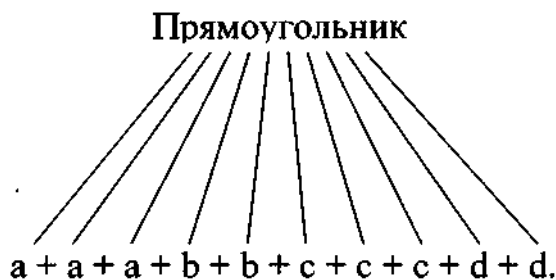
Синтаксический подход к распознаванию образов дает возможность описывать большое множество сложных объектов путем использования небольшого множества непроизводных элементов и грамматических правил. Грамматическое правило (правило подстановки) может быть применено любое число раз, так что оказывается возможным очень компактно выразить некоторые основные, структурные характеристики бесконечного множества предложений. Одним из наиболее привлекательных аспектов этой возможности является использование рекурсивной природы грамматик. Практическая полезность такого подхода зависит, конечно, от способности распознавать непроизводные элементы образов и их взаимные отношения, выраженные операциями композиции.

Различные отношения, определенные между подобразами, или операции композиции, обычно могут быть выражены логическими и (или) математическими операциями. Если, например, в качестве единственного отношения (операции композиции) для описания образов выбрать «конкатенацию» (конкатенацией элементов  $a$  и  $b$  называется составленная из них цепочка  $ab$ ), то при непроизводных элементах





прямоугольник будет представлен цепочкой  $aaabbccdd$ . Точнее, если использовать знак плюс для обозначения операции «конкатенация начала одного элемента к концу другого», то прямоугольник будет представлен предложением  $a + a + a + b + b + c + c + c + d + d$ . Древоподобная структура этого предложения:



Другим представлением структурной информации образа служит «граф отношений». Путем использования графа отношений для описания образов можно расширить класс допустимых отношений, включив в него любое отношение, которое удобно определяется из образа. Заметим, что, во-первых, конкатенация – единственная естественная операция для одномерных языков и, во-вторых, что граф, вообще говоря, содержит циклы, тогда как дерево их не содержит. Поэтому при помощи графа можно выражать более богатые описания, чем с помощью древоподобных структур.

**Пример 2.** На рис. 3 показан граф отношений изображения F (рис. 1).

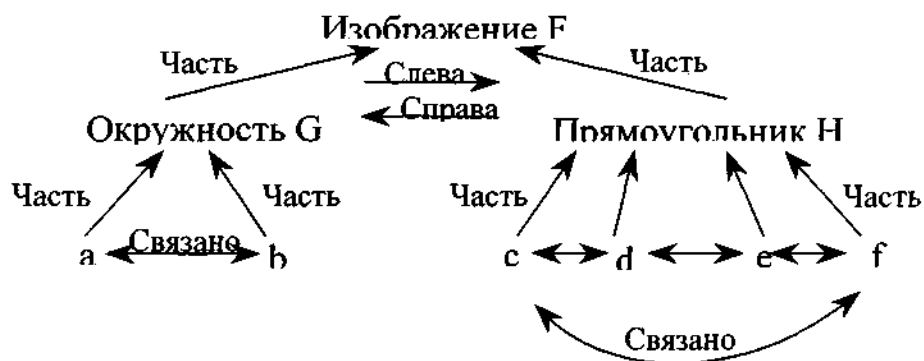


Рис. 3

Использование древовидных структур позволяет непосредственно приспособить методы теории формальных языков к задаче компактного представления и анализа образов, имеющих существенное структурное содержание. Синтаксический подход называют лингвистическим потому, что в нем используются методы теории формальных языков.

### Система синтаксического распознавания образов

Систему синтаксического распознавания образов можно считать состоящей из трех основных частей, а именно: из блока предобработки, блока описания или представления объекта и блока синтаксического анализа. Блок предобработки осуществляет, во-первых, функции кодирования и аппроксимации и, во-вторых, функции фильтрации, восстановления и улучшения объекта. Исходный объект сначала кодируется или аппроксимируется так, чтобы это было удобно для дальнейшей обработки. Черно-белое изображение, например, можно кодировать при помощи решетки (или матрицы), состоящей из нулей и единиц, а кривую можно аппроксимировать отсчетами в дискретные моменты времени или конечным набором коэффициентов Фурье. Затем используют методы фильтрации, восстановления и (или) улучшения для ликвидации шума, восстановления искажений и (или) улучшения качества закодированных (или аппроксимированных) объектов. Обычно предполагают, что на выходе предпроцессора имеются объекты относительно «хорошего качества». Каждый подвергнутый предобработке объект затем представляют в виде структуры языкового типа (например, цепочки). Этот процесс представления объекта состоит, во-первых, из сегментации и, во-вторых, из выделения производных элементов (признаков). Чтобы представить объект через его подобразы, нужно этот объект сегментировать и в то же время идентифицировать (или выделить) в нем производные элементы. Другими словами, каждый подвергнутый предобработке объект разделяется на части и производные элементы на основе заранее заданных синтаксических операций (или операций композиции).

Каждая выделенная часть в свою очередь идентифицируется относительно заданного множества производных элементов. На этой стадии каждый объект получает свое представление через множество производных элементов и определенные синтаксические операции. Например, если задана операция конкатенации, то каждый объект представляется цепочкой примыкающих производных элементов. Решение о том, является ли представление объекта синтаксически правильным (т. е. принадлежит ли он классу образов, описываемых данным синтаксисом или данной грамматикой), принимается «блоком синтаксического анализа» или «блоком грамматического разбора». По ходу синтаксического анализа или грамматического разбора этот блок обычно может давать полное синтаксическое описание объекта в терминах грамматических единиц или дерева грамматического разбора, если представление объекта синтаксически правильно. В противном случае объект либо исключают из рассмотрения, либо анализируют на основе других заданных грамматик, которые, быть может, описывают другие возможные классы рассматриваемых образов.

