

Министерство образования Республики Беларусь  
Учреждение образования  
«Белорусский государственный университет  
информатики и радиоэлектроники»

Кафедра интеллектуальных информационных технологий

**С. А. Самодумкин, М. Д. Степанова, Д. Г. Колб**

## ***ПРАКТИКУМ ПО КОМПЬЮТЕРНОЙ ГРАФИКЕ***

В 3-х частях

Часть 3

Под редакцией  
профессора В. В. Голенкова

*Рекомендовано УМО по образованию  
в области информатики и радиоэлектроники  
в качестве учебно-методического пособия  
для специальности 1-40 03 01 «Искусственный интеллект»*

Минск БГУИР 2014

УДК 004.92(076)  
ББК 32.973.26-018.2я73  
С17

Р е ц е н з е н т ы:

кафедра информационно-вычислительных систем учреждения образования  
«Военная академия Республики Беларусь»  
(протокол №2 от 1 октября 2012г.);

профессор кафедры интеллектуальных систем  
Белорусского государственного университета, кандидат технических наук,  
доцент В. С. Садов

**Самодумкин, С. А.**

С17 Практикум по компьютерной графике. В 3 ч. Ч. 3 : учеб.-метод. пособие / С. А. Самодумкин, М. Д. Степанова, Д. Г. Колб; под ред. проф. В. В. Голенкова. – Минск : БГУИР, 2014. – 78 с. : ил.  
ISBN 978-985-488-911-5 (ч. 3).

Практикум содержит теоретические сведения и задания для выполнения лабораторных занятий по дисциплине «Графический интерфейс интеллектуальных систем». В третью часть практикума включены полигональные алгоритмы компьютерной графики и когнитивная графика.

Часть 1-я издана в БГУИР в 2007 г.

**УДК 004.92 (076.5)**  
**ББК 32.973.26-018.2 я 73**

**ISBN 978-985-488-911-5 (ч. 3)**  
**ISBN 978-985-488-173-7**

© Самодумкин С. А., Степанова М. Д.,  
Колб Д. Г., 2014  
© УО «Белорусский государственный  
университет информатики  
и радиоэлектроники», 2014

## СОДЕРЖАНИЕ

1. ТРИАНГУЛЯЦИЯ. ПОСТРОЕНИЕ ДИАГРАММЫ ВОРОНОГО .....	4
1.1. Общие сведения о триангуляции Делоне .....	4
1.2. Инкрементальный алгоритм триангуляции Делоне .....	6
1.3. Алгоритм построения диаграммы Вороного методом «линии зачистки» .....	10
2. УДАЛЕНИЕ НЕВИДИМЫХ ЛИНИЙ И ПОВЕРХНОСТЕЙ.....	23
2.1. Отсечение.....	23
2.2. Двухмерные преобразования .....	23
2.3. Алгоритм Коэна – Сазерленда.....	28
2.4. Алгоритм разбиения средней точкой.....	28
2.5. Алгоритм отсечения Кируса – Бэка .....	29
2.6. Формальная запись алгоритма Кируса – Бэка.....	33
2.7. Удаление невидимых граней. Алгоритм Робертса.....	34
3. КОГНИТИВНАЯ ГРАФИКА.....	37
3.1. Графические методы анализа данных.....	40
3.1.1. Цель графических методов анализа данных .....	40
3.1.2. Роль визуализации в анализе данных .....	42
3.1.3. Пространственное представление структуры многомерных данных.....	45
3.1.4. График «ящик с усами» .....	47
3.2. Диаграммы рассеивания.....	51
3.2.1. Диаграммы рассеивания в регрессионном анализе.....	57
3.2.2. Диаграммы рассеивания для поиска наилучшего разделения классов в дискриминантном анализе .....	62
3.2.3. Диаграммы рассеивания в кластерном анализе.....	64
3.3. Лица Чернова.....	66
ЛИТЕРАТУРА.....	76

# 1. ТРИАНГУЛЯЦИЯ. ПОСТРОЕНИЕ ДИАГРАММЫ ВОРОНОГО

## 1.1. Общие сведения о триангуляции Делоне

Задача построения триангуляции Делоне является одной из базовых в вычислительной геометрии. К ней сводятся многие другие задачи, она широко используется в машинной графике и геоинформационных системах для моделирования поверхностей и решения пространственных задач.

Триангуляцией называется планарный граф, все внутренние области которого являются треугольниками.

Триангуляция для конечного набора точек  $S$  является задачей триангуляции выпуклой оболочки  $CH(S)$ , охватывающей все точки набора  $S$ . Отрезки прямых линий при триангуляции не могут пересекаться – они могут только встречаться в общих точках, принадлежащих набору  $S$ . Поскольку отрезки прямых линий замыкают треугольники, будем считать их ребрами.

На рис. 1.1 показаны два различных варианта триангуляции для одного и того же набора точек.

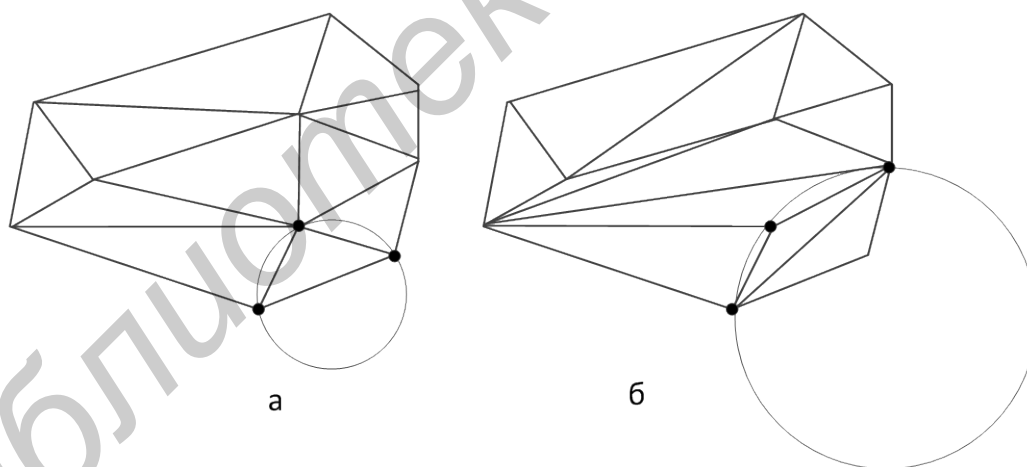


Рис. 1.1. Два различных варианта триангуляции для одного набора точек

Для данного набора точек  $S$  можем видеть, что все точки из набора  $S$  могут быть подразделены на **граничные точки** – те точки, которые лежат на границе выпуклой оболочки  $CH(S)$ , и **внутренние точки** – лежащие внутри выпуклой оболочки  $CH(S)$ . Также можно классифицировать и ребра, полученные в результате триангуляции  $S$ , как **ребра оболочки** и **внутренние ребра**.

К ребрам оболочки относятся ребра, расположенные вдоль границы выпуклой оболочки  $CH(S)$ , а к внутренним ребрам – все остальные ребра, образующие сеть треугольников внутри выпуклой оболочки. Отметим, что каждое ребро оболочки соединяет две соседние граничные точки, тогда как внутренние ребра могут соединять две точки любого типа. В частности, если внутреннее ребро соединяет две граничные точки, то оно является хордой выпуклой оболочки  $CH(S)$ . Заметим также, что каждое ребро триангуляции является границей двух областей: каждое внутреннее ребро находится между двумя треугольниками, а каждое ребро оболочки – между треугольником и бесконечной плоскостью.

Любой набор точек, за исключением некоторых тривиальных случаев, допускает более одного способа триангуляции. Но при этом существует свойство: любой способ триангуляции для данного набора определяет *одинаковое число треугольников*, что следует из теоремы:

**Теорема о триангуляции набора точек.** Предположим, что набор точек  $S$  содержит  $n > 3$  точек и не все из них коллинеарны. Кроме того,  $i$  точек из них являются внутренними (т. е. лежащими внутри выпуклой оболочки  $CH(S)$ ). Тогда при любом способе триангуляции набора  $S$  будет получено точно  $n+i-2$  треугольников.

Набор точек считается **круговым**, если существует некоторая окружность, на которой лежат все точки набора. Такая окружность будет **описанной** для данного набора точек. Описанная окружность для треугольника проходит через все три его вершины. Говорят, что окружность будет **свободной от точек** в отношении к заданному набору точек  $S$ , если внутри окружности нет ни одной точки из набора  $S$ . Однако точки из набора  $S$  могут располагаться на самой свободной от точек окружности.

Триангуляция набора точек  $S$  будет являться **триангуляцией Делоне**, если описанная окружность для каждого треугольника будет свободна от точек.

На схеме триангуляции (рис. 1.1, а) – показана окружность, которая явно не содержит внутри себя других точек (можно провести окружности и для других треугольников, чтобы убедиться, что они также свободны от точек набора).

Это правило не соблюдается на рис. 1.1, б внутрь проведенной окружности попала одна точка другого треугольника, следовательно, эта триангуляция не относится к типу Делоне.

Можно сделать два предположения относительно точек в наборе  $S$ , чтобы упростить алгоритм триангуляции:

- чтобы существовала триангуляция, следует полагать, что набор  $S$  содержит по крайней мере три точки и они неколлинеарны;
- для уникальности триангуляции Делоне необходимо, чтобы никакие четыре точки из набора  $S$  не лежали на одной описанной окружности. Легко видеть, что без такого предположения триангуляция Делоне не будет уникальной, т. к. четыре точки на одной описанной окружности позволяют реализовать две различные триангуляции Делоне.

## 1.2. Инкрементальный алгоритм триангуляции Делоне

Алгоритм работает путем постоянного наращивания текущей триангуляции по одному треугольнику за один шаг. Вначале текущая триангуляция состоит из единственного ребра оболочки, по окончании работы алгоритма текущая триангуляция становится триангуляцией Делоне. На каждой итерации алгоритм ищет новый треугольник, который подключается к *границе* текущей триангуляции.

Определение границы зависит от следующей схемы классификации ребер триангуляции Делоне относительно текущей триангуляции. Каждое ребро может быть спящим, живым или мертвым:

- спящие ребра: ребро триангуляции Делоне является спящим, если оно еще не было обнаружено алгоритмом;
- живые ребра: ребро живое, если оно обнаружено, но известна только одна примыкающая к нему область;
- мертвые ребра: ребро считается мертвым, если оно обнаружено и известны обе примыкающие к нему области.

Вначале живым является единственное ребро, принадлежащее выпуклой оболочке, к нему примыкает неограниченная плоскость, а все остальные ребра спящие. По мере работы алгоритма ребра из спящих становятся живыми, затем мертвыми. Граница на каждом этапе состоит из набора живых ребер.

На каждой итерации выбирается любое одно из ребер  $e$  границы и оно подвергается обработке, заключающейся в поиске неизвестной области, к которой принадлежит ребро  $e$ . Если эта область окажется треугольником  $f$ , определяемым концевыми точками ребра  $e$  и некоторой третьей вершины  $p$ , то ребро  $e$  становится

мертвым, поскольку теперь известны обе примыкающие к нему области. Каждое из двух других ребер треугольника  $f$  переводится в следующее состояние: из спящего в живое или из живого в мертвое.

Здесь вершина  $p$  будет называться *сопряженной с ребром  $e$* . В противном случае, если неизвестная область оказывается бесконечной плоскостью, то ребро  $e$  просто умирает. В этом случае ребро  $e$  не имеет сопряженной вершины.

Для поиска сопряженной вершины предположим, что окружность  $C_r$  является описанной окружностью для известной области ребра  $ab$  (на рис. 1.2 такой известной областью будет треугольник  $abc$ ). Если известная область для ребра  $ab$  является неограниченной, то  $r = -\infty$  и  $C_r$  представляет собой полуплоскость, лежащую слева от  $ab$ . Нужно найти такое наименьшее значение  $t > r$ , чтобы некоторая точка из набора  $S$  (отличная от точек  $a$  и  $b$ ) принадлежала окружности  $C_t$ . Если не существует такого значения  $t$ , то ребро  $ab$  не имеет сопряженной точки. Более образно это соответствует надуванию двумерного пузыря, привязанного к отрезку  $ab$ . Если такой пузырь достигает некоторой точки из набора  $S$ , то эта точка является сопряженной отрезку  $ab$  (точка  $d$  на рис. 1.2). В противном случае, если не встретится ни одной такой точки из  $S$ , то пузырь разрастается до заполнения всей бесконечной полуплоскости справа от отрезка  $ab$  и тогда отрезок  $ab$  не имеет сопряженной точки.

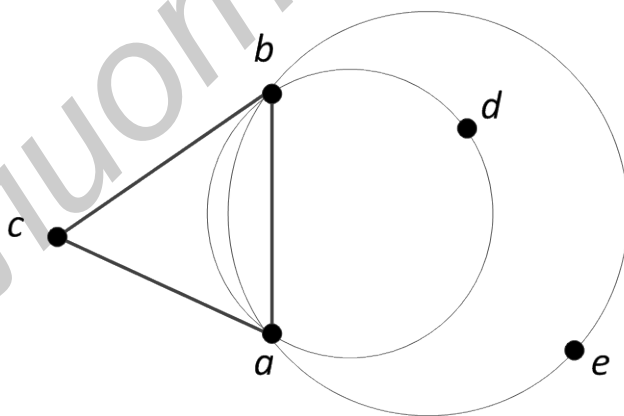


Рис. 1.2. Определение сопряженной точки ( $d$ ) для ребра  $ab$

**Алгоритм** инкрементальный триангуляции Делоне

Входные данные: массив точек  $S$ .

**Шаг 1.** Инициализация: проинициализировать массив треугольников, множество живых ребер.

**Шаг 2.** Найти ребро оболочки  $e$  среди массива точек  $S$  (можно использовать первую итерацию метода Джарвиса); добавить это ребро во множество живых ребер, пока множество живых ребер не пусто.

**Шаг 3.** Извлечь очередное ребро  $e$  из множества живых.

**Шаг 4.** Найти сопряженную точку для ребра  $e$ .

**Шаг 5.** Если сопряженная точка нашлась (точка  $p$ ), произвести обновление множества живых ребер по следующему правилу:

а) построить два новых ребра: одно из начальной точки ребра  $e$  и сопряженной точки  $p$ , другое – из конечной точки ребра  $e$  и точки  $p$ ;

б) произвести поиск по множеству живых ребер на предмет существования этих двух ребер. Если какое-либо ребро найдено, удалить его из множества живых ребер, иначе добавить ребро во множество живых;

в) построить треугольник по трем точкам: начальной и конечной ребра  $e$  и сопряженной точки  $p$ ; добавить треугольник в массив треугольников.

**Конец алгоритма.**

**Пример.** На рис. 1.3 продемонстрировано пошаговое выполнение инкрементального алгоритма триангуляции Делоне. Ребра, входящие в состав границы, выделены толстой линией.

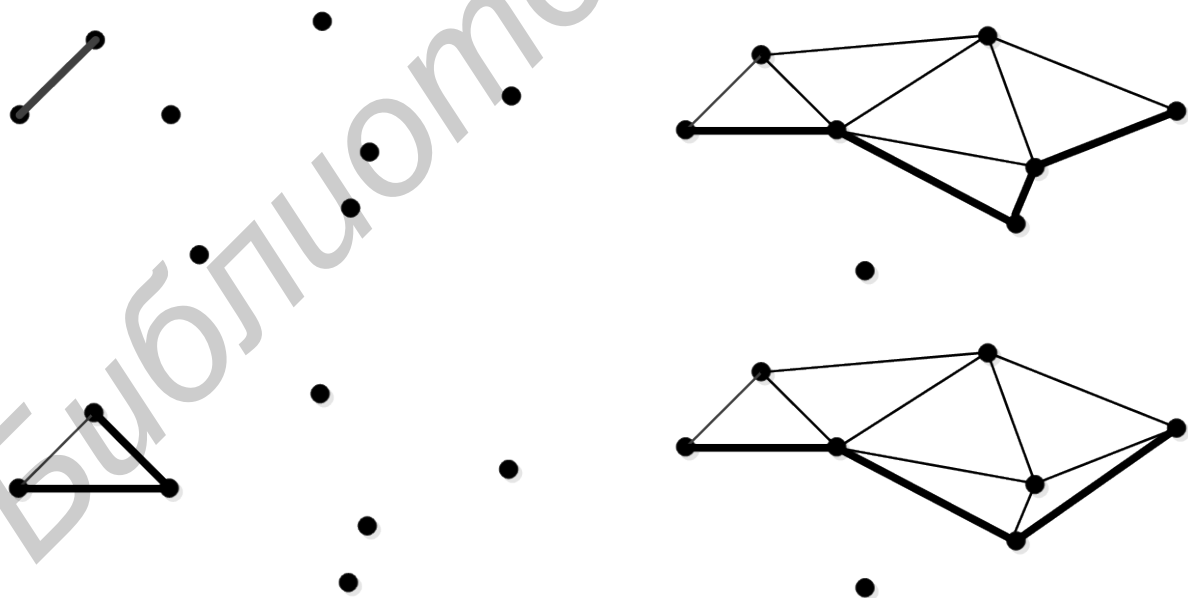


Рис. 1.3. Инкрементальный алгоритм триангуляции Делоне (начало)



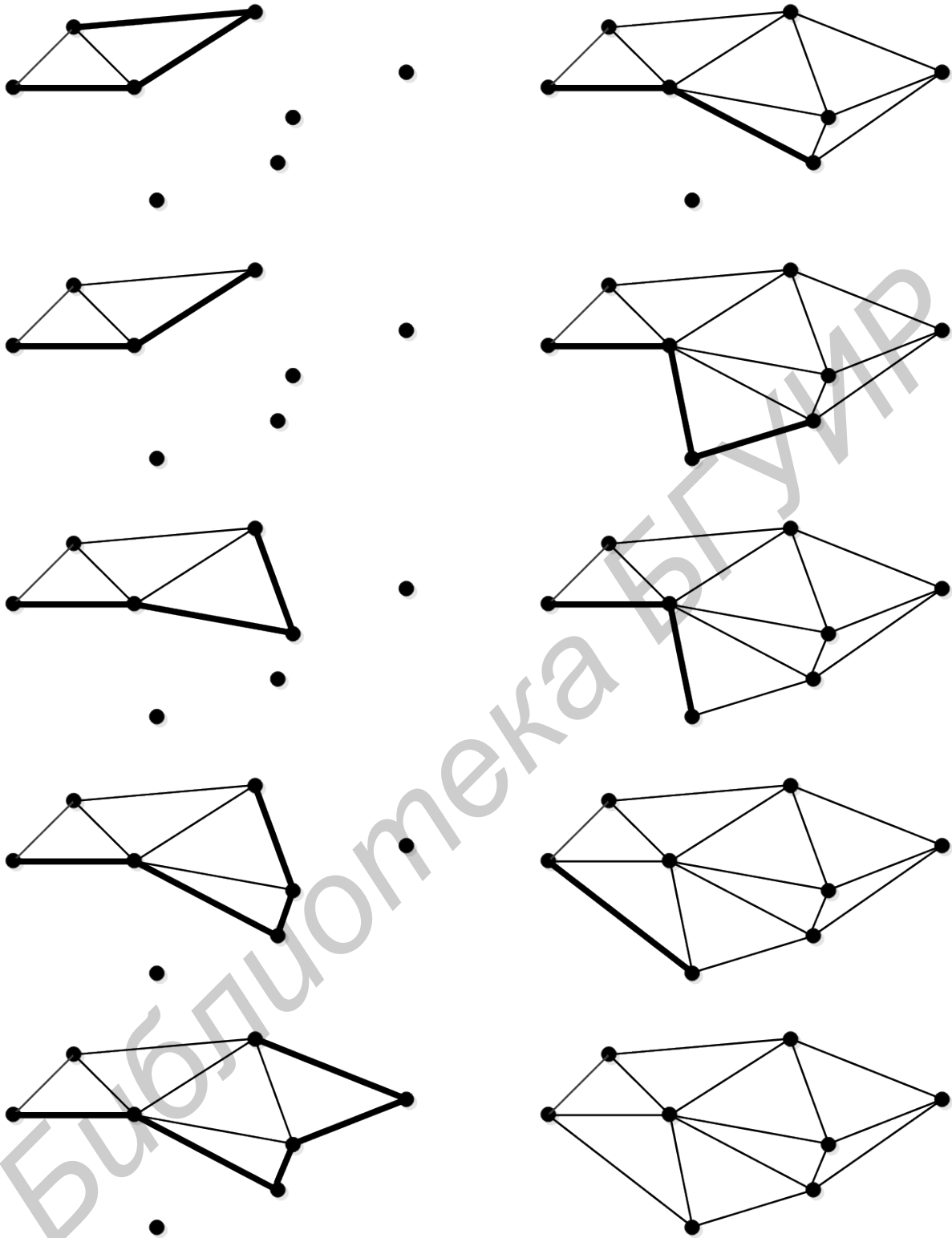


Рис. 1.3. Инкрементальный алгоритм триангуляции Делоне (окончание)

### 1.3. Алгоритм построения диаграммы Вороного методом «линии зачистки»

Диаграмма Вороного как фундаментальный инструмент вычислительной геометрии активно применяется как в теории, так и на практике: например, в распознавании образов, робототехнике, геодезии, компьютерной графике, САПР интегральных схем. Диаграмма Вороного является одной из фундаментальных структур данных вычислительной геометрии.

Диаграмма Вороного – геометрическое разбиение области на многоугольники, обладающие следующим свойством: для любого центра системы  $\{A\}$  можно указать область пространства, все точки которой ближе к данному центру, чем к любому другому центру системы. Такая область называется многогранником Вороного, или областью Вороного (рис. 1.4).

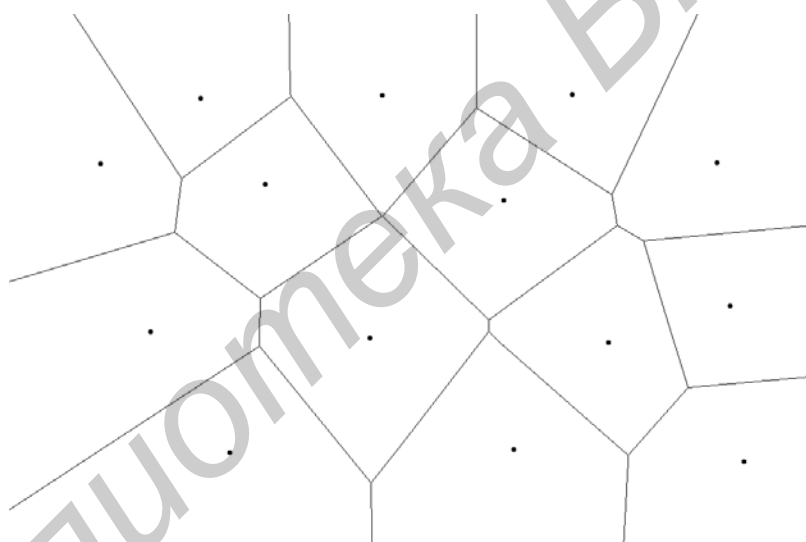


Рис. 1.4. Диаграмма Вороного

Строгое определение многоугольника Вороного  $T_i$  определяется следующим образом:

$$T_i = \{x \in R^2: d(x, x_i) < d(x, x_j) \forall j \neq i\}. \quad (1.1)$$

Вершина ячейки Вороного – центр наибольшей пустой окружности, проходящей через три или более точки из заданного множества (рис.1.5). При этом ребром ячейки Вороного является прямая, проходящая между двумя узлами  $p_i$  и  $p_j$ , такая,

что центр пустой окружности, проходящей через эти точки, принадлежит прямой (см. рис.1.5).

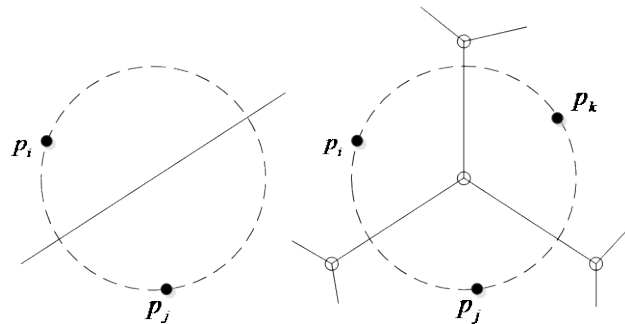


Рис. 1.5. Свойства ячеек диаграммы Вороного

Идея методов «линии зачистки» заключается в перемещении горизонтальной линии сверху вниз по плоскости (рис. 1.6). Во время выполнения «зачистки» формируется информация относительно структуры, описывающей диаграмму Вороного. При этом необходимо *сохранять точки пересечения диаграммы с линией зачистки*. Трудность заключается в том, что в момент, когда линия зачистки пересекает самую верхнюю вершину ячейки Вороного, невозможно определить узел  $p_i$ , вокруг которого строится данная ячейка. Значит, нужно искать вершины ячейки Вороного в модифицированном варианте методов.

Для построения диаграмм Вороного рассматриваются события двух видов – *событие точки и событие круга*. Важнейшей задачей является определение точки области, в которой наступающее событие произойдет. Основная сложность данной задачи заключается в том, что *точка, лежащая впереди линии зачистки, может породить вершину Вороного, которая будет располагаться позади этой линии*. Стивен Фортун сделал следующее предположение: вместо того чтобы вычислять диаграмму Вороного через линию зачистки в ее заключительной форме, лучше строить искаженную, но *топологически эквивалентную* версию диаграммы.

Пусть задано множество точек  $P = \{p_1, p_2, \dots, p_n\}$ . Для преодоления проблемы взаимного влияния точек, расположенных по разные стороны от линии зачистки, будем строить только ту часть диаграммы, которую нельзя «испортить» точками, лежащими ниже линии зачистки. Для этого разделим полуплоскость, лежащую над линией зачистки, на *две области*, одна из которых будет содержать точки, распо-

ложенные над линией зачистки, *ближе к некоторой точке  $p$  из  $P$ , чем к самой линии*, другая – *точки, расположенные ближе к линии зачистки, чем к некоторой точке из заданного множества  $P$* . Множество точек  $q$ , равноудаленных от линии зачистки и самой близкой точки  $p$  из множества  $P$ , называется *береговой линией* (рис.1.6).

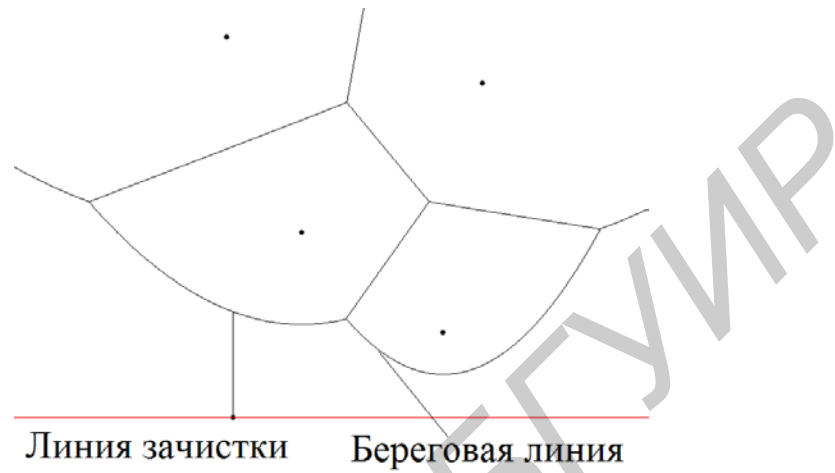


Рис. 1.6. Ключевые элементы построения диаграммы Вороного методом «линии зачистки»

*Из геометрии известно, что множество точек, равноудаленных от заданной точки  $p$  из  $P$ , лежащей выше горизонтальной линии, и самой линии, образует параболу. Можно показать, что парабола сужается по мере приближения точки  $p$  к линии зачистки и постепенно вырождается в луч, выходящий из  $p$ . Уравнение параболы относительно фокуса  $F(x_F, y_f)$  и директрисы  $y = L$  можно записать в следующем виде:*

$$y = \frac{1}{2(y_F - L)}((x - x_F)^2 + y_F^2 - L^2). \quad (1.2)$$

Таким образом, *береговая линия состоит из дуг нижних частей парабол. Следует заметить, что точка пересечения двух парабол (контрольная точка) равноудалена от двух точек из  $P$  (для которых строились параболы) и береговой линии. Следовательно, она должна лежать на ребре ячейки Вороного.*

Таким образом, рассматриваемый алгоритм состоит в *моделировании роста береговой линии во время движения линии зачистки сверху вниз и отслеживании точного пути, пройденного контрольной точкой, перемещающейся вдоль ребра ячейки Вороного.*

До тех пор пока линия зачистки движется вниз, параболы, формирующие береговую линию, изменяются непрерывно. Как и во всех алгоритмах, использующих идею метода линии зачистки, необходимо проводить *дискретное моделирование движения линии зачистки по точкам, в которых возникают события, изменяющие топологическую структуру береговой линии*. Эти события могут быть двух видов:

- *события точки*: линия зачистки по ходу движения переходит через новую точку из множества. В результате будет вставлена новая дуга в береговую линию;
- *события круга*: удаление дуги из береговой линии при уменьшении размера дуги до точки. Событие наступает в тот момент, когда по мере движения линии зачистки вниз сталкиваются две контрольные точки. В результате в точке столкновения будет образована вершина ячейки Вороного.

Реализация рассматриваемого алгоритма включает в себя обработку описанных выше типов событий. Рассмотрим их более подробно.

**Событие точки.** Данное событие возникает всегда, когда линия зачистки пересекает точку из множества  $P$  (рис. 1.7, а). Как упоминалось ранее, в тот момент, когда линия зачистки касается точки  $p$  из  $P$ , соответствующая точке дуга вырождается в луч, исходящий из этой точки и направленный в сторону береговой линии (рис. 1.7, б). При движении линии зачистки вниз данный луч расширится в параболу на береговой линии (рис. 1.7, в). Для обработки данного события определим дугу береговой линии, которая лежит непосредственно выше новой точки. Затем разделим данную дугу береговой линии на две вставкой новой бесконечно малой дуги в точку пересечения береговой линии и вертикального луча, исходящего из  $p$ . При дальнейшем движении линии вниз вставленная дуга будет расширяться до тех пор, пока не соединится с другими ребрами диаграммы.

Контрольные точки движутся по ребрам ячейки. Во время события точки образуются две новые контрольные точки. В начальный момент времени они совпадают, затем начинают двигаться в противоположных направлениях, вычерчивая одно и то же ребро диаграммы (рис. 1.7, г).

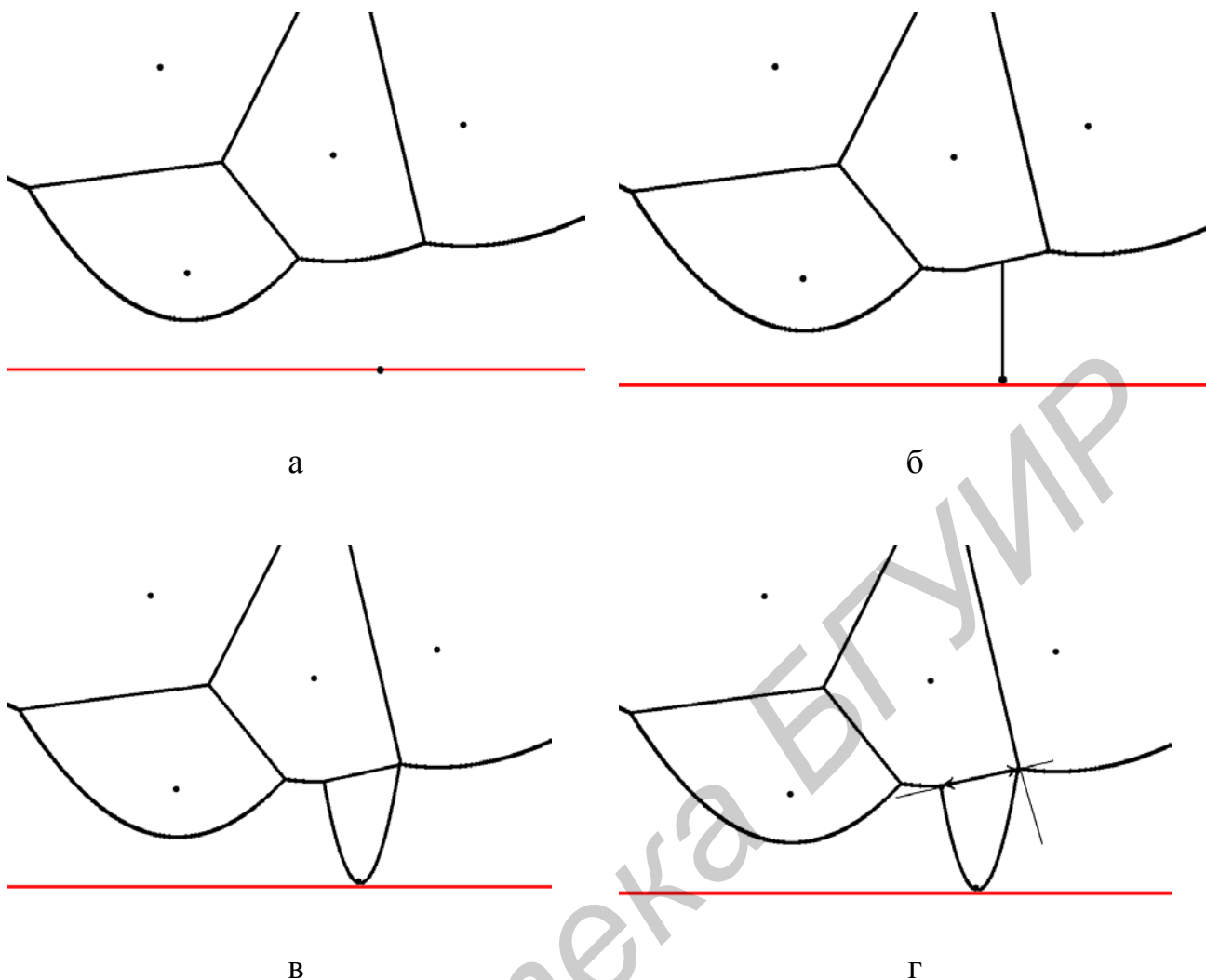


Рис. 1.7. Возникновение события точки

При этом образующееся ребро не соединено с остальной частью диаграммы. По мере движения линии зачистки вниз ребро ячейки увеличивается до тех пор, пока одна из контрольных точек, образующих ребро, не столкнется с какой-нибудь другой контрольной точкой. Новая дуга добавляется в береговую линию только после обработки события точки. Отсюда видно, что береговая линия состоит из  $2n - 1$  параболических дуг, где  $n$  – количество точек, находящихся выше «линии зачистки», для которых еще не построена замкнутая ячейка.

**Событие круга (вершины).** Следующий тип события – удаление дуги из береговой линии при уменьшении размера дуги до точки – событие круга (вершины). В противопоставление событию точки события круга создаются динамически по ходу работы алгоритма. Каждое событие вершины формируется объектами, являющимися соседними на береговой линии. События здесь генерируются тройкой точек (рис. 1.8).

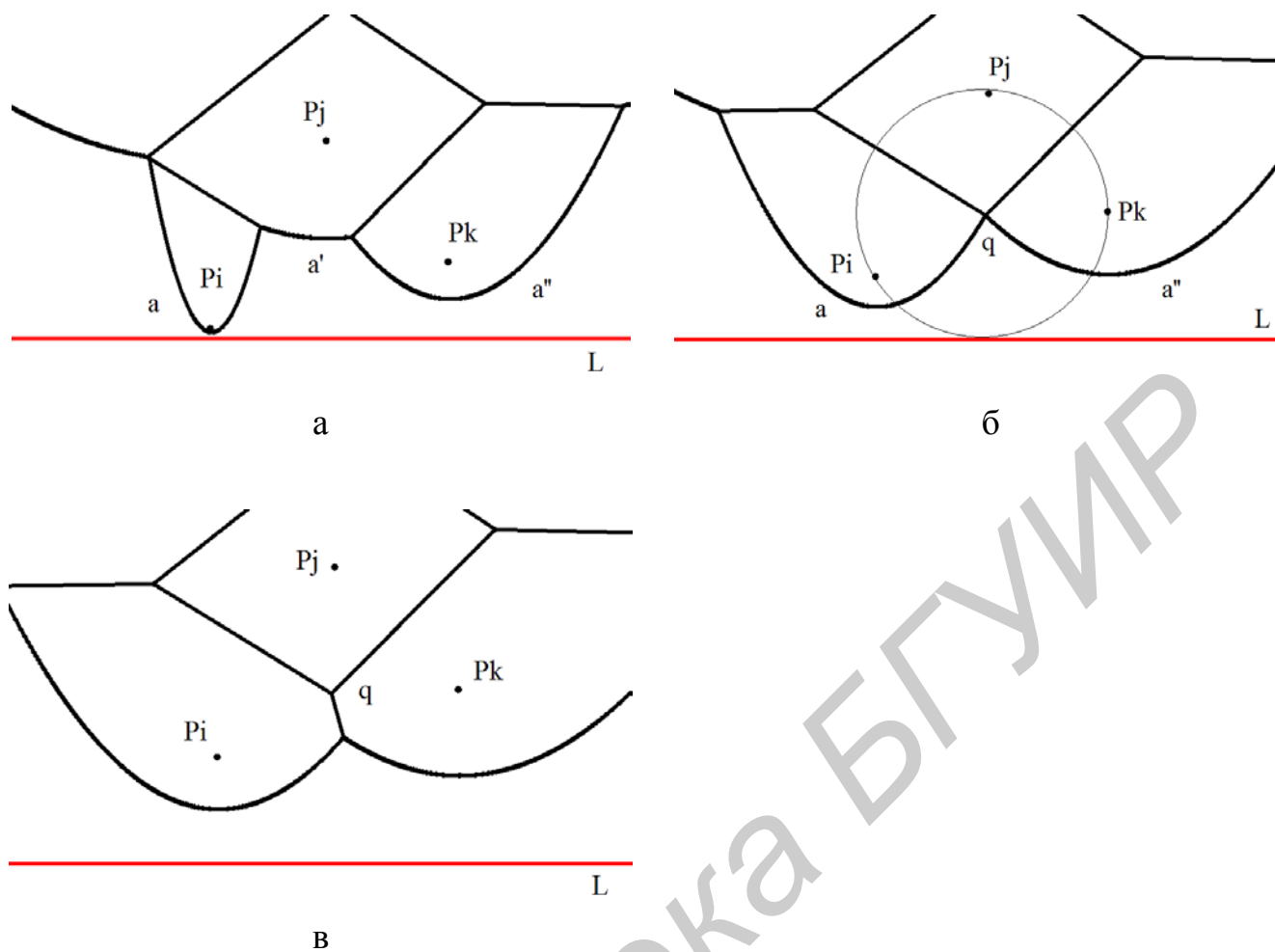


Рис. 1.8. Возникновение события круга

Пусть  $\alpha'$  – дуга, которая в следующий момент времени будет удалена, а  $\alpha$  и  $\alpha''$  – соседние с ней дуги. Дуги  $\alpha$  и  $\alpha''$  не могут быть частями одной параболы. Таким образом, три дуги определяются тремя различными точками. В момент, когда дуга  $\alpha'$  – исчезает, все три параболы проходят через общую точку  $q$ . Точка  $q$  равноудалена от прямой  $L$  и трех заданных точек. Очевидно, что существует окружность с центром в точке  $q$ , проходящая через  $p_i, p_j, p_k$ , причем нижняя точка окружности лежит на прямой  $L$  (касательная к окружности). При этом не существует точек из  $P$ , лежащих внутри окружности: такая точка была бы ближе к  $q$ , чем  $q$  к  $L$ , тем самым противоречая тому факту, что  $q$  находится на береговой линии. Отсюда следует, что  $q$  есть вершина ячейки Вороного. Таким образом, в момент удаления дуги с береговой линии две контрольные точки сталкиваются, соединяя два новых ребра диаграммы. Будем называть событие, при котором линия зачистки достигает самой нижней точки окружности, построенной по трем после-

довательным точкам на береговой линии, *событием круга*. Удаление дуги из береговой линии происходит только при возникновении события круга.

Тогда при возникновении события точки образуется новая дуга, а при событии круга – удаляется существующая. *При событии точки – новое ребро начинает расти, при событии круга – два растущих ребра сталкиваются, образуя вершину диаграммы.*

### **Реализация алгоритма**

При наступлении новых событий составная структура береговой линии меняется следующим образом: при возникновении события точки образуется *новая дуга*, при возникновении события круга дуга *удаляется, образуя вершину ячейки Вороного*. Таким образом, необходима структура данных, способная хранить части диаграммы, вычисленные на текущий момент времени. Как правило, для всех алгоритмов метода используется очередь событий и некоторая структура для хранения состояния береговой линии: *двусвязный список или двоичное дерево поиска*. Для реализации алгоритма применяются следующие конструкции.