Простейшей формой распознавания является, по-видимому, «сравнение с эталоном». Цепочка производных элементов, представляющая исходный объект, сравнивается с цепочками производных элементов, представляющих каждый эталонный образ. Из этих образов выбирается тот, который наиболее «согласован» или «близок» к исходному объекту в соответствии с некоторым заданным критерием. При этом иерархическая, структурная информация игнорируется. С другой стороны, при полном грамматическом разборе цепочки, представляющей исходный объект, исследуется полное структурное описание. Между этими крайностями существует ряд промежуточных подходов. Например, могут быть сконструированы последовательности тестов для проверки наличия определенных подобразов либо определенных комбинаций подобразов (или производных элементов). Результаты тестов, полученные, например, с помощью просмотра таблицы, дерева решений или логических операций, используются для классификации. Каждый тест может быть схемой сравнения с эталоном или грамматическим разбором поддерева, представляющего подобраз. Выбор подхода зависит от задачи. Если для распознавания требуется полное описание объекта, то необходим грамматический разбор. В противном случае можно обойтись без полного грамматического разбора и для увеличения эффективности использовать более простые подходы.

Для того чтобы получить грамматику, описывающую структурную информацию класса образов, необходимо устройство вывода грамматики, позволяющее восстановить ее по заданному множеству объектов в виде, подобном языковому. Функции этого устройства аналогичны процессу «обучения» в дискриминантных системах распознавания образов.

В результате обучения на примерах объектов рассматриваемого класса формируется структурное описание этого класса. Затем полученное в виде грамматики описание используется для описания образов и синтаксического анализа. По-видимому, для нахождения наилучшего множества производных элементов и соответствующего структурного описания класса рассматриваемых образов необходимо обучение более общего вида.

## Методы предобработки

### 1. Кодирование и аппроксимация

Для обработки на компьютере объекты сначала переводят в цифровую форму (задают функциями времени или положения в пространстве), а затем представляют в более компактном виде (сжатие данных) с использованием кодирования или аппроксимации. Рассмотрим в качестве объекта одномерную (например, кривая) или двумерную (например, изображение) функцию. Ниже приведены теоремы, из которых следует, что данная функция может быть точно восстановлена по ее значениям на конечном множестве отсчетных точек.

- **Теорема 1.** Пусть преобразование Фурье функции  $f(t)$  равно нулю вне интервала  $-W \leq \omega \leq W$ . Тогда  $f(t)$  может быть точно восстановлена по ее значениям, взятым на расстоянии, не большем, чем  $(1/2)W$ , друг от друга.
- **Теорема 2.** Пусть преобразование Фурье функции  $f(x,y)$  равно нулю вне ограниченной области  $B$  частотной плоскости. Пусть далее  $(a,b)$  и  $(c,d)$  – такие точки, что для любых двух пар целых чисел  $(m,n)$  и  $(p,q)$  множества, получаемые путем сдвига  $B$  на величины  $(m(a+c), n(b+d))$  и  $(p(a+c), q(b+d))$  соответственно не имеют общих точек. Пусть также  $(s,t)$  и  $(u,v)$  таковы, что  $as + bt = cu + dv = 2\pi$ ,  $au + bv = cs + dt = 0$ . Тогда функция  $f(x,y)$  может быть точно восстановлена по ее значениям в отсчетных точках  $(hs + ku, t + kv)$ , где  $h$  и  $k$  – целые числа.

На практике обычно преобразование Фурье функции вне интервала  $(-W, W)$  близко к нулю. Поэтому в результате восстановления функции по отсчетам будет получена лишь ее аппроксимация.

Другим видом аппроксимации является квантование. В этом случае действительное значение функции аппроксимируется с использованием конечного множества значений, называемых уровнем квантования. Проще использовать равномерно расположенные уровни квантования. Однако иногда неравномерно расположенные уровни квантования предпочтительнее. Предположим, например, что на изображении уровни яркости в определенном диапазоне встречаются чаще, чем в других диапазонах. В этом случае может оказаться удобным использовать уровни квантования, рас-

положенные в этом диапазоне плотнее, чем в других. В результате увеличится средняя точность квантования без увеличения числа используемых уровней. Точки отсчета или уровни квантования не обязательно должны быть заданы заранее. Их выбор может зависеть от природы аппроксимируемых объектов. В этом случае, конечно, потребуется более сложная схема дискретизации или квантования. После дискретизации и квантования объекта уровень квантования в каждой точке отсчета кодируют. Пусть  $m$  – число возможных уровней квантования. Тогда для представления любого заданного уровня необходимо по крайней мере  $\log_2 m$  битов информации. Если вероятность появления каждого из возможных уровней известна заранее, то можно построить более эффективный код, снизив среднее число бит на один уровень. Основная идея построения таких кодов состоит в том, чтобы использовать короткие коды для наиболее часто встречающихся уровней квантования.

## 2. Фильтрация, восстановление и улучшение

Для того чтобы «отфильтровать» объект с целью обнаружения в нем заданной конфигурации (сравнение с эталоном) или «восстановить» первоначальный объект, который был испорчен в результате аппроксимации или выполнения других операций подобного рода, можно использовать операции, инвариантные к времени и к изменению положения (т. е. инвариантные к сдвигу). Эти операции используются также для повышения качества посредством «сглаживания» или, наоборот, «повышения контраста».

Пусть  $F$  – множество функций, каждая из которых задает объект, а  $T$  – операция переноса (сдвига) во времени или пространстве, определенная на  $F$ . Операция называется инвариантной по отношению к изменению момента времени (или к изменению положения в пространстве), если она коммутирует с любой операцией  $T$  во времени (или в пространстве). Иными словами,

$$\varphi(T(f)) = T(\varphi(f)) \quad \text{для всех } f \in F.$$

Такие операции обладают тем свойством, что их действие на значение функции в любой данный момент времени, или в любой точке пространства, не зависит от этого момента и от положения этой точки. К этой категории относятся операции сдвига, точечные операции ( $\varphi(f)$  в точке зависят только от значений функции  $f$  в этой точке) и локальные операции ( $\varphi(f)$  в точке зависят только от значений функции  $f$  в некотором интервале или окрестности этой точки). Типичными примерами линейных операций, инвариантных по отношению к изменению момента времени или положения служат одномерное или двумерное преобразование Фурье, передаточная функция. В результате вторичного преобразования Фурье будет получено повернутое исходное изображение.

В двумерном непрерывном случае результат операции  $\phi(f)$  можно представить в виде свертки двух функций  $f$  и  $g$

$$\phi(f(x, y)) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g_{\phi}(x-u, y-v) f(u, v) du dv = f * g_{\phi}.$$

Функция  $g_{\phi}$  называется импульсной функцией рассеивания операции  $\phi$ .

Часто необходимо определить, насколько точно два объекта согласуются друг с другом (насколько они похожи), или обнаружить сходство одного объекта с частью другого. Подсчет взаимной корреляции между двумя функциями  $f$  и  $g$  служит одним из простых методов установления идентичности двух объектов с точностью до переноса и умножения их на константу. Этот метод может быть очень полезным в тех задачах распознавания образов или их производных элементов, в которых легко задать протипы или эталоны этих образов. При этом задача распознавания сводится к подсчету взаимной корреляции исходного объекта с каждым из эталонов.

Другой способ сравнения объекта с эталоном состоит в подсчете не свертки, а произведения фурье-преобразований объектов и в дальнейшем вычислении обратного фурье-преобразования полученного произведения. В этом случае операция сравнения определена не на оси времени или в пространстве, а в области частот или пространственных частот.

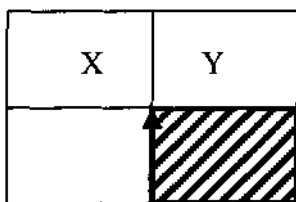
Для подавления присутствующего в объекте шума часто применяют операцию сглаживания. Сглаживание можно осуществить простой заменой значений функции в каждой точке средним значением, взятым по некоторой окрестности этой точки.

### 3. Сегментация

Определение 4. Сегментация – это разбиение изображения на фрагменты. Универсального метода сегментации изображения на фрагменты не существует. В зависимости от характера требуемого описания могут быть полезны разные типы фрагментов изображения. Основным методом ограничения какого-либо фрагмента является применение к данному изображению пороговой операции. Фрагменты можно определить при помощи простого сравнения самого изображения с порогом. Фрагментом может быть совокупность точек, яркость которых больше или, наоборот, меньше порога, или лежит между двумя пороговыми значениями. Например, в задаче распознавания знаков или рисунков, состоящих из линий, сами эти знаки или рисунки представлены совокупностью «черных», а бумага или фон – совокупностью белых точек. При помощи простой пороговой операции можно отделить черную фигуру от белого фона. Пороговую операцию можно комбинировать с другими операциями предварительной обработки, например с сглаживанием, подчеркиванием и с согласованной фильтрацией.

Существенная трудность, возникающая при использовании пороговой операции, состоит в выборе значения порога. Иногда можно найти удачный порог в результате исследования распределения яркостей точек изображения. Например, если известна приблизительная площадь желаемого фрагмента, то можно выбрать нижний (или верхний) порог так, чтобы площадь фрагмента, в точках которого яркость превосходит порог, была меньше (или больше) предсказанной доли площади всего изображения (*метод процентилей*). В других случаях, когда, например, требуемый фрагмент содержит точки другого интервала яркости, чем остальные точки изображения, распределение яркости обычно имеет пик, соответствующий яркости фрагмента. Тогда для выделения фрагмента нужно выбрать пороги, соответствующие локальным минимумам распределения, расположенным по обе стороны от этого пика (*метод моды*). Другой метод определения порога состоит в последовательной проверке точек на их принадлежность к фрагменту. Этот метод позволяет варьировать значение порога в зависимости от характера уже проверенных точек. Разумеется, результат такой процедуры зависит от того, какая последовательность будет выбрана.

Другой подход к задаче выделения фрагментов состоит в прослеживании границ (или контуров). При обработке изображений знаков и вообще рисунков применение метода отслеживания границ позволяет не только выделять фрагменты, но и получать информацию о форме изображений. Простой способ отслеживания границ на двухградационном изображении состоит в систематическом просмотре до тех пор, пока не встретится пара точек или клеток раstra разной яркости. Предположим, что в процессе движения вдоль границы между белыми (скажем, слева) и черными (справа) клетками, достигается «распутье» следующего типа:



Тогда в зависимости от значений яркостей  $x$  и  $y$  «следующая» граничная точка выбирается в соответствии с правилом

X	Y	Следующая граничная точка
0 или 1	0	Повернуть вправо
0	1	Следовать прямо
1	1	Повернуть влево

где 0 – белое, а 1 – черное. Эта процедура позволяет отслеживать связные области черных клеток в направлении часовой стрелки. Ее можно обобщить на случай полутоновых изображений, если задать критерий существования границы. Таким критерием может быть, например, наличие соседних клеток, разница яркостей которых превышает некоторый порог. Отслеживать

границы можно даже тогда, когда в них есть разрывы, например, на изображениях отпечатков пальцев и на изображениях треков частиц в пузырьковых или искровых камерах. Для этого нужно при обнаружении конца границы следовать в прежнем направлении на расстояние, соответствующее максимально допустимому разрыву.

К выделенному фрагменту можно применять новые операции, предназначенные для выделения новых фрагментов, обладающих специфическими свойствами. Например, на выделенном при помощи пороговой операции фрагменте (который может состоять из областей, линий или контуров) можно отследить границы или края и получить изображения этих границ или краев, т. е. новые фрагменты, обладающие свойством связности. Соответствующей маркировкой можно выделить (поименовать) каждый фрагмент, или присвоить ему свой код. Например, это можно сделать, присваивая каждой компоненте фрагмента положительное число так, чтобы клетки одной связной компоненты получили одно и то же число, а клетки разных компонент – разные числа.