*Двусвязный список полуребер*. Строящаяся диаграмма Вороного сохраняется в обычной для нее структуре – двусвязном списке полуребер  $D$ . Но при этом в ней присутствуют ребра и неограниченные ребра (лучи), которые не могут быть помещены в двусвязный список. Во время процесса построения диаграммы это не является проблемой, т.к. представление береговой линии позволяет эффективно обращаться к соответствующим частям двусвязного списка. Для получения правильного двусвязного списка для всей диаграммы Вороного необходимо ограничить область построения большой прямоугольной областью, в которую бы помещались все вершины.

*Двоичное дерево поиска*. Береговая линия представляется сбалансированным двоичным деревом поиска  $T$  (рис.1.9). Его листья соответствуют дугам на береговой линии: крайний левый лист представляет крайнюю левую дугу, следующий лист представляет вторую крайнюю левую дугу и т. д. Каждый лист хранит точку из  $P$ , представляющую соответствующую дугу. Внутренние узлы дерева  $T$  соответствуют контрольным точкам на береговой линии и представляются в виде упорядоченной пары  $\langle p_i, p_j \rangle$ , где  $p_i$  обозначает параболу, располагающуюся слева от точки, а  $p_j$  – справа. Используя такое представление береговой линии, можно найти дугу, располагающуюся непосредственно над новой точкой. Во внутреннем



узле сравниваем значение абсциссы новой точки со значением абсциссы контрольной точки, которая может быть вычислена из пары точек и позиции линии зачистки.

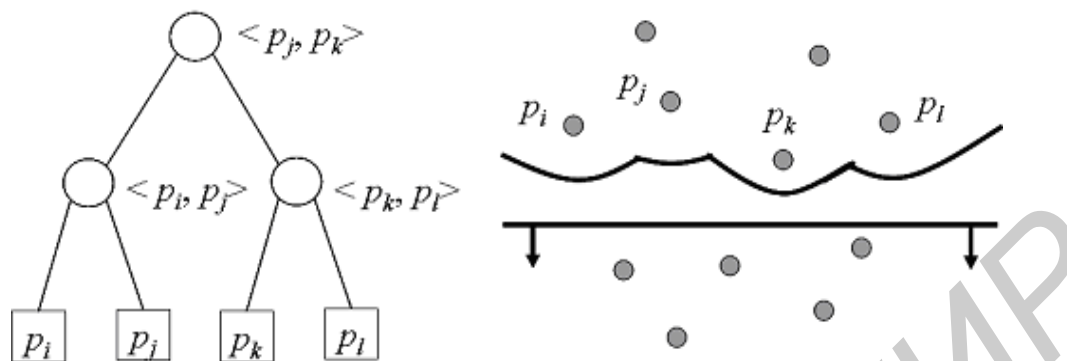


Рис. 1.9. Представление береговой линии двоичным деревом поиска

В двоичном дереве  $T$  дополнительно храним указатели на две структуры, используемые во время зачистки. Каждый лист  $T$ , представляющий дугу  $\alpha$ , хранит один указатель на узел в очереди событий, а именно, на узел, соответствующий событию круга, при возникновении которого дуга  $\alpha$  будет удалена. Наконец, каждый внутренний узел содержит указатель на полурешбро, которое прослеживается контрольной точкой.

*Очередь событий.* Очередь событий  $Q$  представляется очередью с приоритетом, в которой элементы очереди сортируют по убыванию значения координаты  $y$ . Очередь хранит два типа событий: события точки, которые известны заранее до начала работы алгоритма, и события круга, которые вычисляются по мере движения линии вниз. Для событий точки сохраняются координаты самих точек. Для событий круга сохраняются координаты самой нижней точки окружности, а также указатель на лист  $T$ , представляющий дугу, которая должна исчезнуть.

**Алгоритм** построения диаграммы Вороного методом «линии зачистки».

**Шаг 1.** Инициализация очереди событий  $Q$  значениями координат точек из множества  $Q$ .

Пока очередь не пуста.

**Шаг 2.** Получить событие из очереди с максимальным значением  $y$ .

**Шаг 3.** Если событие  $p_i$  является событием точки, то вызвать подпрограмму обработки события точки, иначе вызвать подпрограмму обработки события круга.

**Шаг 4.** Удаление событие из очереди.

**Конец алгоритма.**

**Пример.** Для примера рассмотрим случай, представленный на рис.1.10.

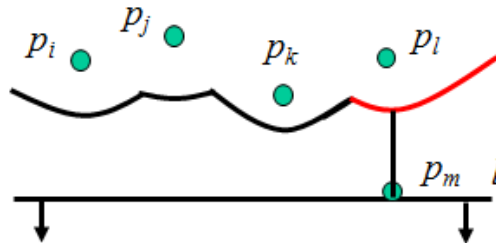


Рис. 1.10. Событие точки

Береговая линия пересекла точку  $p_m$ . Возникает событие точки. Определим дугу, которую пересечет луч от новой точки (рис. 1.11). Для этого абсциссу точки сравниваем со значениями абсцисс контрольных точек в бинарном дереве.

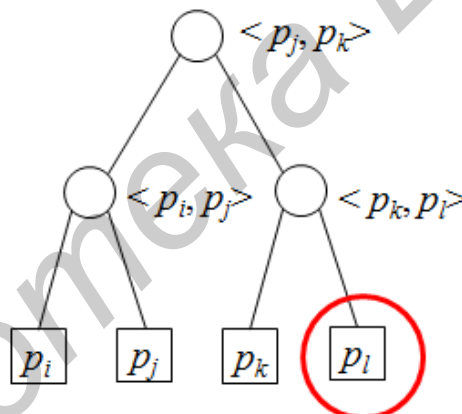


Рис. 1.11. Нахождение дуги

Соответствующий лист заменяется новым поддеревом. Стоит отметить, что разные дуги могут быть соотнесены с одной и той же точкой. Также добавим образовавшееся новое ребро в двусвязный список полуребер (рис. 2.12 ).

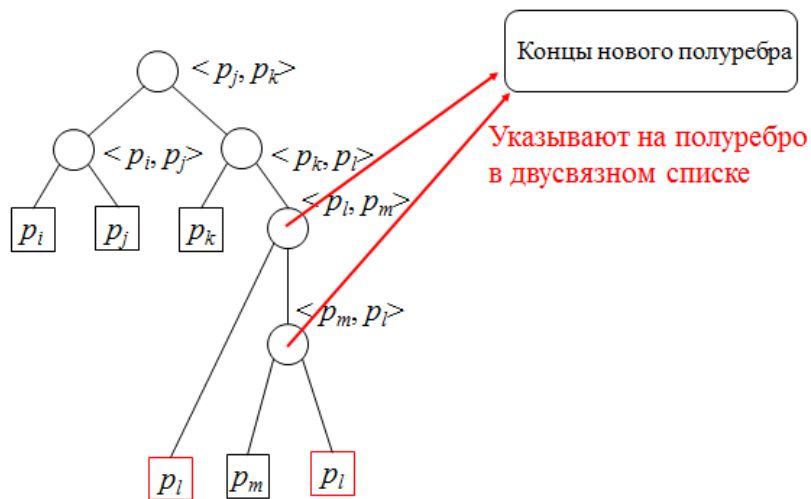


Рис. 1.12. Новые контрольные точки в бинарном дереве

На каждом новом шаге береговой линии проверяем наличие событий круга. В нашем случае следующее событие круга наступит в ситуации, когда контрольные точки  $\langle p_k, p_l \rangle$  и  $\langle p_l, p_m \rangle$  совпадут (рис. 1.13).

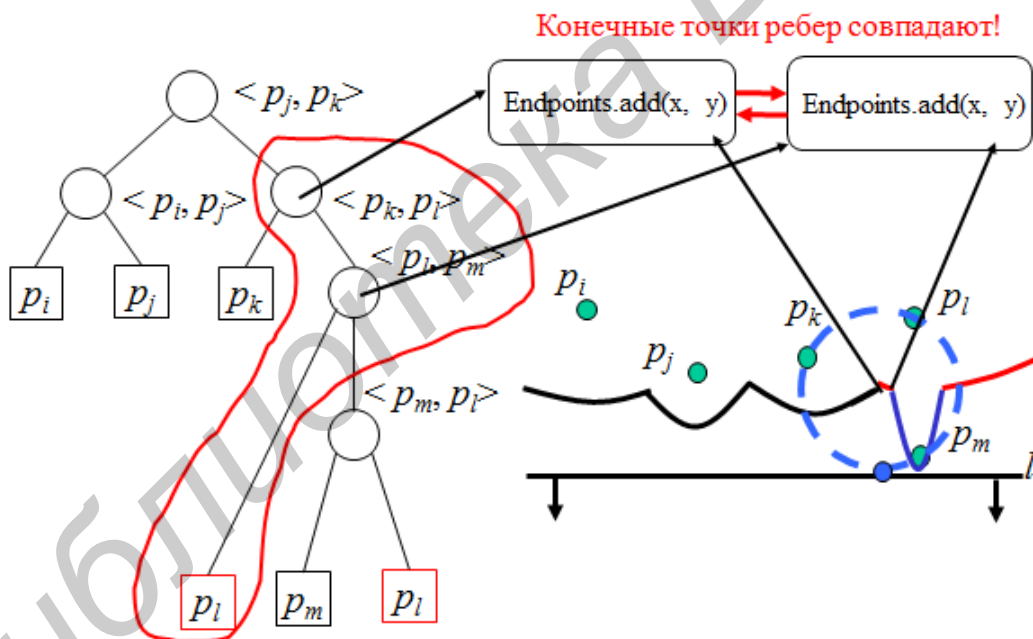


Рис. 1.13. Совпадение крайних точек двух ребер

В результате совпадения двух крайних точек различных ребер исчезает их общая дуга, которую они пересекают (рис. 1.14). Ее следует удалить из бинарного дерева.

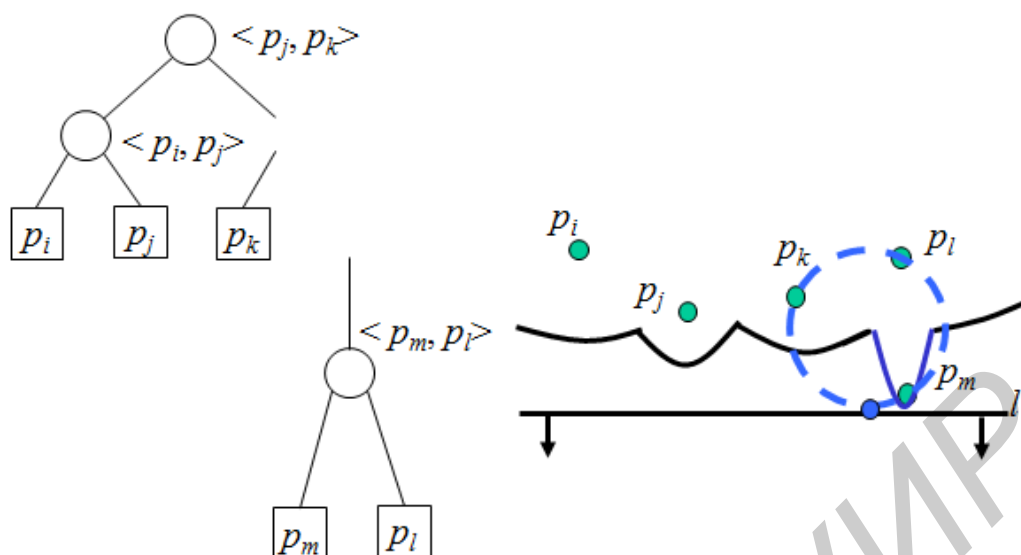
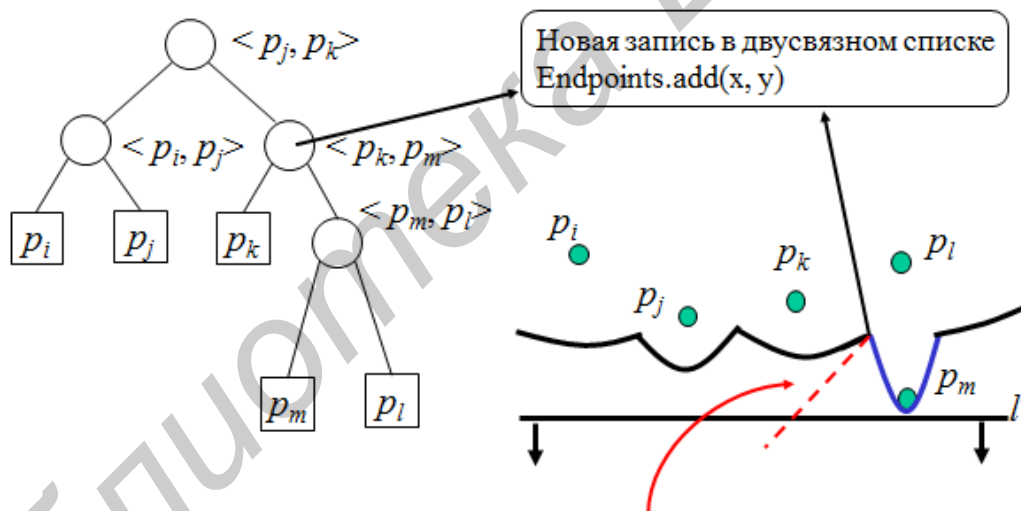


Рис. 1.14. Удаление исчезающей дуги

Создаем новую контрольную точку вместо двух слившихся (рис. 1.15).



Новое ребро создается благодаря новой контрольной точке  $\langle p_k, p_m \rangle$

Рис. 1.15. Новая контрольная точка

В очереди событий у нас больше не осталось событий точки. При движении береговой линии вниз возникнет событие круга (нижняя точка окружности – положение береговой линии при возникновении события) – рис. 1.16.

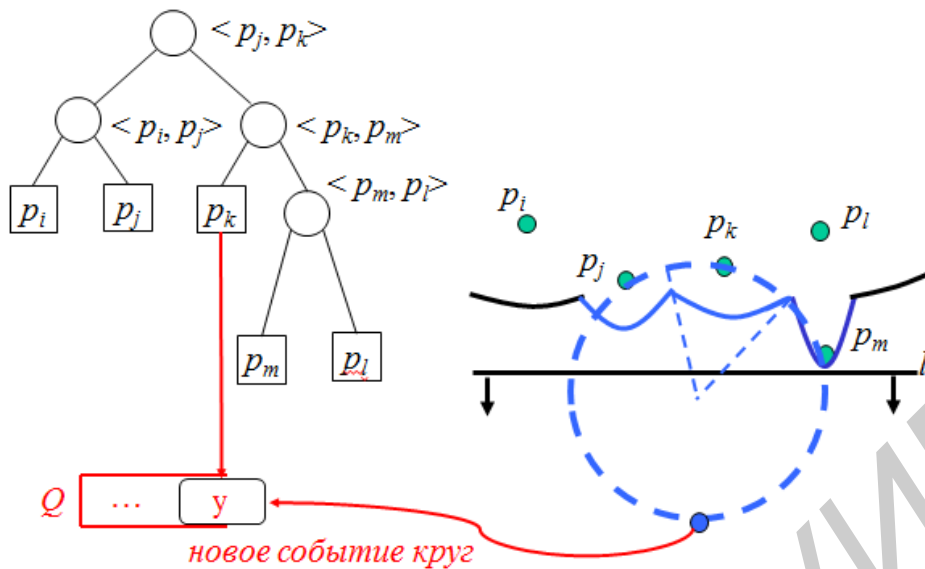


Рис. 1.16. В очереди осталось одно событие

При возникновении очередного события производим действия, аналогичные вышеописанным (рис. 1.17).

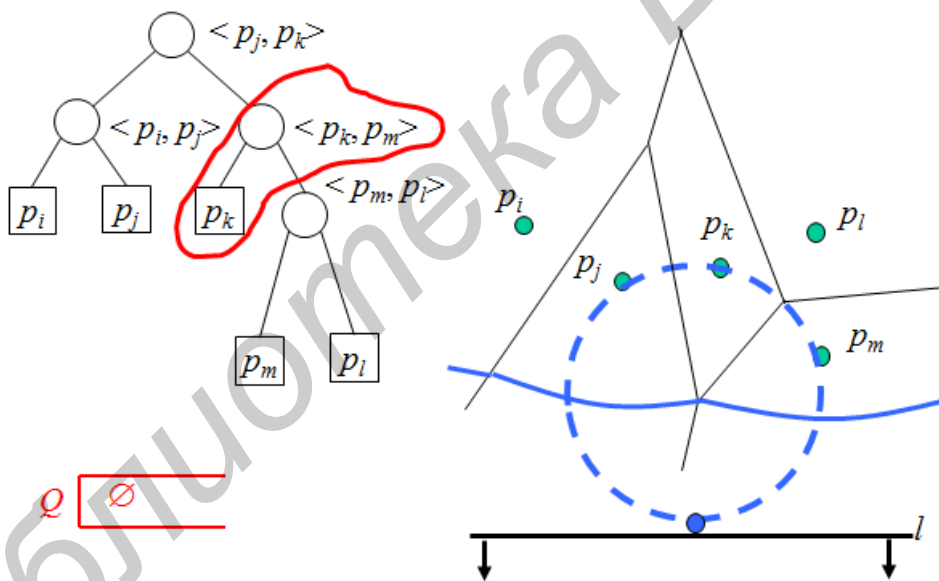


Рис. 1.17. Убираем исчезнувшую дугу

В итоге событий в очереди не осталось и бинарное дерево имеет вид, представленный на рис. 1.18. Для полного завершения работы алгоритма остается ограничить область (рис. 1.19).

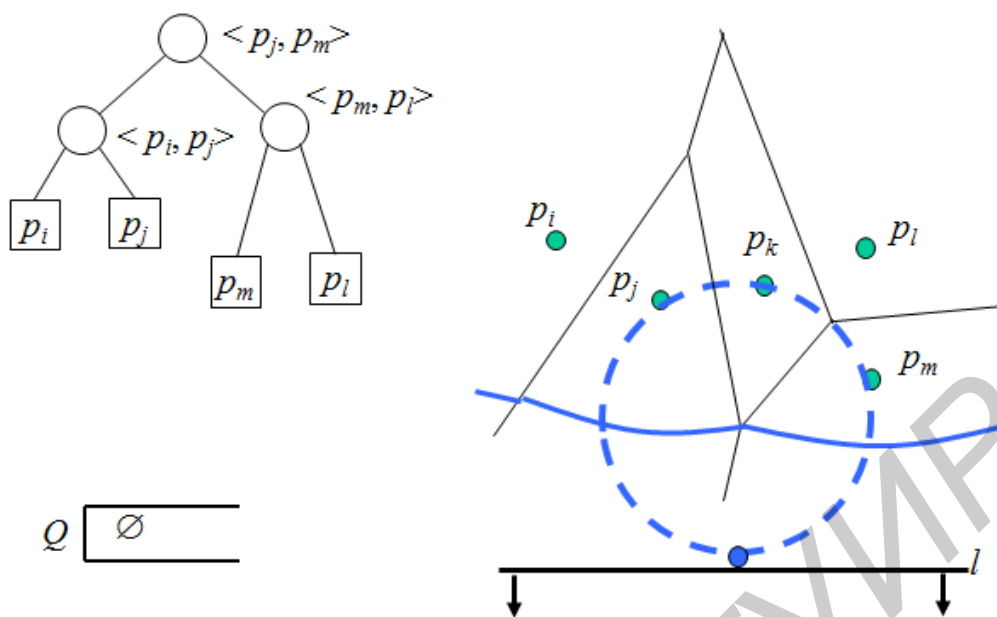


Рис. 1.18. Очередь пуста

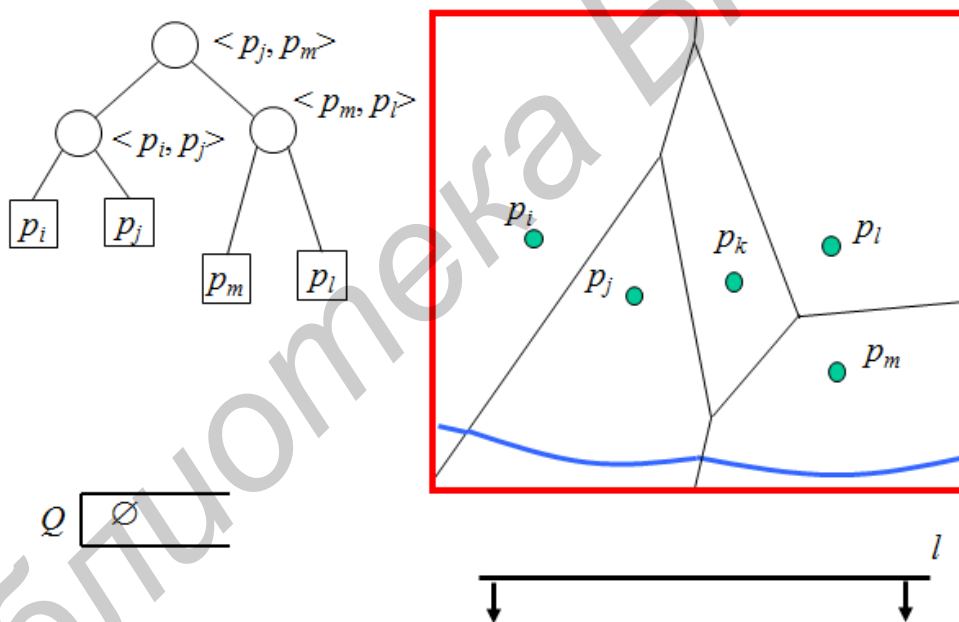


Рис. 1.19. Завершение алгоритма

### Задание к лабораторной работе

Разработать графическую программу, выполняющую триангуляцию Делоне и построение диаграммы Вороного по заданному набору точек.

## 2. УДАЛЕНИЕ НЕВИДИМЫХ ЛИНИЙ И ПОВЕРХНОСТЕЙ

### 2.1. Отсечение

**Определение.** Выделение части базы данных (БД) называется отсечением.

Отсечение применяется при удалении невидимых линий и поверхностей, формировании фактур, построении теней. Алгоритмы отсечения бывают двух- или трехмерными и применяются как к регулярным областям (прямоугольники, параллелепипеды со сторонами, параллельными осям координат), так и к нерегулярным областям и объемам. Реализация алгоритмов может быть программной и аппаратной.

### 2.2. Двухмерные преобразования

Рассмотрим двухмерные отсечения. Рассмотрим фрагмент сцены, который представлен на рис. 2.1, в котором показана плоская сцена и отсекающее окно регулярной формы.

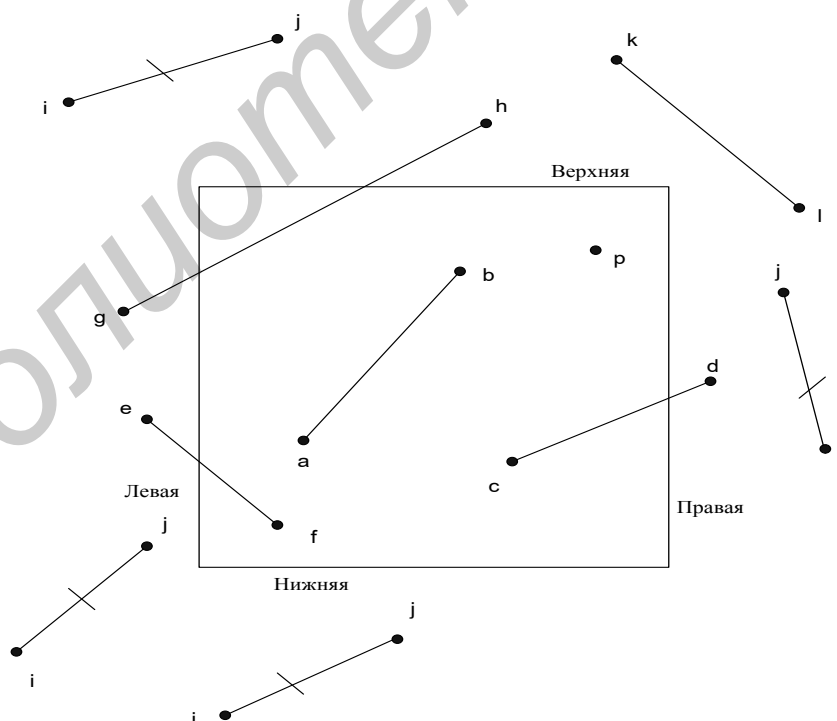


Рис. 2.1. Сцена и отсекающее окно регулярной формы

Цель алгоритма: определение тех точек, отрезков или их частей, которые лежат внутри отсекаемого окна. Эти точки, отрезки или их части остаются для визуализации, а все остальное отбрасывается.

Главное требование – эффективность, т. к. в реальных сценах приходится реально отсекалть большое количество отрезков и точек.

Во многих случаях большое число точек или отрезков лежит целиком внутри или вне отсекаемого окна. Поэтому важно уметь быстро отбирать отрезки, подобные  $ab$  или точки, подобные  $p$ , и отбрасывать отрезки, подобные  $ij$ , или точки, подобные  $q$ .

Точки, лежащие внутри отсекаемого окна, удовлетворяют условию:

$$x_l \leq x \leq x_n;$$

$$y_n \leq y \leq y_v.$$

Знак равенства показывает, что точки, лежащие на границе окна, считаются находящимися внутри него.

Отрезок лежит внутри окна и, следовательно, является видимым, если обе его концевые точки лежат внутри окна (например отрезок  $ab$ ). Однако, если оба конца отрезка лежат вне окна, то этот отрезок необязательно лежит целиком вне окна, например отрезок  $gh$ . Если же оба конца отрезка лежат справа, слева, выше или ниже окна, то этот отрезок целиком лежит вне окна, а значит, невидим. Проверка последнего условия устраняет все отрезки, помеченные как  $ij$ , но не устраняет ни отрезка  $gh$ , видимого частично, ни отрезка  $kl$ , которые целиком невидим.

**Метод Козна – Сазерленда** (осуществляет тесты полной видимости или невидимости отрезков).

Используется четырехразрядный код, который можно увидеть на рис. 2.2.

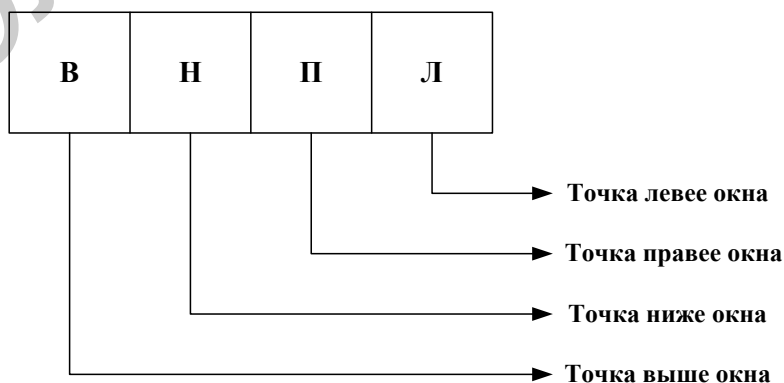


Рис. 2.2. Четырехразрядный код



Теперь можно получить коды областей рисунка «Фрагмент сцены», который представлен в виде кодировки точек относительно окна:

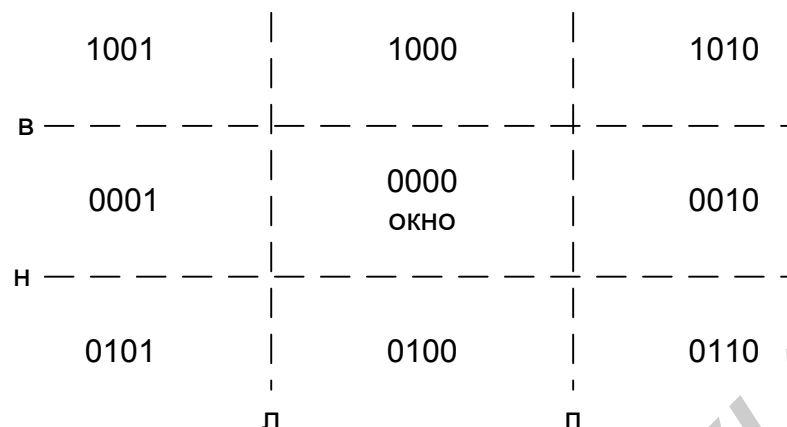


Рис. 2.3. Фрагмент сцены

«1» – точка, находящаяся выше, ниже, правее, левее окна.

1. Если побитовое логическое «И» кодов конечных точек отрезка не равно нулю, то отрезок полностью невидим, и его можно отбросить тривиально.

2. Если побитовое логическое «И» кодов конечных точек отрезка равно нулю, то отрезок может оказаться частично видимым или даже целиком видимым.

Результаты тестов Коэна-Сазерленда для данных на рис. 2.1 приведены в табл. 2.1.

Таблица 2.1

Отрезок	Коды 1 концов	Коды 2 концов	Побитовое «И»	Тест
<i>ab</i>	0000	0000	0000	Целиком видим
<i>ij</i> (1)	0010	0110	0010	Целиком невидим
<i>ij</i> (2)	1001	1000	1000	Целиком невидим
<i>ij</i> (3)	0101	0001	0001	Целиком невидим
<i>ij</i> (4)	0100	0100	0100	Целиком невидим
<i>cd</i>	0000	0010	0000	Частично видим
<i>ef</i>	0001	0000	0000	Частично видим
<i>gh</i>	0001	1000	0000	Частично видим
<i>kl</i>	1000	0010	0000	Целиком невидим

Для определения полной видимости необходимо проверить значения кодов обоих концов отрезка по отдельности. Если коды обоих концов отрезка – 0000, то отрезок целиком лежит в окне (тривиально видим).

После того как определены целиком видимые и тривиально невидимые отрезки, необходимо обработать частично видимые отрезки.

Для этого необходимо найти пересечение двух отрезков для следующих случаев:

- 1) для наклонных отрезков;
- 2) частный случай, когда отрезки вертикальны или горизонтальны.

### 1. Для наклонных отрезков

Пусть заданы две точки  $P_1(x_1, y_1)$  и  $P_2(x_2, y_2)$ . Уравнение бесконечной прямой, проходящей через эти точки, имеет вид

$$y = m * (x - x_1) + y_1 \quad \text{или} \quad y = m * (x - x_2) + y_2 ,$$

где  $m = \frac{y_2 - y_1}{x_2 - x_1}$  тангенс угла наклона данной прямой к оси абсцисс.

Точки пересечения прямой со сторонами окна имеют следующие координаты:

Таблица 2.2

	$x$	$y$	Условие
С левой	$x_l$	$m * (x_l - x_1) + y_1$	$m \neq \infty$
С правой	$x_n$	$m * (x_n - x_1) + y_1$	$m \neq \infty$
С нижней	$x_1 + \frac{1}{m} * (y_n - y_1)$	$y_n$	$m \neq 0$
С верхней	$x_1 + \frac{1}{m} * (y_v - y_1)$	$y_v$	$m \neq 0$

### 2. Для частного случая, когда отрезки вертикальны или горизонтальны

Если наклон бесконечен, то отрезок параллелен левой и правой стороне. Необходимо искать пересечение с верхней и нижней сторонами.

Если  $m = 0$ , то отрезок параллелен верхней и нижней стороне, ищем пресечение с левой и правой стороной.

## Пример

Рассмотрим отсекающее окно и отрезки, изображенные на рис. 2.3, с координатами  $P_1 (-2, -1)$ ,  $P_2 (1, 2)$ .

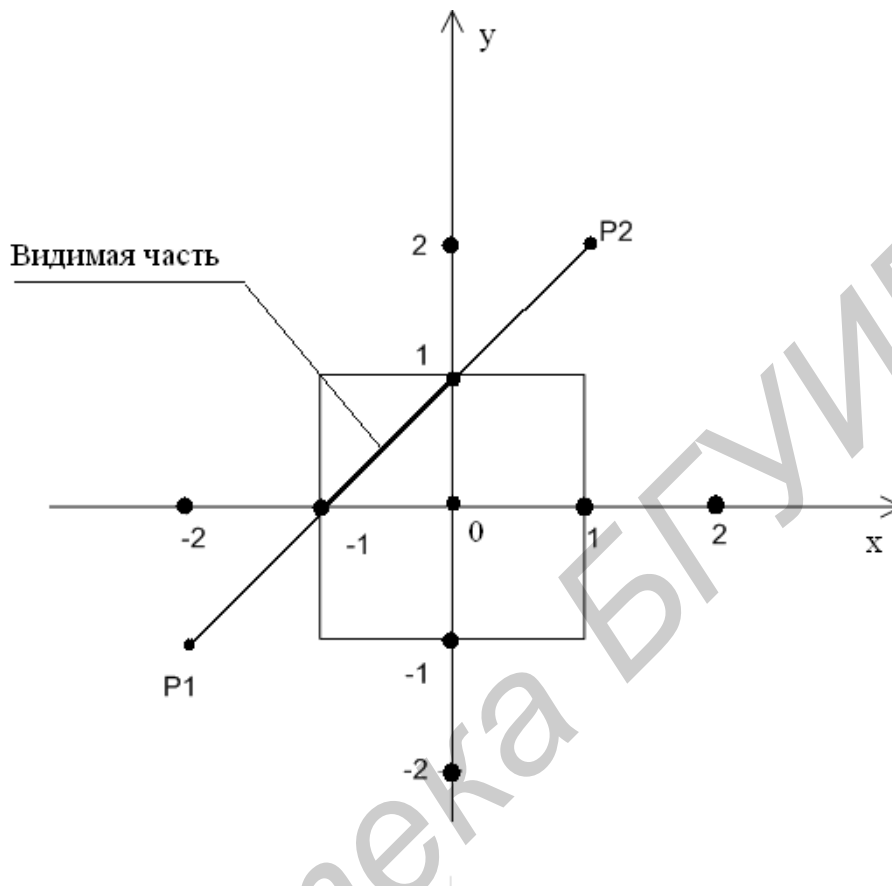


Рис. 2.4. Отсекающее окно и отрезки с координатами  $P_1 (-2, -1)$ ,  $P_2 (1, 2)$

1. Вычислим наклон отрезка:  $m = 1$ .

2. Пересечения отрезка будут следующими:

а) с левой:  $x = -1$ ;  $y = m \cdot (x_l - x_1) + y_1 = 1 \cdot (-1 + 2) - 1 = 0$ ;

б) с правой:  $x = 1$ ;  $y = m \cdot (x_n - x_1) + y_1 = 1 \cdot (1 + 2) - 1 = 2$ .

Этот случай не рассматриваем, т. к. значение  $y$  больше, чем значение  $y_0$  на единицу;

в) с верхней:  $y = 1$ ,  $x = x_1 + m \cdot (y_0 - y_1) = -2 + 1 \cdot (1 + 2) = 0$ ;

г) с нижней:  $y = -1$ ,  $x = -2 + 1 \cdot (-1 + 1) = -2$ .

Этот случай не рассматриваем, т. к. значение  $x$  больше, чем значение координаты  $x$  нижней стороны.

3. Результатом отсечения будет  $([-1;0][0;1])$ .

### 2.3. Алгоритм Козна – Сазерленда

Отсекающее окно является координатно-ориентированным прямоугольником.

Для каждой стороны окна выполнить:

- для каждого отрезка  $P_1 P_2$  определить, не является ли он полностью видимым или, может быть, тривиально отвергнут как невидимый;
- если  $P_1$  – вне окна, то продолжить выполнение, иначе – поменять  $P_1$  и  $P_2$  местами;
- заменить  $P_1$  на точку пересечения  $P_1 P_2$  со стороной окна.

### 2.4. Алгоритм разбиения средней точкой

Отсекающее окно является координатно-ориентированным прямоугольником.

В алгоритме Козна – Сазерленда следовало вычислять координаты пересечения отрезка со сторонами окна. Можно избежать непосредственного вычисления, если организовать двоичный поиск такого пересечения путем деления его средней точкой.

Программная реализация такого алгоритма медленнее, чем вышеописанного, однако аппаратная реализация быстрее и эффективнее, т. к. можно использовать параллельную архитектуру, а также сдвиги вместо операций умножения и деления на 2.

1. В алгоритме используются коды концевых точек отрезка для проверки на тривиальную видимость или невидимость.

2. Те отрезки, которые нельзя отнести к этим двум категориям, делятся средней точкой на две равные части.

3. Каждая полученная часть отрезка проверяется на тривиальную видимость или невидимость.

4. Повторить деление с оставшимися отрезками. Деление происходит до тех пор, пока не будет обнаружено пересечение с одной из сторон окна или длина делимого отрезка не станет пренебрежимо мала (т. е. не выродится в точку).

## 2.5. Алгоритм отсечения Кируса – Бэка

Прямоугольное окно повернуто относительно системы координат на некоторый угол.

В предыдущем алгоритме предполагалось, что отсекающее окно является координатно-ориентированным прямоугольником. Предположим, что прямоугольное окно повернуто относительно системы координат на некоторый угол. В этом случае неприменим ни один из описанных выше алгоритмов.

Кирус и Бэк предложили использовать алгоритм, который позволяет работать с любой произвольной выпуклой областью. В алгоритме Кируса – Бэка используется вектор нормали для определения местоположения относительно окна (внутри, на границе, вне) точки, принадлежащей отрезку.

Внутреннюю нормаль можно увидеть на рис. 2.4.

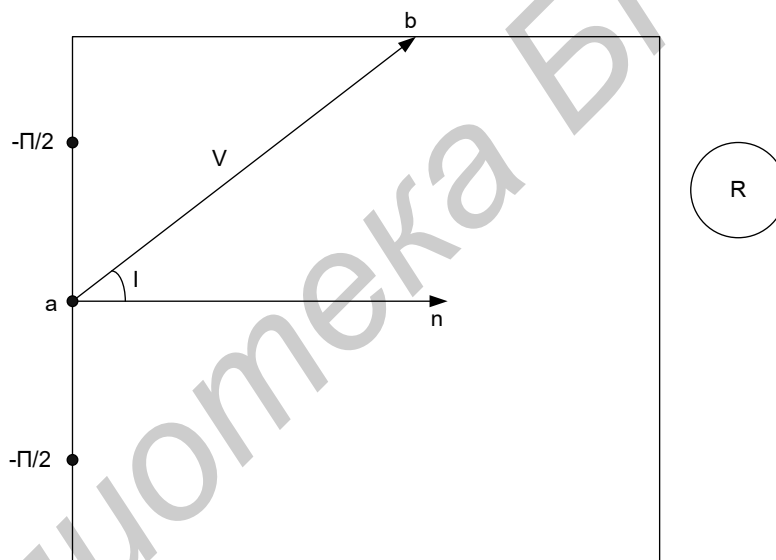


Рис. 2.5. Внутренняя нормаль

Внутренняя нормаль  $\vec{n}$  в произвольной точке  $a$ , лежащей на границе  $R$ , удовлетворяет условию

$$\vec{n} \cdot (b - a) \geq 0,$$

где  $b$  – любая другая точка на границе  $R$ .

Это условие следует из произведения двух скалярных векторов:

$$\vec{V}_1 \cdot \vec{V}_2 = \left| \vec{V}_1 \right| \cdot \left| \vec{V}_2 \right| \cdot \cos \theta$$

$$\vec{V}_1 = b - a, \vec{V}_2 = \vec{n}, -\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}.$$

Пусть задан отрезок  $p_1p_2$  следующим параметрическим уравнением:

$$P(t) = P_1 + (P_2 - P_1) \cdot t, 0 \leq t \leq 1,$$

где  $P_1$  и  $P_2$  – координаты точек отрезка.

Будем искать пересечение отрезка со сторонами окна:

пусть  $f$  – граничная точка выпуклой области,  $\vec{n}$  – внутренняя нормаль с одной из двух сторон, примыкающих к точке  $f$ .

Возможны три случая, рассмотрим их:

$$1. \vec{n} [P(t) - f] > 0.$$

$$2. \vec{n} [P(t) - f] = 0.$$

$$3. \vec{n} [P(t) - f] < 0.$$

$[P(t) - f]$  – задает вектор от точки  $f$  до точки  $P(t)$ , принадлежащей отрезку.

Это все можно представить на рис. 2.6.