Другая операция, также основанная на связности, состоит в сжатии, или утоньшении фрагмента. В том случае, когда фрагмент состоит из большого числа небольших связных компонент, желательно упростить изображение, сжав каждую компоненту до одной клетки. Для замены компонент одиночными клетками можно просто промаркировать их, а затем из каждой компоненты удалять символы до тех пор, пока не будет получено одиночного вхождения каждого символа. Иногда нужно сжать до толщины линии фрагмент, вытянутый в каком-либо направлении. Утоньшение фрагмента можно сделать итеративным применением операции сжатия. При этом на каждом шаге из фрагмента будет удален очередной слой краевых клеток. В этом случае могут понадобиться дополнительные ограничения на операцию сжатия. Иначе во фрагменте могут возникнуть нарушения связности.

Для выделения разного рода фрагментов изображения часто бывает необходимо ввести расстояние на клеточном растре. Пусть  $d$  – функция расстояния, определенная для любой пары клеток растра, т. е. функция положительная, симметричная и удовлетворяющая неравенству треугольника.



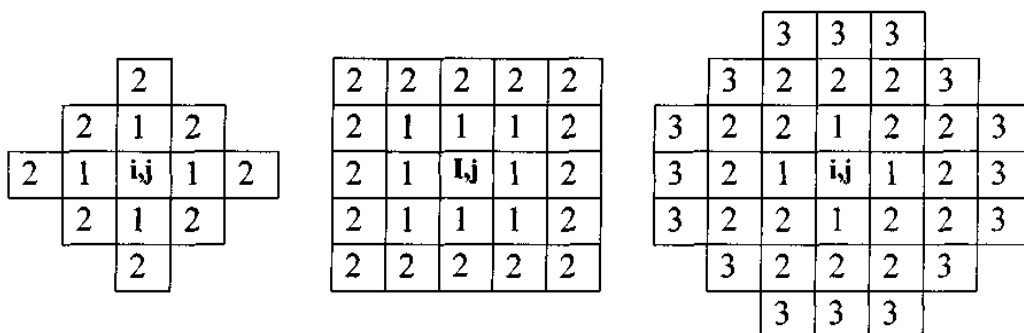


Рис. 5. Расстояние от клетки  $(i, j)$  : ромб,  $r = 2$ ; квадрат,  $r = 2$ ; восьмиугольник,  $R = 2$

Как показано на рис. 5, множества точек  $(l, m)$ , находящихся на расстоянии от заданной точки  $(i, j)$ , не меньшем порога  $r$ , имеют разную форму в зависимости от вида функции расстояния. Множество точек, на которых достигается локальный максимум образуют остров. Нахождение островов – один из подходов к решению задачи утоньшения. Остров также используют для получения информации о форме фрагментов. На изображениях можно выделять фрагменты по заданной форме или заданному направлению. Фрагменты рисунка выделяют при помощи маркировки точек символами, обозначающими приблизительное направление кривой, проходящей через точку. Для сегментации кривой или границы, или края выделяют точки расчленения (позиционные экстремумы, точки перегиба). Для дальнейшего описания выделенного фрагмента изображения вычисляют его параметры. Эти параметры могут быть выражены числами (функции) или логическими высказываниями (предикаты).

## Языки описания образов

### 1. Выбор производных элементов

Первый этап построения синтаксической модели образов состоит в определении множества производных элементов, при помощи которых можно эти образы описать. Это определение существенно зависит от природы образов, от специфики рассматриваемой области применения и от характера доступных технологических средств. Общего решения проблемы выбора производных элементов пока не найдено. Обычно при этом выборе стараются выполнить следующие требования:

1. Производные элементы должны служить основными элементами образов и обеспечивать адекватное и сжатое описание исходных данных в терминах заданных структурных отношений (например, отношение конкатенации).
2. Поскольку структурная информация в производных элементах не важна, и они считаются простыми и компактными образами, их выде-

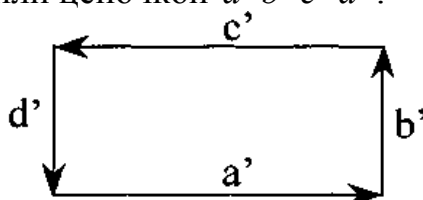
ление и распознавание должны легко осуществляться существующими несинтаксическими методами.

Для речевых образов и для отношения конкатенации «хорошим» множеством непроеизводных элементов считают совокупность фонем. Для описания рукописного текста в качестве непроеизводных элементов было предложено использовать штрихи. Однако вообще для изображений не найдено «универсальных» элементов, аналогичных штрихам в рукописном тексте и фонемам в речи. В ряде случаев для обеспечения адекватного описания образов требуют, чтобы непроеизводные элементы содержали информацию, важную для данного конкретного приложения. Например, если в задаче распознавания важен размер, или форма, или положение, то в непроеизводных элементах должна быть информация о размере (форме или положении), чтобы классы объектов были различимы при любом методе анализа описаний. Разные постановки задачи, даже при одних и тех же исходных данных, обуславливают выбор разных непроеизводных элементов.

**Пример.** Пусть необходимо отличать прямоугольники (разного размера) от не прямоугольников. Тогда выбирают следующие непроеизводные элементы:

- $a'$  – горизонтальный отрезок;
- $b'$  – вертикальный отрезок;
- $c'$  – горизонтальный отрезок;
- $d'$  – вертикальный отрезок.

Множество всех прямоугольников (разного размера) можно представить одним предложением или цепочкой  $a' b' c' d'$ .



Если задача состоит в различении прямоугольников разного размера, то это описание не годится. В этом случае в качестве непроеизводных элементов нужно выбрать отрезки единичной длины:



Тогда множество прямоугольников разного размера можно описать при помощи языка:

$$L = \{a^n b^m c^n d^m / n, m = 1, 2, \dots\}.$$

Требование (2) часто вступает в противоречие с требованием (1), поскольку выбранные в соответствии с (1) непроеизводные элементы не всегда легко распознать при помощи соответствующих методов. С другой стороны, в соответствии с требованием (2) можно выбирать довольно сложные не-

производные элементы, лишь бы они были доступны распознаванию. Чем сложнее производные элементы, тем проще могут быть структурные описания, т. е. они могут укладываться в рамки простых грамматик. В реализации распознающей системы достижение компромисса между указанными требованиями может стать весьма важной задачей.

## 2. Выделение производных элементов на границах

Множество производных элементов, которые обычно используют для описания границ и остовов, получают по схеме цепного кодирования, предложенной Фрименом. На двумерное изображение накладывают прямоугольную сетку, и узлы сетки, которые наиболее близки к точкам изображения, соединяют отрезками прямых. Каждому такому отрезку в соответствии с наклоном присваивают восьмеричное число. Таким образом, изображение представляется цепью (последовательностью) или цепями восьмеричных чисел.

На рис. 6 показано множество начальных элементов и кодовая цепочка, описывающая кривую. Эта схема обладает рядом полезных свойств. Например, поворот изображения на угол, кратный  $45^\circ$ , сводится к прибавлению восьмеричного числа (сложение по модулю 8) к каждому числу цепочки. Конечно, при этом изображение может исказиться. Только поворот на угол, кратный  $90^\circ$ , никогда не приводит к искажениям изображения. Изменяя зернистость сетки, накладываемой на изображение, можно получить любое желаемое разрешение. Этот метод не ограничен изображениями с односвязными замкнутыми границами. Его можно применять для описания произвольных двумерных фигур, составленных из прямых и кривых линий и отрезков.

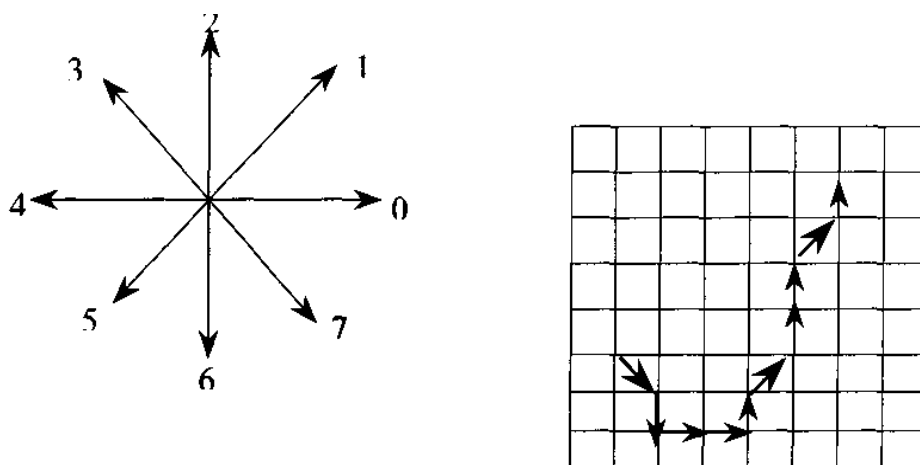


Рис. 6. Цепной код Фримена. Производные элементы и кодовая цепочка кривой 7600212212

## Языки и порождающие грамматики

Допустим, что рассматриваемая задача выделения непроеизводных элементов может быть удовлетворительно решена. Следующий шаг состоит в построении грамматики, порождающей язык для описания исследуемых образов. При построении структурного описания объекта, т. е. при порождении соответствующего языка в рамках выбранной или заданной грамматики, исходным элементом служит начальный символ  $I$ , входящий в множество вспомогательных символов  $W$ . Правило, принадлежащее набору правил  $R$  данной грамматики, позволяет преобразовать начальный символ  $I$  в цепочку символов, например,  $I \rightarrow aA$ , где  $a \in V, A \in W$ , что соответствует замене символа  $I$  символами  $aA$ . Некоторые символы этой цепочки могут входить в множество вспомогательных символов  $W$ , а другие – принадлежать множеству непроеизводных элементов  $V$ . Вспомогательные символы, входящие в преобразованную цепочку, могут снова подвергнуться преобразованиям в соответствии с правилами грамматики, а вспомогательные символы, появляющиеся в цепочках, полученных в результате очередного преобразования, подвергаться преобразованиям и т. д. Предложение (язык) считается сформированным, если представляющая его цепочка содержит только непроеизводные элементы.

Более строго, порождающая грамматика  $G$  есть четверка

$$G = (V, W, I, R),$$

где  $V$  – множество непроеизводных элементов;

$W$  – множество вспомогательных символов;

$I$  – начальный символ ( $I \in W$ );

$R$  – конечное множество правил вывода или правил подстановки.

Язык, порождаемый грамматикой  $G$ , есть

$$L(G) = \{x / x \in V, x - \text{такая цепочка, что } I \Rightarrow x\}.$$

Считается, что порождающая грамматика неоднозначна, если существует цепочка  $x \in L(G)$ , которая может иметь более одного вывода. Ясно, что в языках описания образов, следует избегать неоднозначности, поэтому интересной задачей в этой области является поиск семейства однозначных грамматик. Порождающие грамматики по форме правил подстановки разделены на четыре типа.

**Граматики типа 0** (неограниченные). Они составляют самый широкий класс и характеризуются отсутствием каких-либо ограничений на вид правил подстановки: любые цепочки могут стоять как слева, так и справа от стрелки. Однако для использования этот класс слишком широк. Языками типа 0 называют языки, порождаемые грамматиками типа 0.