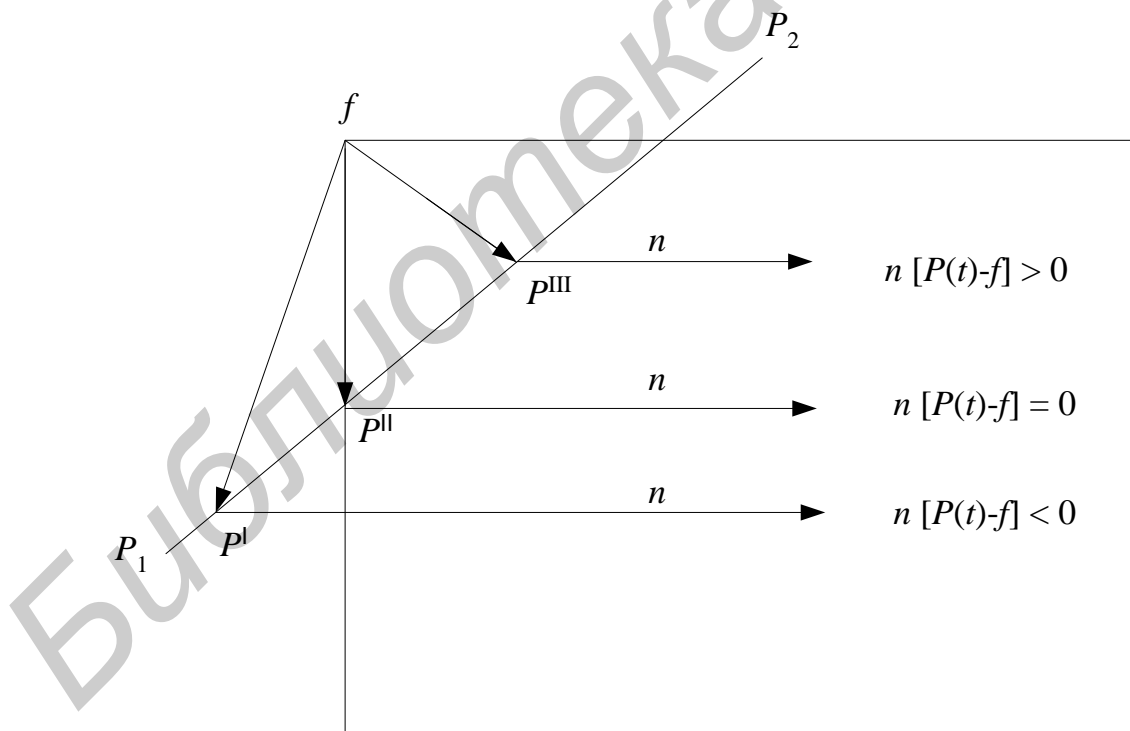


Рис. 2.6. Пересечение отрезка со сторонами окна

Если решать уравнения относительно  $t$ , получим координаты пересечения отрезка с заданным ребром.

## Пример

Частично видимый отрезок (рис. 2.7):

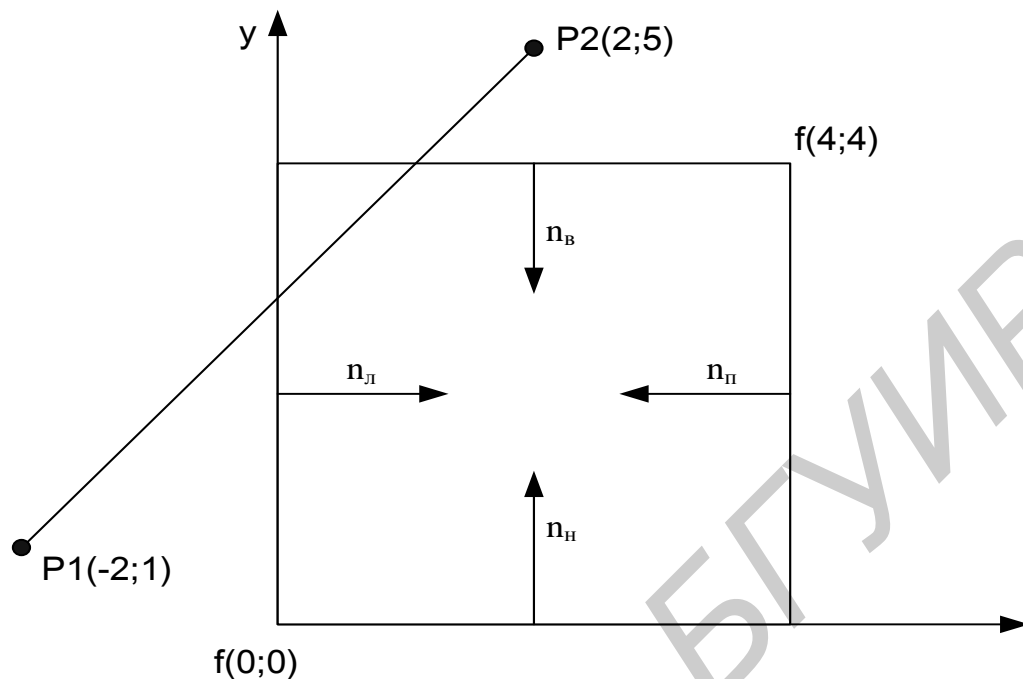


Рис. 2.7. Пример частично видимого отрезка

Составим параметрическое уравнение прямой:

$$P(t) = P_1 + (P_2 - P_1) \cdot t = [-2, 1] + [4, 4] \cdot t = (-2 + 4 \cdot t) \cdot i + (1 + 4 \cdot t) \cdot j$$

$$\vec{n}_x = i, \vec{n}_y = -i, \vec{n}_z = j, \vec{n}_w = -j.$$

Выберем две точки  $f(0,0)$  и  $f(4,4)$ :

$$f(0,0): P(t) - f = (-2 + 4 \cdot t) \cdot i + (1 + 4 \cdot t) \cdot j.$$

$$n_x(P(t) - f) = (-2 + 4 \cdot t) = 0 \Rightarrow t = 1/2.$$

$$n_y(P(t) - f) = (1 + 4 \cdot t) = 0 \Rightarrow t = -1/4.$$

$$f(4,4): P(t) - f = (-6 + 4 \cdot t) \cdot i + (-3 + 4 \cdot t) \cdot j.$$

$$n_z(P(t) - f) = -(-3 + 4 \cdot t) = 0 \Rightarrow t = 3/4.$$

$$n_w(P(t) - f) = -(-6 + 4 \cdot t) = 0 \Rightarrow t = 3/2.$$

Так как  $0 \leq t \leq 1$ , то  $t = 3/2$  и  $t = -1/4$  отбрасываем,

$$t = 1/2: [-2, 1] + \frac{1}{2}[4, 4] = [0, 3]$$

$$t = 3/4: [-2, 1] + \frac{3}{4}[4, 4] = [1, 4]$$

## Пример

Полностью видимый и невидимый отрезки (рис. 2.8):

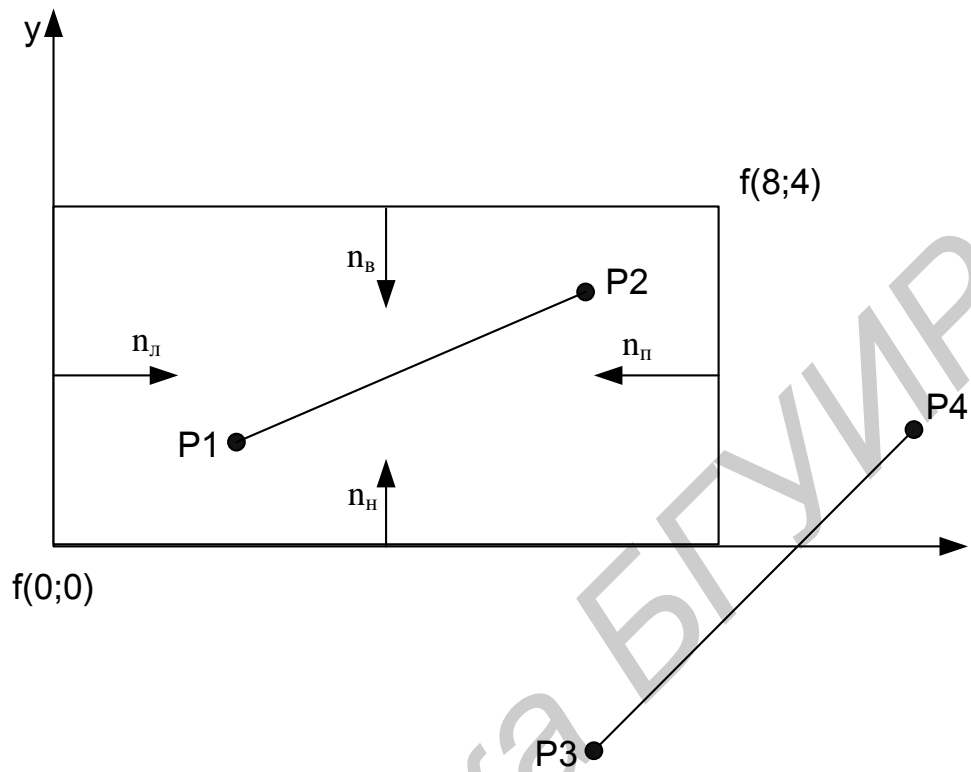


Рис. 2.8. Пример полностью видимого и невидимого отрезка

$$1) P_1P_2 : P(t) = [1,1] + [6,2] \cdot t = (1+6 \cdot t)i + (1+2 \cdot t) \cdot j$$

В результате получим:

Ребро	$n$	$f$	$P(t)-f$	$n[P(t)-f]$	$t$
Л	$i$	(0,0)	$(1+6t)i+(1+2t)j$	$1+6t$	$-1/6$
П	$-i$	(8,4)	$(-7+6t)i+(-3+2t)j$	$7-6t$	$7/6$
Н	$j$	(0,0)	$(1+6t)i+(1+2t)j$	$1+2t$	$-1/2$
В	$-j$	(8,4)	$(-7+6t)i+(-3+2t)j$	$3-2t$	$3/2$

Все значения  $t$  лежат вне предела  $[0,1]$ . С другой стороны, при  $t = 0$  и  $t = 1$   $n(P(t) - f) > 0$ . Значит, отрезок целиком виден.

$$2) P_3P_4 : P(t) = [6,-2] + [4,3] \cdot t = (6+4t)i + (-2+3t)j$$



В результате получим:

Ребро	$n$	$f$	$P(t)-f$	$n[P(t)-f]$	$t$
Л	$i$	(0,0)	$(6+4t)i+(-2+3t)j$	$6+4t$	$-3/2$
П	$-i$	(8,4)	$(-2+4t)i+(-6+3t)j$	$-(-2+4t)$	$1/2$
Н	$j$	(0,0)	$(6+4t)i+(-2+3t)j$	$-2+3t$	$2/3$
В	$-j$	(8,4)	$(-2+4t)i+(-6+3t)j$	$-(-6+3t)$	$2$

Значение параметра  $t$  при пересечении с правой и нижней сторонами допустимо, но учет ориентации отрезка от  $P_3$  к  $P_4$  показывает, что такой отрезок не может пересечь правую сторону окна при  $t = 1/2$  прежде, чем пересечет его нижнюю сторону при  $t = 2/3$ , и при этом иметь с окном общие точки. Поэтому отрезок целиком невиден.

## 2.6. Формальная запись алгоритма Кируса – Бэка

Пусть дано уравнение прямой  $P(t)$ ,  $\vec{n}_i$  – нормали к сторонам,  $f_i$  – точки, принадлежащие ребрам отсекающего окна.

Чтобы найти пересечение с ребрами, надо решить следующее уравнение:

$$n_i[P(t) - f_i] = 0, \quad \text{или} \quad n_i[P_1 + (P_2 - P_1)t - f_i] = 0,$$

$$n_i[P_1 - f_i] + n_i[P_2 - P_1]t = 0,$$

Пусть  $D = P_2 - P_1$ ,  $w_i = P_1 - f_i$ .

Тогда  $t(n_i \cdot D) + w_i \cdot n_i = 0$ ,

$$t = -\frac{w_i \cdot n_i}{D \cdot n_i}.$$

Алгоритм

1. Если  $n_i \cdot D = 0$ , то отрезок вырождается в точку либо параллелен ребру, иначе вычисляется параметр  $t$ .

2. Тривиальное отсечение. Если для всех ребер значение параметра  $t$  выходит за границы  $[0, 1]$  и, кроме того, выполняются условия:

$$\begin{cases} w_i \cdot n_i < 0 \\ w_i \cdot n_i + n_i \cdot D < 0 \end{cases}, \text{ то отрезок отсекается тривиально.}$$

3. Тривиальное изображение. Если для всех ребер значение параметра  $t$  выходит за пределы интервала  $[0, 1]$  и, кроме того, для каждого ребра выполняются условия

$$\begin{cases} w_i \cdot n_i > 0 \\ w_i \cdot n_i + n_i \cdot D > 0 \end{cases}$$

4. Изображение части отрезка.

Если  $t$  лежит в допустимых пределах, то проверяется знак  $D \cdot n_i$ .

Если  $D \cdot n_i > 0$ , то найдется точка – кандидат на нижний предел, иначе – на верхний. Когда все кандидаты вычислены, то среди нижних кандидатов выбираются  $t_n$  с максимальным значением. Среди верхних –  $t_b$  с минимальным значением.

Если  $t_b < t_n$ , то отрезок невидим, иначе  $[t_n, t_b]$  видимая часть отрезка.

## 2.7. Удаление невидимых граней. Алгоритм Робертса

Алгоритм удаления невидимых линий служит для удаления ребер и граней, которые невидимы для наблюдателя, находящегося в данной точке пространства. Все алгоритмы удаления включают в себя сортировку. Сортировка ведется по геометрическому расстоянию от тела до точки наблюдения. Предполагается, что далеко расположенные от объекта грани с большей вероятностью будут заслоняться теми объектами, которые расположены ближе. Все алгоритмы удаления невидимых граней в большей степени зависят от этого этапа сортировки.

### Алгоритм Робертса

Тела представляются в виде выпуклых многогранников, описываемых координатами вершин, а также ребрами и гранями. Алгоритм работает только с выпуклыми телами. В том случае, если тело невыпуклое, его необходимо разрезать и получить несколько выпуклых тел.

Шаги алгоритма:

**Шаг 1.** Из каждого тела удаляются ребра и грани, которые экранируются самим телом.

**Шаг 2.** Каждое из видимых ребер каждого тела сравнивается со всеми оставшимися телами для определения того, какая его часть экранируется.

Алгоритмическая сложность алгоритма может быть уменьшена с помощью различных характеристик: сортировка по удаленности и т. д.

В нашем случае будем рассматривать только первую часть алгоритма, т. е. удаление ребер и граней, которые экранируются самим телом.

Тело описывается множеством его граней. Каждая грань может быть задана уравнением плоскости, на которой она лежит.

Уравнение плоскости:

$$[x \ y \ z \ 1]^* \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = 0 \quad (*)$$

Уравнения всех плоскостей можно объединить в матрицу тела, которая имеет вид

$$V = \begin{bmatrix} a_1 & \dots & \dots & a_n \\ b_1 & \dots & \dots & b_n \\ c_1 & \dots & \dots & c_n \\ d & \dots & \dots & d_n \end{bmatrix} .$$

Для определения, где находится точка на плоскости (перед или за плоскостью), надо осуществить следующее.

Пусть задана точка в однородных координатах. Тогда, если  $(*)=0$ , то точка лежит на плоскости, если  $(*)<0$  или  $(*)>0$ , то соответственно за или перед плоскостью.

Предполагается, что точки, лежащие внутри тела, должны давать отрицательные значения при подстановке. Корректность матрицы тела проверяется умножением на нее точки, которая заведомо лежит внутри тела. Если получившееся значение  $< 0$ , то матрица корректна, в противном случае столбцы-нарушители умножаются на  $-1$ .

Для получения уравнения плоскости возможны следующие методы.

1. По трем неколлинеарным точкам. Для этого необходимо решить систему линейных уравнений:

$$ax_1 + by_1 + cz_1 = -1$$

$$ax_2 + by_2 + cz_2 = -1$$

$$ax_3 + by_3 + cz_3 = -1$$

где  $(x_i; y_i; z_i)$  – координаты точек.

2. По вектору нормали.

Необходимо найти вектор нормали:

$$\vec{n} = \vec{V}_1 \otimes \vec{V}_2.$$

Пусть вектор нормали  $\vec{n}(a,b,c)$ . Тогда уравнение плоскости будет иметь вид  $ax+by+cz+d = 0$ . Параметр  $d$  можно вычислить подстановкой в уравнение плоскости произвольной точки, лежащей на этой плоскости.

Возможны две интерпретации алгоритма Робертса:

#### 1. Интерпретация с БУТ.

Подставим в уравнение плоскости координаты БУТ, и если результат  $> 0$ , то точка находится перед плоскостью и грань видна, а если результат  $< 0$ , то точка находится за плоскостью и грань не видна.

#### 2. Интерпретация по скалярному произведению между вектором направления и внутренней нормалью.

Видимость грани определяется по знаку скалярного произведения вектора направления и внутренней нормали.

В итоге для определения видимых и невидимых граней необходимо умножить координаты БУТ или направляющего вектора на матрицу тела.

#### Пример

Для куба со стороной 2 и центром в начале координат матрица тела будут иметь вид

$$V = \begin{bmatrix} B & H & П & Л & П\partial & 3\partial \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix}.$$

Если задан вектор направления  $\vec{P} = [-1 \ 0 \ 0 \ 0]$ , то в итоге будем иметь матрицу, в которой со значением  $-1$  стоят видимые грани, а со значением  $1$  – невидимые, т. е.

$$\vec{P} * \vec{V} = \begin{bmatrix} B & H & П & Л & П\partial & 3\partial \\ -1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Таким образом, определяем невидимые и видимые грани.

### Задание к лабораторной работе

Разработать графическую программу, выполняющую отсечение невидимых линий двумерных объектов и удаление невидимых граней трехмерных объектов.

### 3. КОГНИТИВНАЯ ГРАФИКА

В настоящее время принято выделять в компьютерной графике (КГ) следующие направления:

- изобразительная компьютерная графика (ИКГ);
- обработка и анализ изображений;
- анализ сцен (перцептивная компьютерная графика);
- компьютерная графика для научных абстракций (когнитивная компьютерная графика, т. е. графика, способствующая познанию).

**Изобразительная компьютерная графика.** Объектами изобразительной компьютерной графики являются синтезированные изображения. Задачи ИКГ можно сформулировать следующим образом:

- построение модели объекта и генерация изображения;
- преобразование модели и изображения;
- идентификация объекта и получение требуемой информации.

**Обработка и анализ изображений.** Объектами этого вида КГ являются дискретное числовое представление фотографий. В задачи обработки и анализа изображений входят:

- повышение качества изображения;
- оценка изображения – определение формы, местоположения, размеров и других параметров требуемых объектов;
- распознавание образов – выделение и классификация свойств объектов (обработка аэрокосмических снимков, ввод чертежей, системы навигации, обнаружения и наведения).

В основе данного вида КГ лежат специальные методы обработки и анализа изображений и изобразительная компьютерная графика для представления результатов.

**Анализ сцен** предназначен для исследования абстрактных моделей графических объектов и взаимосвязей между ними, причем графические объекты могут быть как синтезированными, так и выделенными на фотоснимках. В основе анали-

за сцен (перцептивной компьютерной графики) находятся специализированные средства анализа изображений и изобразительная графика.

**Когнитивная компьютерная графика** – это компьютерная графика для научных абстракций, способствующая рождению нового научного знания. Когнитивная компьютерная графика представляет собой эффективный инструмент воздействия на образное интуитивное мышление исследователя. Функция когнитивной графики заключается в наглядном изображении содержания предмета, которым может быть, в частности, любое абстрактное научное понятие, гипотеза или теория.

Содержанием когнитивной графики является отображение (визуализация) неких математических закономерностей для их более глубокого понимания <sup>1</sup>. В современных информационных технологиях **когнитивная графика** определяется как совокупность приемов и методов образного представления условий задачи, которое позволяет либо сразу увидеть решение, либо получить подсказку для его нахождения.

Известно, что человек использует два механизма мышления:

- логический (связан с правым полушарием); этот механизм работает с абстрактными последовательностями символов (объектов), семантикой символов, прагматическими представлениями, связанными с символами;
- интуитивный, образный (связан с левым полушарием); обеспечивает работу с чувственными образами и представлениями о них.

Важно отметить, что мозг не только умеет работать с двумя способами представления информации, но может соотносить эти два способа и совершать переходы от одного представления к другому.

Поэтому, если удачно представить входные данные какой-либо задачи в виде рисунка, когнитивного образа, то при анализе этой информации правым полушарием мозга будет включен интуитивный механизм мышления и ответ может быть найден без сложных вычислений с помощью интуиции человека. Для того чтобы представить условия задачи в графическом виде, следует выбирать такие графические образы, в которых можно удачно сопоставить каждое из условий задачи с отдельной частью изображения. Выбранный графический образ должен позволять исследователю (аналитику, лицу, принимающему решения) использовать свойства

---

<sup>1</sup> Поспелов Д. А. Серые и/или черно-белые. Специальный выпуск.–М.: ВЦРАН, 1994. № 1. 290 с.

выбранного абстрактного изображения для визуального решения поставленной задачи.

Известный специалист в области искусственного интеллекта Д. А. Поспелов сформулировал три основные задачи когнитивной графики [2]:

1) создание моделей представления знаний, в которых можно представить как объекты, характерных для логического (символического, алгебраического) мышления, так и объекты, характерные для образного мышления;

2) визуализация тех знаний, для которых пока невозможно подобрать текстовые описания;

3) поиск путей перехода от образа к формулировке гипотезы о механизмах и процессах, представленных этими образами.