**Пример 1.**  $G = (V, W, S, R)$  при  $V = \{a, b, c\}, W = \{S, A, B\}$

$R: S \rightarrow aAbc$

$Ab \rightarrow bA$

$Ac \rightarrow Bbcc$

$bB \rightarrow Bb$

$aB \rightarrow aaA$

$aB \rightarrow S_0$  ( $S_0$  – пустое предложение)

Эта неограниченная грамматика порождает предложение вида  $x = a^n b^{n+2} c^{n+2}$ , где  $n \geq 0$  означает длину цепочки символов. Например, для порождения цепочки  $x = bbcc$  мы применим первые четыре правила и затем последнее, т. е.  $S \Rightarrow aAbc \Rightarrow abAc \Rightarrow abBbcc \Rightarrow aBbbcc \Rightarrow bbcc$ .

Последнее правило допустимо лишь в неограниченных грамматиках. **Грамматика типа 1** (грамматика непосредственно составляющих) характеризуются правилами вида  $\alpha_1 A \alpha_2 \rightarrow \alpha_1 \beta \alpha_2$ , где  $A \in W$ . Эта грамматика допускает замещение вспомогательного символа  $A$  цепочкой  $\beta \in V^+$  только в том случае, если  $A$  появляется в контексте  $\alpha_1 A \alpha_2$ , составленном из цепочек

$\alpha_1$  и  $\alpha_2 \in V^*$

( $V = \{a, b\}$ , то  $V^* = \{S_0, a, b, aa, ab, ba, \dots\}$   $V^+ = \{a, b, aa, ab, ba, \dots\}$ )

**Пример 2**  $G = (V, W, S, R)$  при  $V = \{a, b, c\}, W = \{S, A, B\}$

$R: S \rightarrow abc$

$S \rightarrow aAbc$

$Ab \rightarrow bA$

$Ac \rightarrow Bbcc$

$bB \rightarrow Bb$

$aB \rightarrow aaA$

$aB \rightarrow aa$

порождает предложение вида  $x = a^n b^n c^n$ , где  $n \geq 1$ .

**Грамматика типа 2** (бесконтекстные грамматики) характеризуются правилами подстановки вида  $A \rightarrow \beta$ , где  $A \in W, \beta \in V^+$ . Само название «бесконтекстная» указывает на то, что переменная  $A$  может замещаться цепочкой  $\beta$  независимо от контекста, в котором появляется  $A$ .

**Пример 3.** Бесконтекстная грамматика

$G = (V, W, S, R)$  при  $V = \{a, b\}, W = \{S\}$

$R: S \rightarrow abc$

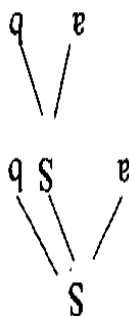
$S \rightarrow aSb$

порождает цепочки вида  $x = a^n b^n$ , где  $n \geq 1$ .

Другим методом описания вывода в бесконтекстной грамматике является использование деревьев вывода. Дерево вывода для бесконтекстной грамматики  $G = (V, W, S, R)$  может быть построено следующим образом.

- 1) Каждая вершина дерева имеет в качестве метки символ из алфавита  $W$  и  $V$ .
- 2) Корень дерева имеет метку  $S$ .
- 3) Если вершина с меткой  $A$  имеет хотя бы одну подчиненную вершину, то  $A \in W$ .
- 4) Если вершины  $n_1, n_2, \dots, n_r$  прямо подчинены вершине  $n$  (с меткой  $A$ ) и помечены (слева направо)  $A_1, \dots, A_k$ , то правило  $A \rightarrow A_1 \dots A_k$  должно  $\in R$ .

**Пример 4.**  $N = 2$



**Грамматика типа 3** (автоматные или регулярные) – это грамматики с правилом подстановок вида  $A \rightarrow aB$  или  $A \rightarrow a$ , где  $A, B \in W$ . Альтернативными допустимыми правилами подстановки являются  $A \rightarrow Ba$  и  $A \rightarrow a$ . Выбор одного из этих двух типов правил исключает, однако, применение правил другого типа.

**Пример 5.** Регулярная грамматика  
 $G = (V, W, S, R)$  при  $V = \{a, b\}, W = \{S\}$

$R: S \rightarrow a$

$S \rightarrow b$

$S \rightarrow b$

$S \rightarrow aS$

$S \rightarrow bS$

порождает цепочки из  $a$  и  $b$ .

Из определений четырех типов грамматики ясно, что каждая автоматная грамматика является бесконтекстной, бесконтекстная грамматика – грамматика непосредственно составляющих, а грамматика непосредственно составляющих принадлежит типу 0.

При описании образов цепочками используется лишь одно отношение – конкатенация. Это означает, что каждый подобраз или непроезвод-

ный элемент можно сочленять с остальными только один раз слева. Для описания двухмерных или трехмерных образов это одномерное отношение недостаточно эффективно. Естественно применить более общий формализм, включающий другие полезные отношения.

Пусть  $R$  – множество  $n$ -арных отношений ( $n > 1$ ). Тот факт, что производные элементы  $x_1, x_2, \dots, x_n$  удовлетворяют отношению  $r \in R$  обозначают через  $r(x_1, x_2, \dots, x_n)$ .

Например, запись ТРЕУГОЛЬНИК ( $a, b, c$ ) означает, что три отрезка линии  $a, b, c$ , удовлетворяют треножному отношению ТРЕУГОЛЬНИК, а запись НАД ( $X, Y$ ) означает, что элемент  $X$  находится над элементом  $Y$ . Рассмотрим пример, иллюстрирующий применение этого формализма к описанию образов.

**Пример 6.** Математическое выражение  $(a+b)/c$  может быть описано предложением : НАД ( НАД ( СЛЕВА ( $a$  , СЛЕВА ( $+$ ,  $b$ )),  $-$  ),  $c$ ),

где запись СЛЕВА ( $X, Y$ ) означает, что элемент  $X$  находится слева от элемента  $Y$ .

Для описания двумерных отношений используется и другая полезная методика, основанная на древовидных структурах.

Дерево – это конечное множество, состоящее из одного или более узлов, таких что

- существует один выделенный узел, называемый корнем дерева;
- остальные узлы разделены на  $m \geq 0$  непересекающихся множеств, каждое из которых, в свою очередь, является деревом.

Эти деревья называются поддеревьями корня. Представление образа в виде дерева называется деревом образа.

Использование деревьев для описания многомерных структур довольно простая процедура. В сущности, любая иерархическая упорядоченная схема ведет к представлению объекта в виде дерева. Например, пусть упорядочение состоит в группировке областей, где область  $b$  находится в области  $a$ , в свою очередь находящейся в области  $r$  (рис. 7). Обозначив корень дерева символом  $r$ , получаем древовидную структуру:

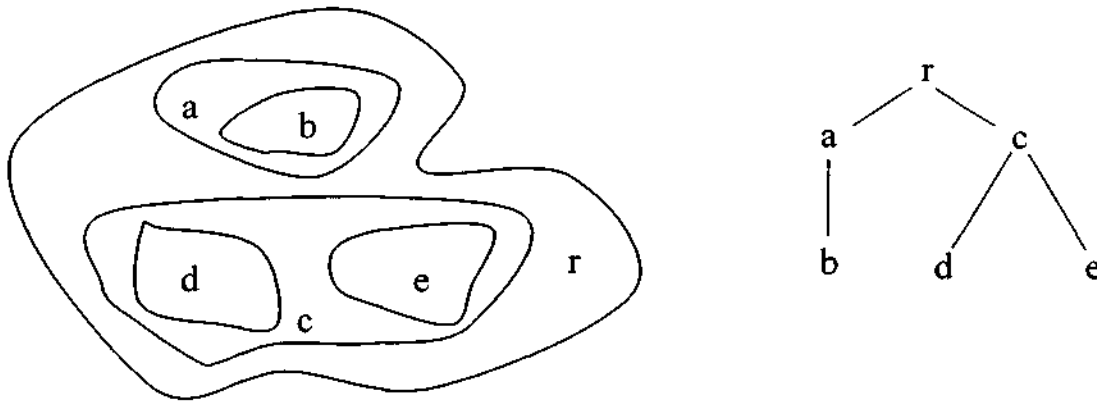


Рис. 7

## Обработка изображений

### 1. Классы изображений

Обработка изображений связана с решением таких задач, в которых и входные, и выходные данные являются изображениями. Говоря об обработке изображений с помощью компьютера целесообразно выделять четыре класса изображений. Эта классификация связана не столько с природой зрительного восприятия изображений, сколько с подходом к их представлению и обработке.

#### Класс 1. Тоновые и цветные изображения

В класс 1 входят изображения, имеющие вид обычных телевизионных изображений. Они обеспечивают довольно точное воспроизведение «реальности» и представляются матрицами с целочисленными элементами (для обозначения последних обычно используют термины *элемент изображения*, *пиксел*, *пэл*). В большинстве прикладных задач эти матрицы имеют очень большие размеры (512\*512), являющиеся наиболее общепринятыми. В связи с этим представлением матрицы не всегда хранятся в памяти в виде обычных матриц – часто используются более изощренные разновидности структур данных. Цветные изображения могут представляться либо при помощи трех матриц (для красного, зеленого и синего цветов), либо с помощью одной матрицы таким образом, что отдельные биты каждого элемента представляют различные цвета. Поскольку человеческий глаз обычно не в состоянии различать уровни освещенности, отличающиеся друг от друга менее чем на 1 %, то для представления цветного изображения достаточно затрачивать по одному байту на цвет на пиксел. Но приемлемых результатов удастся добиться, используя по 3 бита для передачи каждого из двух цветов, и 2 бита для передачи третьего так, что в конечном счете для хранения изображения в памяти затрачивается лишь по одному байту на пиксел. С математической точки зрения иногда удобнее рассматривать цветные изображения как матрицы трехмерных векторов.



## **Класс 2. Двухуровневые или представляемые в нескольких цветах изображения**

Изображение книжной страницы служит типичным примером представителя класса 2 – черно-белого изображения. Подобные изображения можно представлять матрицами, затрачивая по одному биту на элемент, а также в виде «карт», так как на этих изображениях имеются хорошо различимые области одного цвета. Поэтому и объединяются в один класс изображения, представленные в нескольких цветах, и двухуровневые изображения, несмотря на то, что представление с затратой одного бита на элемент матрицы удачно лишь для двухуровневых изображений.

## **Класс 3. Непрерывные кривые и линии**

Примерами изображений класса 3 служат контуры областей, сигналы и диаграммы (графики). Соответствующие данные являются последовательностями точек, допускающих представление через координаты  $(x, y)$ . Однако этот метод представления довольно неэффективен, причем то же самое относится к представлению, основанному на использовании разностей значений координат у соседних точек. Более эффективным является представление с помощью цепных кодов, при использовании которых вектору, соединяющему две соседние точки, ставится в соответствии один символ, принадлежащий некоторому конечному множеству. Представление с помощью цепного кода требует затраты лишь трех битов на точку.

## **Класс 4. Точки или многоугольники**

Изображения класса 4 состоят из множеств отдельных точек, отстоящих друг от друга столь далеко, что для их представления цепным кодом пользоваться нельзя. Вместо него следует применять матрицу, содержащую координаты  $(x, y)$ . Соответствующая аппаратура отображения позволяет соединять точки прямыми или простыми кривыми. Различие между классами 4 и 3, как и между предыдущими классами, довольно неопределенное и приобретает смысл лишь при обращении к способу, используемому для представления изображения. Для представления точек, далеко отстоящих друг от друга, можно применять и цепные коды, требующие затраты более одного кодового символа на точку, причем выбор одной из форм представления должен основываться на характере распределения вероятностей расстояний между точками.