**Иллюстративная и когнитивная функция компьютерной графики.** В настоящее время компьютерная графика – одно из наиболее быстро развивающихся направлений информационных технологий. Характерно, что исследователи все чаще переходят от иллюстративной функции ИКГ к использованию тех возможностей ИКГ, которые позволяют активизировать свойственную человеку способность мыслить сложными пространственными образами. В связи с этим начинают четко различать две функции ИКГ: иллюстративную и когнитивную.

**Иллюстративная функция ИКГ** позволяет дать адекватное визуальное оформление лишь того, что уже известно, т. е. уже существует либо в окружающем нас мире, либо как идея в голове исследователя. **Когнитивная функция ИКГ** состоит в том, чтобы с помощью некоего КГ-изображения получить новое, т. е. еще не существующее даже в голове специалиста знание или по крайней мере способствовать интеллектуальному процессу получения этого знания.

### **Графический анализ в интеллектуальных системах**

Анализ геометрической структуры данных широко применяется на этапе формирования знаний. Формирование знаний – это задача обработки баз данных (БД) с целью перехода к базам знаний (БЗ). В БД накапливаются и хранятся эмпирические факты из исследуемой предметной области (фактические данные, примеры экспертных заключений, элементарные высказывания с некоторой оценкой и т.п.), представленные в виде троек < объект, признак, значение признака >. В БЗ заносятся сведения, выражающие закономерности структуры множества эмпирических фактов, соответствующие контексту прикладной задачи.

Контекст определяет отношения между объектами БД. Он может задаваться вне БД (например экспертом) или порождаться признаком или совокупностью признаков из БД. Чаще всего на практике встречаются отношения эквивалентности и порядка. Отношения эквивалентности присущи задачам классификации, диагностики и распознавания образов. Отношения порядка свойственны задачам шкалирования, прогнозирования. Методы формирования знаний имеют много общего с решением классификации, диагностики и распознавания образов. Отличительной чертой первых является функция интерпретации закономерностей, положенных в основу правил вхождения объектов в классы эквивалентности. Поэтому в инженерии знаний наибольшее распространение получили логические методы. Эти методы довольно сложны в реализации.

Альтернативу логическим методам составляет геометрический подход. Он позволяет перевести задачу формирования знаний на язык геометрических соотношений между эмпирическими фактами и отображаемыми точками в пространстве признаков. Это делает более понятными критерии, и принципы построения правил вхождения объектов в определенные классы эквивалентности, которые основываются на сравнении объектов с помощью мер, имеющих интерпретацию расстояний. Использование геометрического подхода при значительном увеличении числа эмпирических фактов приводит к минимальным ошибкам при принятии решений. Визуализация геометрической структуры множества точек позволяет организовать исследование зависимостей в совокупности эмпирических фактов средствами интерактивной когнитивной графики. Можно получать визуальное представление о логических закономерностях в структуре данных.

### **3.1. Графические методы анализа данных**

#### **3.1.1. Цель графических методов анализа данных**

Целью графических методов является простое и интересное смысловое (содержательное) представление одномерных и многомерных данных и результатов. Благодаря этому функциональному назначению статистическую графику считают видом когнитивной графики [2]. Графические методы позволяют решать задачи на следующих этапах анализа данных:

- предварительный (разведочный) анализ;
- выбор модели;



– реализация методов и интерпретация результатов.

Для проведения графического анализа необходимо создать абстрактный образ данных. Отдельный объект из исследуемой совокупности данных может быть представлен с помощью одного из следующих геометрических образов в  $p$ -мерном пространстве признаков  $R^p$  ( $p$  – число признаков, характеризующих объект):

– *точкой* – в случае, если объект характеризуется набором значений признаков или считается, что все его признаки известны с абсолютной точностью; координатами точки являются значения соответствующих признаков;

–  $p$ -мерной *сферой* или *эллипсоидом* – если задается погрешность или допустимое отклонение положения объекта в абстрактном пространстве данных относительно гипотетического точного положения;

–  $p$ -мерным *параллелепипедом* – если погрешность (или допуск) задаются отдельно для каждого признака;

– *отрезком* прямой, параллельной одной из координатных осей – если один из признаков неизвестен, а остальные известны точно; длина отрезка отражает априорный допустимый диапазон, в котором может находиться пропущенное значение;

– куском  $k$ -мерной *плоскости*, если пропущены  $k$  значений из набора признаков объекта;

– куском  $k$ -мерного *слоя* некоторой толщины, если  $k$  значений признаков в наборе пропущены, а остальные известны с некоторой точностью, при этом толщина слоя отражает значение точности (или допуска).

После сопоставления каждому из объектов геометрического образа в абстрактном многомерном пространстве данных возникает облако из геометрических объектов, которое и отражает структуру исследуемых данных. Изучая облако, тем самым будем изучать исходные данные.

**Разведочный анализ данных.** Разведочный анализ выделяет структуру и особенности исходных данных. Он предшествует этапу выбора модели и служит обнаружению отклонений от свойств модели. Графические процедуры разведочного анализа предназначены для решения следующих основных задач:

- 1) визуализация наблюдений с помощью различных графических образов;
- 2) обнаружение аномальностей в данных;

3) поиск шкалы (преобразования), в которой анализ данных существенно упрощается. Использование шкал, отличающихся от исходных, помогает достичь симметрии, постоянства дисперсии, линейности или аддитивности.

**Выбор модели.** На этапе выбора модели графические методы позволяют:

- 1) оценить соответствие анализируемых данных вероятностному распределению, используя для отображения свойств распределения графические объекты;
- 2) выявить невыполнение допущений модели с помощью графической галереи образов, соответствующих определенным видам моделей.

**Реализация методов анализа данных.** Визуализация процесса статистического анализа и его результатов в виде графических образов:

- 1) расширяет возможности содержательной интерпретации, особенно в многомерных методах;
- 2) способствует формированию нового знания об особенностях структуры исследуемых данных и выявленных закономерностях;
- 3) обеспечивает компактность представления информации.

### 3.1.2. Роль визуализации в анализе данных

Многомерный анализ данных (АД) связан с интерпретацией и пониманием информации высокой размерности. Исходная информация представляется  $n$  наблюдениями  $p$ -мерной случайной переменной  $X = (X_1, X_2, \dots, X_p)$ , имеющей  $p$  компонент. Каждая компонента  $X_i$  ( $i = 1, \dots, p$ ) – это одномерная случайная величина. Довольно часто у исследователя отсутствует априорная информация о статистическом или причинном механизме порождения имеющихся у него данных. Прежде чем применять методы, обеспечивающие принятие решений или статистический вывод, необходимо построить некоторую статистическую модель данных (формальные описания их структуры). Обоснованный выбор модели данных начинается с визуального представления исходной информации, выявления ее структуры. Это позволит получить ответы на ряд вопросов, среди которых можно выделить следующие:

- существуют ли компоненты  $X$ , имеющие большее рассеивание по сравнению с другими;
- можно ли выделить в имеющейся выборке объема  $n$  однородные группы данных;

- есть ли резко выделяющиеся наблюдения, сильно отклоняющиеся от центра распределения (резко выделяющиеся наблюдения);
- близко ли распределение данных к нормальному;
- какова структура статистической связи в исследуемых данных.

Ответы на поставленные вопросы можно искать математическими методами. Однако чаще всего в качестве компактного и понятного исследователю способа описания структуры данных или структуры зависимости переменных используется графическое представление этой структуры.

Визуализация данных предполагает получение тем или иным способом их графического отображения. Это позволяет исследователю путем непосредственного визуального анализа изображения выявить, найти структуру данных.

Известно, что человек имеет преимущества перед автоматическими устройствами и компьютерами при распознавании некоторых видов структур, особенно типа «скоплений» (кластеров). Наиболее явно эти преимущества проявляются при выделении однородных групп. Алгоритмы автоматической классификации не могут правильно разделить исходное множество на группы при наличии точек соприкосновения (рис. 3.1, а, б) и частичного перекрытия групп (рис. 3.1, в), при объединении в одном кластере нескольких удаленных друг от друга групп (рис. 3.1, г) и при сложной форме кластеров (рис. 3.1, д, е). В то же время человек, наблюдая взаимное расположение объектов, достаточно хорошо справляется с задачей группирования элементов подобных множеств в тех случаях, когда размерность пространства описания не превышает трех.

При анализе информации задача представления данных в виде изображений является одной из важнейших. Такие изображения часто являются единственной информационной моделью изучаемого явления или объекта. От полноты и ясности их содержания зависят верность суждений, прогностическая ценность выводов, скорость и надежность принятия решений. Изображение на экране должно быть синтезировано по некоторому информационному описанию, например, по таблице данных, не являющемуся исходно изображением, что позволяет рассматривать синтезируемое изображение как информационный образ этого описания.

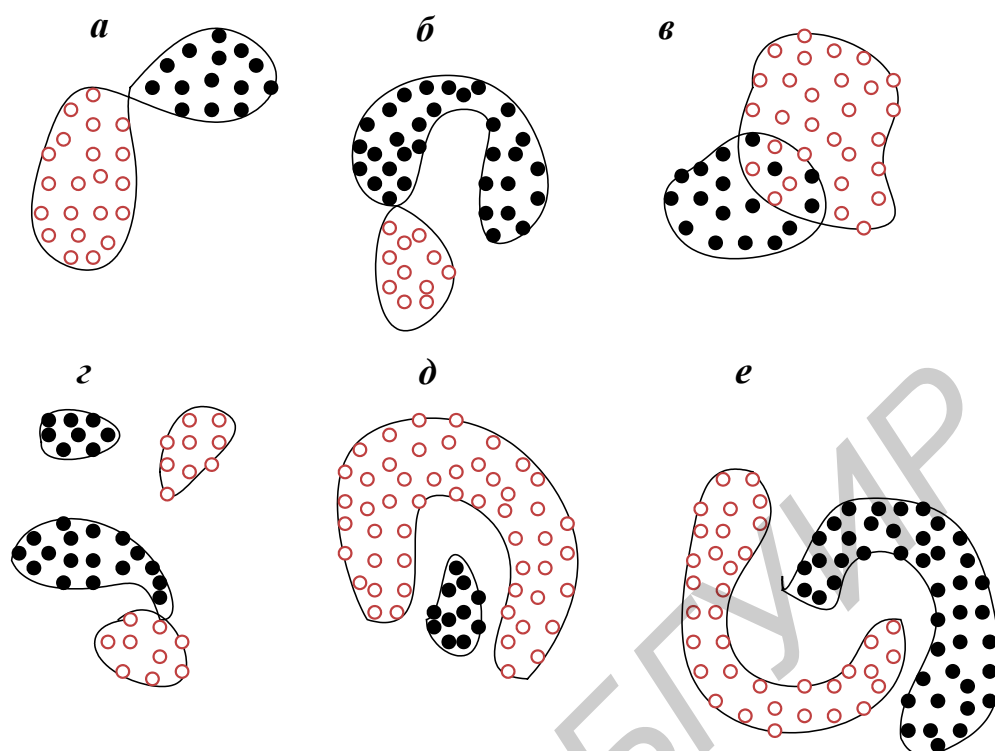


Рис. 3.1. Виды кластеров, для которых затруднено выполнение автоматических процедур классификации (различными значками отмечены объекты различных групп)

Способность зрительного анализатора по переработке и распознаванию изображений наиболее полно раскрывается при графических методах представления информации. В этих случаях глаз человека обнаруживает пространственные и временные изменения данных, которые иногда трудно выявить математическими методами. Адекватной формой представления данных в виде изображения являются скопления точек, семейства графиков, контурные рисунки, сюжетные картины. Выбор той или иной формы представления диктуется как особенностями решаемой задачи, так и возможностями инструментальных средств.

До появления современных информационных технологий визуализация рассматривалась только в качестве вспомогательного средства при анализе информации, служила лишь способом наглядного представления закономерностей, найденных при помощи традиционных методов анализа данных. В настоящее время визуализация является полноправным инструментом анализа эмпирических данных. Визуальный анализ данных основан на принципе перевода абстрактных представлений об исследуемых объектах в графические образы, что означает генерацию ко-

гнитивных изображений числовой информации, представляющей либо результаты наблюдений, либо результаты анализа информации.

Представление чисел в виде графических образов, позволяющих увидеть закономерность, – в этом и заключается смысл визуализации как способа анализа данных. Говоря о визуализации, подразумевают компьютерную графику, способствующую выявлению нового знания и выполняющую когнитивную функцию за счет преобразования числовой информации в графические изображения.

Графическое отображение может быть получено непосредственно в пространстве исходных переменных. Для визуализации многомерных данных применяют специальные методы формирования образов объектов и переменных размерностью не более 3. Выходная размерность данных может быть и больше 3, но для графического отображения все равно берутся какие-либо одна, две или три их координаты, обычно при этом первые координаты более информативны и используются для визуального анализа в первую очередь. Следует подчеркнуть, что при любом способе отображения необходимо наиболее полно сохранять существенные характеристики информации, содержащейся в данных.

### **3.1.3. Пространственное представление структуры многомерных данных**

Несмотря на широко распространенное мнение о существовании большого количества методов пространственного представления структуры исходных данных, на самом деле все они представляют собой варианты сравнительно небольшого набора основополагающих моделей геометрического представления структуры исходных данных.

При анализе данных, представленных матрицей данных, возникают три типа задач:

- получить структуру объектов в исходном пространстве;
- выявить взаимоотношения между признаками;
- получить короткое описание распределения объектов в преобразованном пространстве.

Решение этих задач наряду со сжатием информации имеет большое содержательное значение: если удастся понизить размерность пространства, построить короткое описание распределения объектов, то можно предположить, что вскрыта некоторая объективная закономерность, обусловившая возможность такого описания.

Выше было определено, что предметом визуального анализа является поиск структуры в данных. Полезно проводить различие между двумя типами структуры: теоретическими моделями и моделями-образами. Первая из них представляет собой полное математическое описание множества или подмножества определенного вида данных. Математическая модель – это абстракция реального мира, в которой интересующие исследователя отношения между реальными элементами заменены подходящими отношениями между математическими объектами [5]. Это обеспечивает стандартное статистическое использование модели. Например, мы говорим о регрессионной модели, модели Бокса – Дженкинса анализа временных рядов, кластерной и дискриминантной моделях и т.д. В противоположность моделям первого типа модель-образ является локальной структурой, характеризующей относительно небольшое число объектов. Задачей визуального анализа является обоснованный выбор среди множества возможных теоретических моделей той, которая наилучшим образом соответствует имеющемуся графическому отображению, характеризующему реальное поведение изучаемой системы.

Пусть данные заданы в виде матрицы данных  $X = (x_{ij})$ ,  $i = 1, \dots, p$ ,  $j = 1, \dots, n$ . Графически объекты можно представить в виде точек  $n$  в многомерном ( $p$ -мерном) пространстве. Для описания структуры этого множества точек используется одна из следующих статистических моделей – графических образов:

- 1) модель «облака» точек, примерно эллипсоидальной конфигурации;
- 2) кластерная модель, т. е. совокупность нескольких «облаков» точек, достаточно далеко отстоящих друг от друга;
- 3) модель «засорения» (компактное «облако» точек и при этом присутствуют далекие выбросы);
- 4) модель носителя точек как многообразия (линейного или нелинейного) более низкой размерности, чем исходное;
- 5) дискриминантная модель, когда точки разделены некоторым образом на несколько групп и дана информация о их принадлежности к той или иной группе.

В рамках модели 4 можно рассматривать и регрессионную модель, когда соответствующее многообразие допускает функциональное представление  $X_{II} = F(X_I) + \varepsilon$ , где  $X_I$  и  $X_{II}$  – две группы переменных из исходного набора (переменные из  $X_I$  носят тогда название прогнозируемых переменных, а из  $X_{II}$  – предсказываемых переменных);  $\varepsilon$  – ошибка предсказания.

Графические образы структуры бывают двух типов. Одни из них предназначены для демонстрации соответствия данных и определенной теоретической модели, другие – для обнаружения отклонения от модели. Например, графики для идентификации резко выделяющихся наблюдений или остатков регрессии аномально высокой плотности являются инструментом визуализации возможных отклонений от модели.

В пространстве переменных для описания структуры зависимостей между переменными часто используются следующие модели: модель независимых переменных, модель линейно зависимых переменных, древообразная модель зависимости, факторная модель для линейно зависимых переменных, кластерная модель (произвольные коэффициенты связи), иерархическая модель зависимости.

Реальные данные обычно лишь приближенно могут следовать перечисленным выше моделям, так как модель является идеализацией реальной ситуации. Ее адекватность зависит от того, насколько обоснованы и соответствуют модели предположения, на которых модель базируется. Более того, структура данных может не подходить ни под одну из указанных в описании моделей даже приближенно. С этой точки зрения наиболее интересные и неожиданные графические образы, выявленные в процессе анализа, могут оказаться свидетельствами неадекватности данных.

В этом разделе будут рассмотрены некоторые графические методы, применяемые как на этапах разведочного анализа данных, так и на этапе подбора модели. Будут описаны графики «ящик с усами», представляющие собой инструмент анализа одномерных данных, двумерные и трехмерные диаграммы разброса, весьма полезные для понимания природы взаимосвязи между признаками, характеризующими объект, и позволяющие выявить однородные группы или кластеры в исходных данных. Кроме того, будут показаны способы отображения многомерных данных в виде графиков под названием «лица Чернова».

#### **3.1.4. График «ящик с усами»**

График в виде «ящика с усами» представляет собой метод отображения распределения переменных. Этот графический объект позволяет одновременно увидеть положение центра распределения, характеристики рассеивания, асимметрию,

длину «хвоста» распределения и резко выделяющиеся, аномальные (outliers) наблюдения (рис. 3.2).

Верхняя  $UBV$  и нижняя  $LBV$  границы ящика соответствуют оценкам первой и третьей квартилей, т.е. 50 % значений исследуемой переменной лежат в этом ящике. На графике отмечается положение среднего и медианы. «Усы», отходящие от горизонтальных сторон ящика, показывают наиболее удаленную точку, не являющуюся аномальным наблюдением. Наблюдения  $X_i$ , выходящие из этого диапазона, определяются как аномальные. Для этого необходимо выполнение следующих условий:

$$X_i > UBV + c (UBV - LBV),$$

$$X_i < LBV - c (UBV - LBV),$$

где  $UBV$  – значение верхней границы ящика, равно либо третьей квартили, либо среднему плюс стандартная ошибка;  $LBV$  – значение нижней границы ящика, равно либо первой квартили, либо среднему минус стандартная ошибка;  $c$  – коэффициент для определения резко выделяющихся наблюдений.

На рис. 3.2 показан график «ящик с усами», построенный для изучения распределения численности населения 15 крупнейших городов США в 1960 г. Диапазон, в котором лежат наблюдения, не являющиеся выбросами, соответствует населению Нью-Орлеана и Чикаго. Численность населения Нью-Йорка – это аномальное наблюдение.

Из рис. 3.2 видно, что распределение анализируемой переменной является сильно скошенным: верхняя половина данных (лежащая выше медианы) имеет большее рассеивание по сравнению с нижней (ниже медианы). Кроме того, взаимное положение медианы и среднего говорит о наличии асимметрии и ее виде: если среднее меньше медианы, то распределение скошено влево; если среднее больше медианы, то распределение скошено вправо. В рассматриваемом примере среднее значение больше медианы, следовательно, распределение численности населения скошено вправо.

Графики «ящик с усами» могут быть использованы для сравнения совокупностей данных. На рис. 3.3 представлены результаты сравнения экономичности легковых автомобилей, произведенных в различных странах (США, Японии и европейских странах)<sup>1</sup>.