## **2. Ввод изображений**

Изображение, представленное в аналоговой форме, необходимо преобразовать в некоторую числовую матрицу, прежде чем приступить к его обработке на компьютере. Процесс такого преобразования называется дискретизацией и состоит из двух процессов: выборки и квантования. Первый из них заключается в выборе на поле наблюдения некоторого множества точек. В каждой выбранной точке измеряются характеристики изображения, кото-

рые используются на всех последующих этапах обработки изображения. Результаты полученных измерений необходимо описать конечным числом символов: соответствующая процедура называется процессом квантования. Характеризуя плотность размещения выбранных точек, часто говорят о пространственном разрешении, а характеризуя точность представления результатов указанных измерений, о тоновом (или цветовом) разрешении.

### **3. Преобразование изображений**

Преобразование изображения класса 1 в изображение класса 2. Этот процесс называется сегментацией и обеспечивает выделение областей приблизительно одинаковых цвета и/или яркости.

Преобразование изображения класса 2 в изображение класса 3. Одним из допустимых преобразований этого типа является построение (отслеживание) контура, а другим, прореживание. Первое обеспечивает отражение заданной области в некоторую замкнутую кривую, а второе – отображение области в некоторый граф, называемый остовом области.

Преобразование изображения класса 3 в изображение класса 4. Процесс такого преобразования, называемый иногда сегментацией кривых, предназначен для отыскания критических точек контура. В случае многоугольников этими точками являются углы.

Преобразование изображения класса 4 в изображение класса 3. Это преобразование включает процессы интерполирования, обеспечивающего проведение гладкой кривой через некоторое множество точек, и аппроксимации, обеспечивающей проведение гладкой кривой вблизи некоторого множества точек.

Преобразование изображения класса 3 в изображение класса 2. Если в качестве входной информации задается контур, то возникает задача заполнения контура, которая часто рассматривается как задача штриховки. В последнем случае яркость или цвет некоторой области не остаются одинаковыми, а изменяются в соответствии с определенными правилами. Если входной информацией служит остов, то для восстановления области следует использовать процедуру расширения.

Преобразование изображения класса 2 в изображение класса 1. Изображение, воспроизведенное на экране в нескольких цветах, часто оказывается ущербным в эстетическом отношении, так как глаз человека легко обнаруживает контуры.

### **Задания для выполнения лабораторных работ**

#### **Лабораторная работа № 1**

##### **Задание**

Разработать алгоритм и реализовать программу  
«Распознавание простых геометрических фигур» (круг, квадрат, треугольник,

прямоугольник, эллипс) методом сравнения с эталоном.  
Измерить время выполнения алгоритма.

## **Лабораторная работа № 2**

### **Задание 1**

Разработать алгоритм и реализовать программу  
«Распознавание простых геометрических фигур» (круг, квадрат, треугольник, прямоугольник, эллипс) структурным методом.  
Представление фигур осуществить в виде дерева.  
Измерить время выполнения алгоритма.

### **Задание 2**

Разработать алгоритм и реализовать программу  
«Распознавание простых геометрических фигур» (круг, квадрат, треугольник, прямоугольник, эллипс) структурным методом.  
Представление фигур осуществить в виде кода Фримена.  
Измерить время выполнения алгоритма.

### **Задание 3**

1. Сравнить результаты выполнения программы заданий 1 и 2.
2. Сравнить результаты выполнения программы лабораторных работ 1 и 2.
3. Сделать выводы.

## **Лабораторная работа № 3**

### **Задание 1**

Разработать алгоритм и реализовать программу  
«Распознавание сложных изображений (кораблей)» структурным методом.  
Представление изображений осуществить в виде кода Фримена.

### **Задание 2**

Разработать алгоритм и реализовать программу  
«Распознавание сложных изображений (кораблей)» структурным методом.  
Представление изображений осуществить в виде матрицы.

### **Задание 3**

1. Сравнить результаты выполнения программы заданий 1 и 2.
2. Сделать выводы.

По лабораторным работам сделать отчет в электронном и печатном виде.

Отчет по лабораторной работе должен содержать:

1. Постановку задачи.
2. Программную реализацию алгоритма.
3. Сохраненные на переносном носителе информации файлы программной реализации.
4. Вывод о проделанной лабораторной работе.

## Литература

1. Фу К. Структурные методы в распознавании образов / К. Фу ; пер. с англ. Н.В. Завалишина ; под ред. М.А. Айзермана. – М. : Мир, 1977. – 319 с.
2. Журавлёв Ю.И. Распознавание образов и анализ изображений / Ю.И. Журавлёв, И.Б. Гуревич // Искусственный интеллект: Модели и методы. Т. 1. – М. : Радио и связь, 2000. – 310 с.
3. Павлидис Т. Алгоритмы машинной графики и обработки изображений / Т. Павлидис ; пер. с англ. Н.В. Петрова ; под ред. М.С. Гуревича. – М. : Радио и связь, 1986. – 400 с.

## Содержание

Введение.....	3
Основные понятия распознавания образов.....	3
Основные задачи, возникающие при разработке систем распознавания образов.....	4
Синтаксический подход к распознаванию образов.....	6
Система синтаксического распознавания образов.....	10
Методы предобработки.....	12
Языки описания образов.....	17
Языки и порождающие грамматики.....	20
Обработка изображений.....	24
Задания для выполнения лабораторных работ.....	26
Литература.....	28

*Учебное издание*

## СТРУКТУРНОЕ РАСПОЗНАВАНИЕ ОБРАЗОВ

Учебное пособие для вузов

Составитель:  
**Новикова** Нелля Михайловна

Редактор И.Г. Валынкина

Подписано в печать 14.07.08. 60×84/16. Усл. печ. л. 1,6.  
Тираж 25 экз. Заказ 1047.

Издательско-полиграфический центр  
Воронежского государственного университета.  
394000, г. Воронеж, пл. им. Ленина, 10. Тел. 208-298, 598-026 (факс)  
<http://www.ppc.vsu.ru>; e-mail: [pp\\_center@ppc.vsu.ru](mailto:pp_center@ppc.vsu.ru)

Отпечатано в типографии Издательско-полиграфического центра  
Воронежского государственного университета.  
394000, г. Воронеж, ул. Пушкинская, 3. Тел. 204-133.