---

<sup>1</sup>Адрес доступа [www.mdtech.de](http://www.mdtech.de)



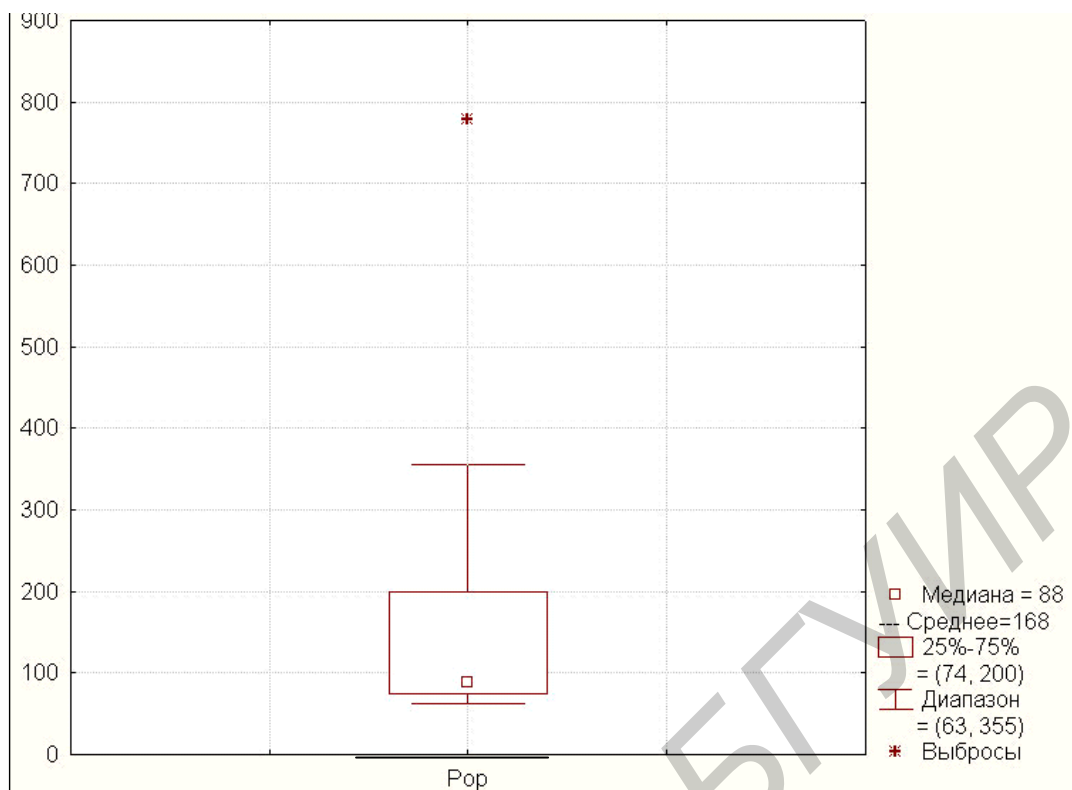


Рис. 3.2. График «ящик с усами»

На основании графика можно сделать следующие выводы:

- японские автомобили обладают большей экономичностью по сравнению с автомобилями США и европейскими автомобилями;
- автомобиль марки VW – Rabbit Diesel показал наибольшую экономичность (отмечен как выброс);
- основная часть значений экономичности автомобилей США (представлена ящиком) лежит ниже, чем у японских автомобилей;
- самый плохой японский автомобиль более экономичен, чем почти половина автомобилей США;
- значение медианы для японских автомобилей выше, чем у европейских автомобилей и автомобилей США.

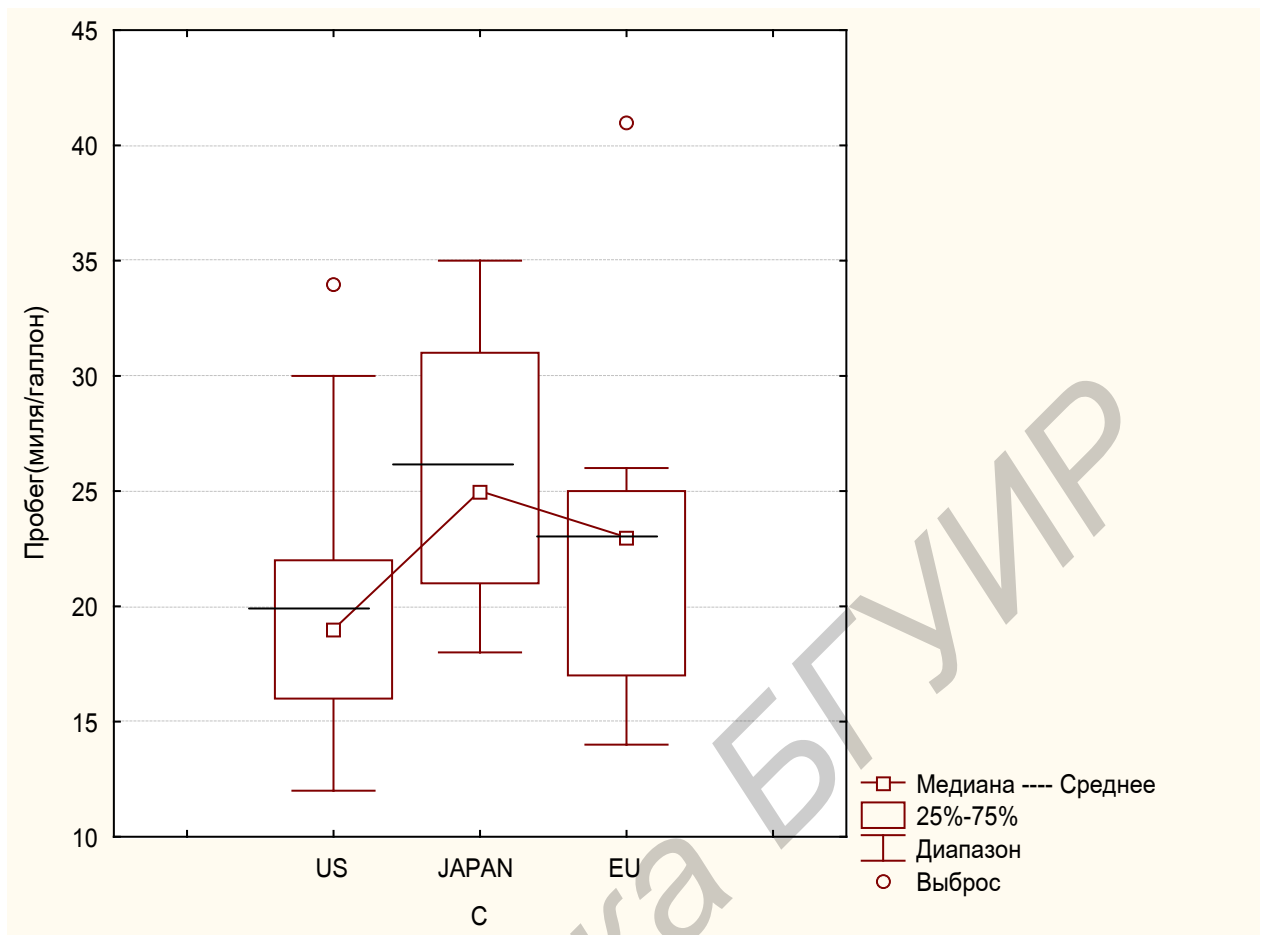


Рис. 3.3. «Ящики с усами» для отображения экономичности автомобилей США, Японии и Европы

## Выводы

График «ящик с усами» и его отдельные элементы являются информационной моделью, с помощью которой устанавливаются характеристики и свойства распределения:

- «ящик» дает представление о медиане и среднем значении, является характеристиками центра распределения;
- относительное положение медианы и среднего внутри «ящика» определяют степень асимметрии распределения;
- высота «ящика» и длина «усов» являются мерами рассеивания;
- длина «усов» служит индикатором длины «хвоста» распределения;
- график выявляет наличие аномальных наблюдений в выборке, нетипично далеко удаляющихся от центра распределения;

– сравнение распределений (либо различных переменных, либо одной переменной в различных группах) можно выполнять, сравнивая относительное положение и размер как «ящика», так и его «усов»;

– график «ящик с усами» не выявляет мультимодальность или наличие кластеров.

### 3.2. Диаграммы рассеивания

Рассмотрим вопросы визуализации многомерных данных, связанные с использованием диаграмм рассеивания (ДР), которые являются широко распространенной формой графического представления данных. Вместо термина «диаграммы рассеивания» иногда используются термины «диаграмма разброса», «скаттерограмма» (от английского слова *scatter* – рассеивать, разбрасывать).

ДР многомерных данных является визуальной формой представления результатов некоторого отображения исходной матрицы данных в двухмерное или трехмерное евклидово пространство. Роль исходной матрицы данных может играть матрица «объект – свойство» или матрица близостей (отношений «объект – объект», «переменная – переменная»). В качестве отображенных на ДР единиц могут выступать объекты, переменные, категории переменных (если переменные неколичественные).

При работе с диаграммами рассеивания удобно интерпретировать анализируемые объекты как точки в соответствующем признаковом пространстве. Если исходные данные представлены в виде матрицы ( $X$ ), то эти точки являются геометрическим изображением многомерных наблюдений  $X_1, X_2, \dots, X_n$  в  $p$ -мерном пространстве с  $p$  координатными осями. Для двухмерной ( $p = 2$ ) ДР на координатных осях откладываются значения любых двух признаков, характеризующих объект (рис. 3.4), в трехмерной ( $p = 3$ ) диаграмме рассеивания – значения любых трех признаков (рис. 3.5). ДР является «облаком» точек различной конфигурации.

Можно построить так называемый матричный график, представляющий собой матрицу, составленную из диаграмм рассеивания для каждой пары переменных, описывающих объект (рис. 3.6). Этот графический образ является отображением матрицы корреляций.

Диаграммы рассеивания помогают понять взаимосвязи, существующие в исходных переменных. Отрицательный наклон ДР указывает, что тенденция изменения признаков, откладываемых на осях графика, противоположная: при увеличе-

нии одного другой уменьшается и наоборот. Положительный наклон свидетельствует об одинаковой тенденции изменения (рис. 3.7).

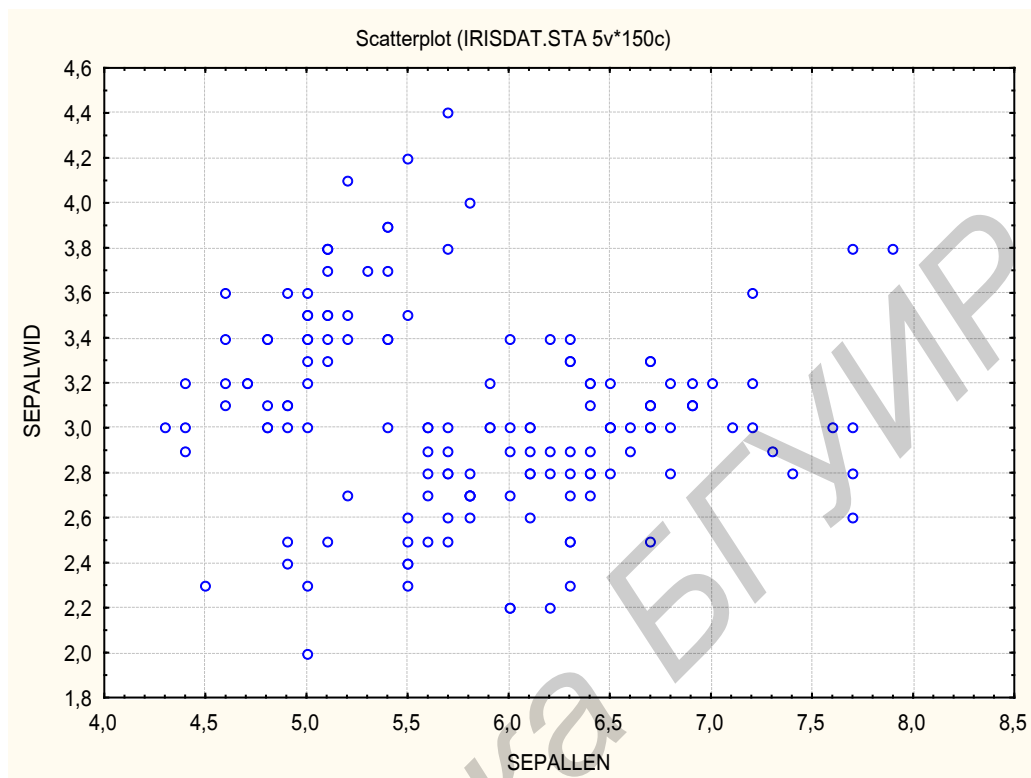


Рис. 3.4. Двухмерная диаграмма рассеяния

Отображение взаимосвязи между тремя переменными может быть выполнено с помощью трехмерной диаграммы рассеяния (рис. 3.5). Одно из основных преимуществ такого вида отображений состоит в том, что в них реализована возможность интерактивного изучения отношений между переменными. Другой областью применения трехмерных диаграмм рассеивания является идентификация кластеров и подмножеств в выборке из неоднородной совокупности на основе изучения эмпирического распределения трех и более переменных. В многомерном анализе данных трехмерные диаграммы рассеивания часто применяются для визуализации результатов, как, например, в факторном анализе или многомерном шкалировании.

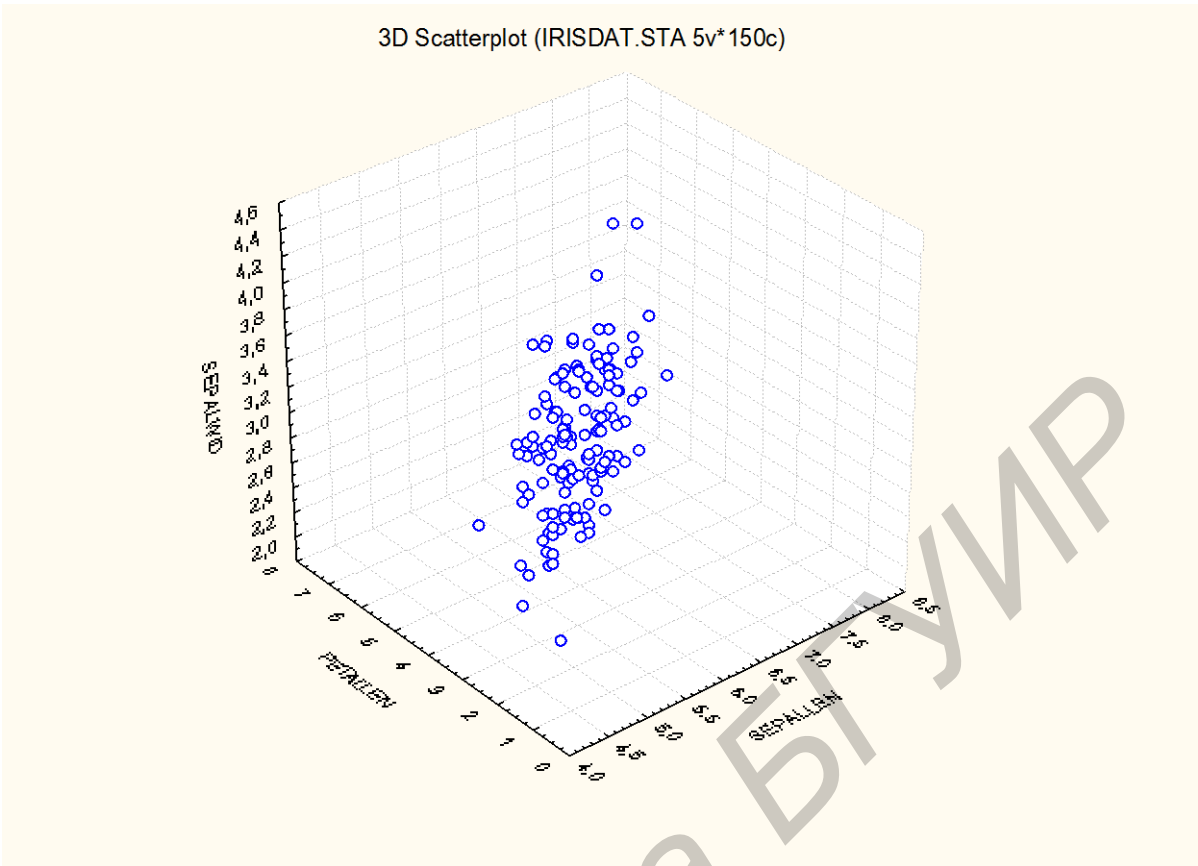


Рис. 3.5. Трехмерная диаграмма рассеивания

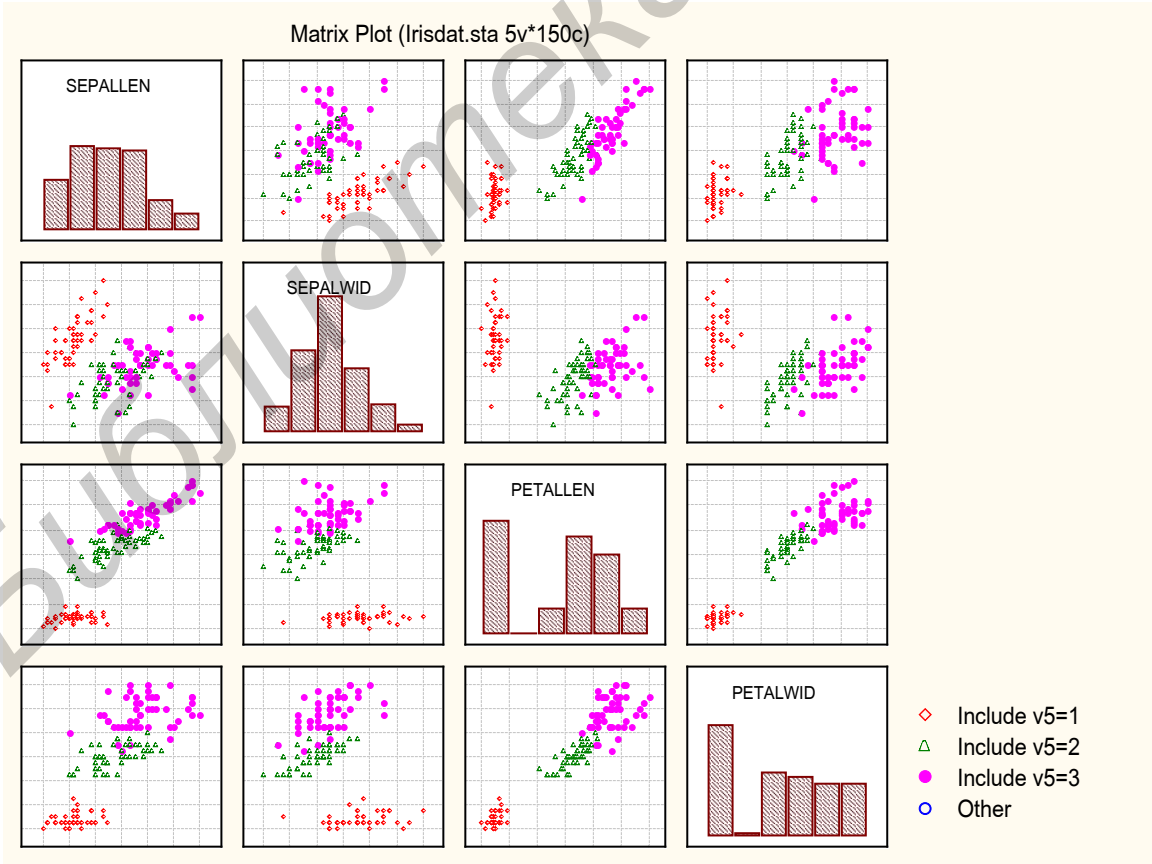


Рис. 3.6. Матричный график – графический образ матрицы корреляции

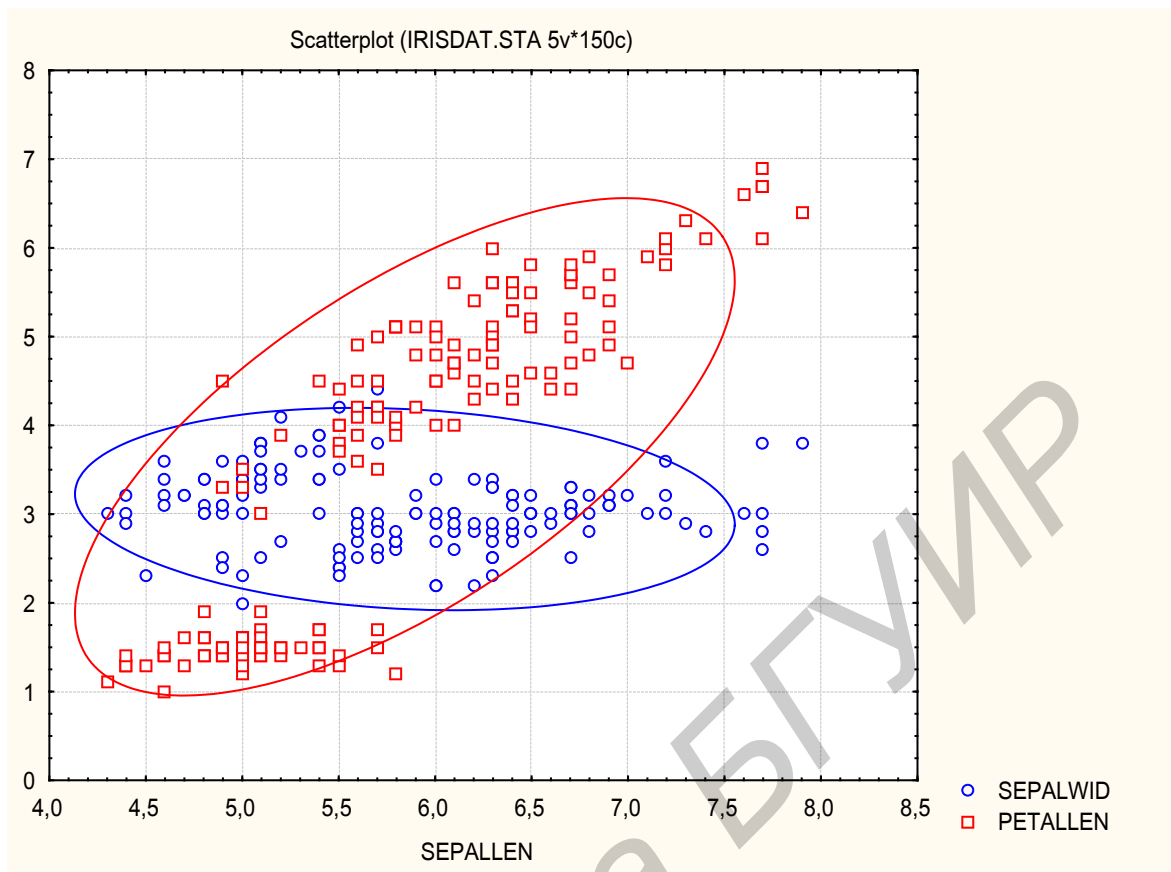


Рис. 3.7. Отображение направления взаимосвязи признаков

С целью получения лучшего представления данных полезно осуществить вращение трехмерной ДР. Фактически поворот означает преобразование многомерного наблюдения в одну или более линейных комбинаций компонент вектора наблюдений. На рис. 3.8 приведен вид трехмерной диаграммы рассеяния после поворота структуры, изображенной на рис. 3.5. В результате найдена проекция, выявляющая наличие однородной группы точек.

Трехмерные изображения целесообразно использовать для представления сложных зависимостей. Поскольку отображение трехмерного графика на плоскости задается указанием угла зрения и перспективой, очень важной функцией является возможность интерактивного вращения трехмерного графического образа (рис. 3.9).

С помощью такого инструмента можно взглянуть на график с разных сторон и выбрать именно тот вариант, который наилучшим образом отображает исследуемую структуру зависимостей.

Более того, часто именно вращение изображения помогает выявить скрытую форму кривой или поверхности, особенно если оно сочетается с возможностью изменения перспективы.

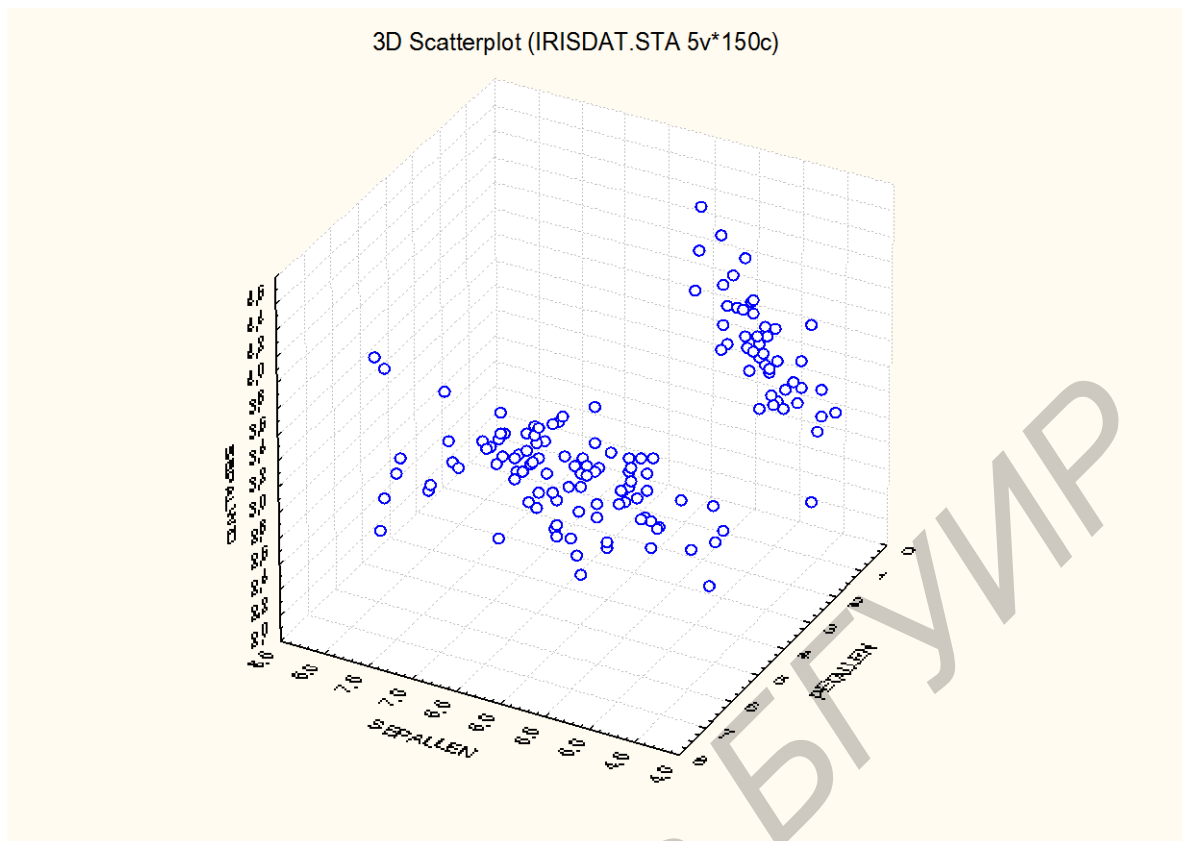


Рис. 3.8. Трехмерная диаграмма рассеивания после поворота

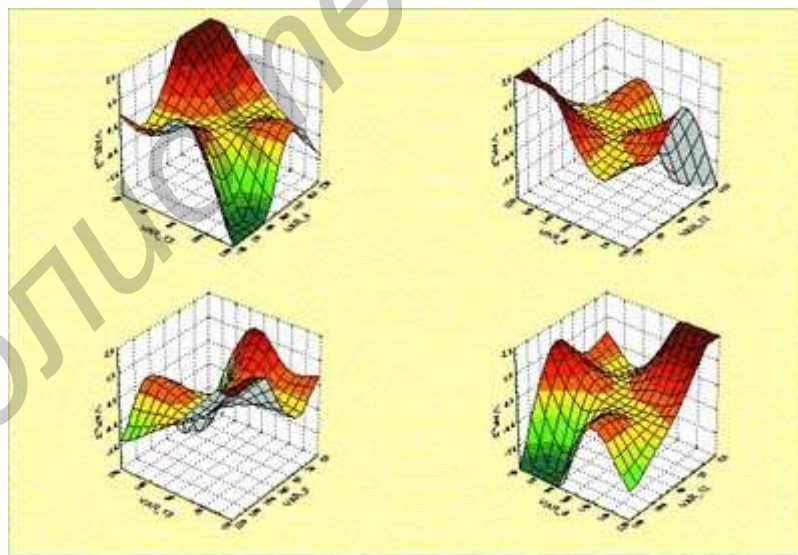


Рис. 3.9. Интерактивное вращение трехмерного графика

Если две переменные статистически сильно связаны, то на ДР точки образуют некоторые регулярные формы – прямолинейные, криволинейные и т. д. (см. рис. 3.6).

Можно подобрать функцию, позволяющую установить вид зависимости между переменными. Если переменные не связаны, точки образуют «облако» в виде круга (рис. 3.11).

Диаграммы рассеяния могут выявлять модель «засорения» – компактное «облако» точек и при этом присутствуют далекие выбросы, т. е. аномально большие или аномально малые значения анализируемых переменных (рис. 3.10). Присутствие таких значений в выборке может привести к увеличению или уменьшению коэффициента корреляции. Если результат проверки гипотезы об однородности этого наблюдения с остальной выборкой отрицательный, то такую точку можно удалить с поля графика в интерактивном режиме, используя инструмент, называемый закрасиванием.

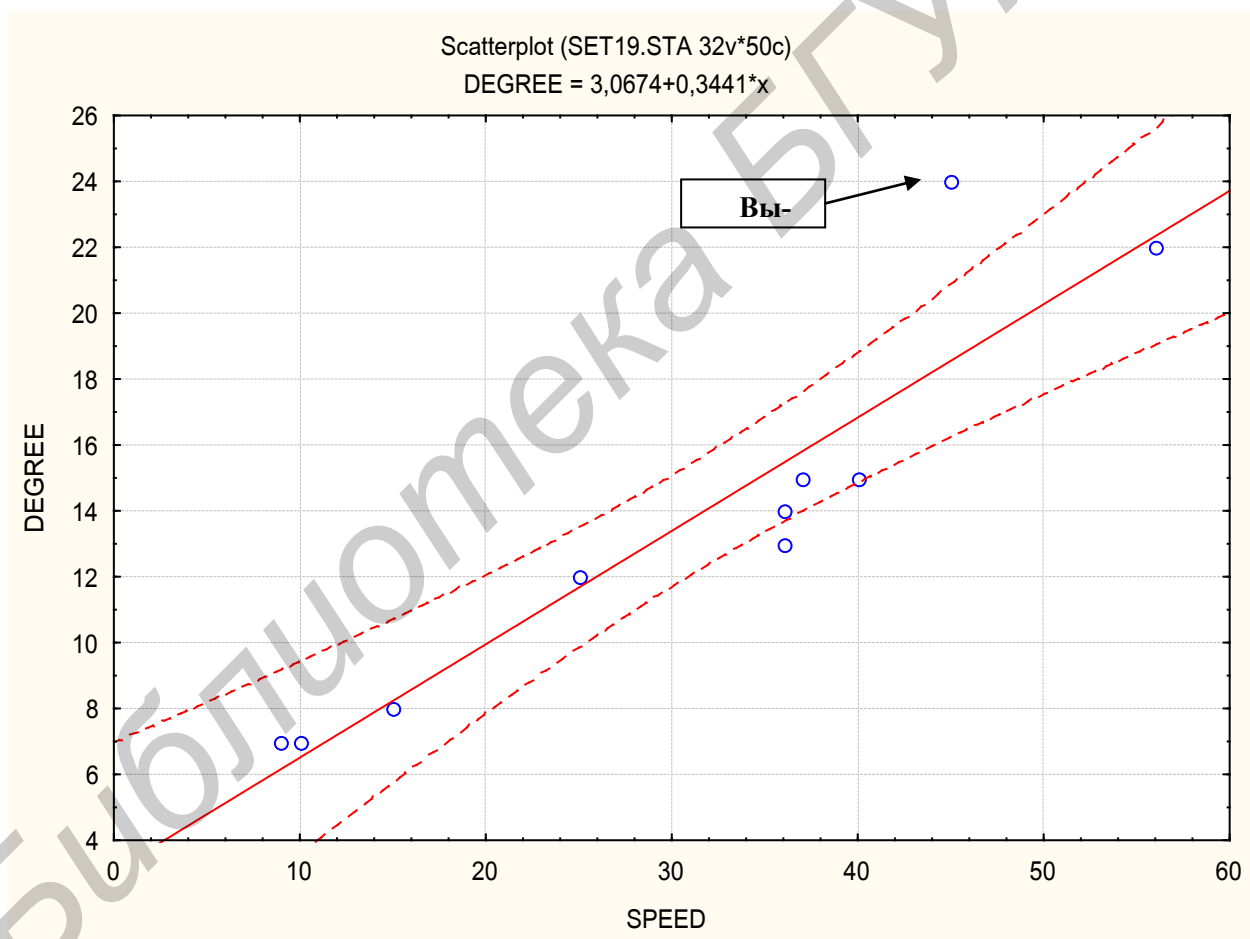


Рис. 3.10. Идентификация резко выделяющихся наблюдений с помощью диаграммы рассеивания

Диаграммы рассеяния дают возможность выявить в общей совокупности группы однородных данных. Например, на матричном графике (см. рис. 3.6) четко выделяется одна группа объектов для всех пар анализируемых признаков.



Помимо отдельных графиков диаграммы рассеивания можно отобразить одновременно на одном, составленном из двумерных ДР для всех пар переменных матрицы данных. Такой график называется матричным графиком. Он позволяет выявить знания о зависимости и структуре анализируемых данных, т. к. содержит информацию о силе, характере и структуре взаимосвязи признаков, однородности объектов. Используя технику подсвечивания и окрашивания наблюдений с помощью инструмента, называемого интерактивным закрашиванием, можно выявить внутренние связи в ДР и изучать условную зависимость.

Интерактивное закрашивание – это чрезвычайно удобный инструмент, который позволяет работать с отдельными точками, изображенными на конкретном графике, не затрагивая все исходные данные. Можно, например, интерактивно исключать некоторые аномальные точки из графика и наблюдать за изменением аппроксимирующей функции или взаимной зависимостью.

## **Выводы**

Диаграммы рассеивания являются визуальной формой представления результатов некоторого отображения исходной матрицы данных в двумерное или трехмерное евклидово пространство. Они позволяют:

- проанализировать структуру данных с точки зрения наличия в ней статистической зависимости, идентифицируя отдельные точки, резко выделяющиеся наблюдения или группы однородных объектов;
- установить положительный или отрицательный характер взаимосвязи двух (трех) переменных;
- определить тип зависимости (прямолинейная, криволинейная) и сделать предположения об ее форме;
- выявить внутренние связи и изучать условную зависимость в интерактивном режиме, используя закрашивание;
- с помощью интерактивного вращения трехмерного изображения можно найти вариант, который наилучшим образом отображает исследуемую структуру зависимостей и выявить скрытую форму кривой или поверхности.

### **3.2.1. Диаграммы рассеивания в регрессионном анализе**

Для проверки выполнения допущений модели регрессии широко используется графический анализ остатков [5]. Остаток (необъясненное рассеяние) – это часть

общего рассеяния, которое не может быть объяснено моделью регрессии. Оно равно разности между наблюдаемым значением зависимой переменной ( $y_i$ ) и значением, предсказанным моделью ( $\hat{y}_i$ ):  $e_i = y_i - \hat{y}_i$ ,  $i = 1, \dots, n$ . Чаще всего остатки для исследования адекватности модели регрессии используют диаграммы рассеивания остатков в зависимости от:

- 1) предсказанных значений  $\hat{y}_i$  зависимой переменной;
- 2) некоторых объясняющих переменных  $x_i$ ;
- 3) значения остатка, полученного в предшествующий момент времени.

Если модель корректна и допущения выполняются, то на любой ДР остаток должен выглядеть как случайная переменная с нулевым средним. Любой иной графический образ остатков будет предполагать наличие неадекватности модели. Чтобы подчеркнуть важность графического анализа, Anscombe<sup>1</sup> представил четыре различных набора данных, которые дали идентичные результаты оценки коэффициентов регрессии, сумм квадратов, квадрата коэффициента множественной корреляции. Полученная модель одинаково хороша во всех четырех случаях, если основываться только на количественных результатах. Диаграммы рассеивания для зависимой ( $Y$ ) и независимой ( $X$ ) переменных показывают существенные различия в структуре данных.

Для первого набора данных прослеживается линейная зависимость между  $Y$  и  $X$  со случайным разбросом точек, лежащих выше и ниже линии регрессии. Вторым набором данных задана конфигурация точек, соответствующую квадратической зависимости, из ДР ясно, что линейная модель неадекватна и требуется ввести квадратичный член. Диаграмма рассеивания, построенная для третьего набора данных, иллюстрирует строгую линейную зависимость между  $Y$  и  $X$  за исключением одной резко выделяющейся точки. Исключение этой точки приводит к тому, что сумма квадратов остатков будет равна нулю. Диаграмма рассеивания указывает на наличие проблемы либо в данных, либо в модели. Если эта точка не является аномальной, значит, модель неадекватна. Четвертым набором данных является пример ситуации, когда регрессионная зависимость определяется на основании наблюдений, в которых все значения независимой переменной одинаковы. В этом случае трудно ожидать адекватности модели, т. к. оценки получены на основании одного наблюдения.

---

<sup>1</sup> F. J. Anscombe. Graphs in statistical analysis. The American Statistician, 27:17–21, 1973.

Рассмотренный пример подчеркивает силу простых графических методов для выявления неадекватности модели регрессии. Для этой цели можно использовать ряд информативных графиков.

Рассмотрим основные виды графиков остатков и исследуемые с их помощью свойства модели регрессии.

**Распределение ошибки регрессии.** Малые отрицательные значения остатков и небольшое число больших положительных остатков могут служить свидетельством положительной асимметрии распределения остатков (предполагается нормальное распределение с нулевым математическим ожиданием). Преобладание отрицательных значений остатков в одних областях предсказанных значений  $\hat{y}$  и положительных остатков в других областях  $\hat{y}$  говорит о наличии систематической ошибки в данных или о том, что существенная независимая переменная в модели отсутствует.

На ДР, как и на любом графике, отображающем остатки, можно идентифицировать аномальные, резко выделяющиеся наблюдения как точки, лежащие вне диапазона, которому принадлежит большинство остатков.

**Однородность (постоянства) дисперсии** устанавливается с помощью ДР для остатков или их квадратов и предсказанных значений  $\hat{y}$  результирующей переменной:  $e = f(\hat{y})$  или  $e^2 = f(\hat{y})$ . Наличие резко выделяющихся точек на графике свидетельствует о нарушении постоянства дисперсии и требует выполнения преобразований, стабилизирующих дисперсию, или использования взвешенного метода наименьших квадратов для получения оценок параметров регрессии.

**Для проверки отсутствия зависимости между ошибкой и каждой из предикторных переменных**  $x_i$  используются диаграммы рассеивания вида  $e = f(x_i)$ . Можно указать четыре типа функциональной формы зависимости, которым соответствуют следующие четыре вида графиков: а) облако, б) полоса, в) кривая, г) веер.

График (рис. 3.11) свидетельствует об отсутствии связи между остатками  $e$  и  $x_i$ , т. е. о правильном выборе переменной  $x_i$  в качестве предикторной, объясняющей.

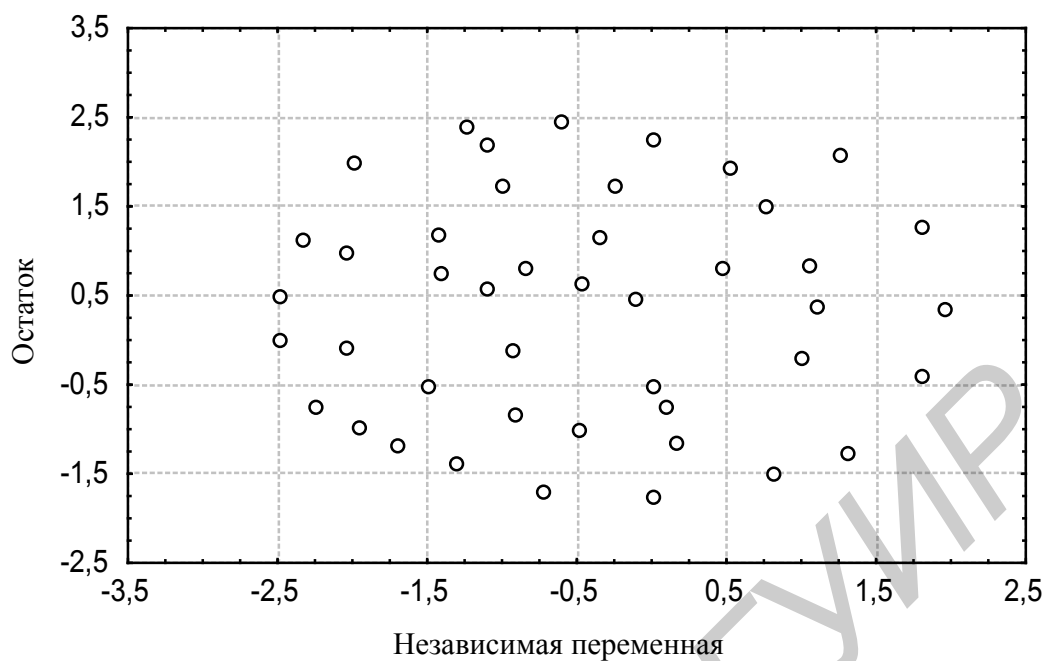


Рис. 3.11. Диаграмма рассеивания остатков и предикторной переменной  $x$  (связь между остатками и переменной  $x$  отсутствует)

График (рис. 3.12) указывает на присутствие выбросов в остатках. Это означает, что остатки не будут иметь нулевого среднего.

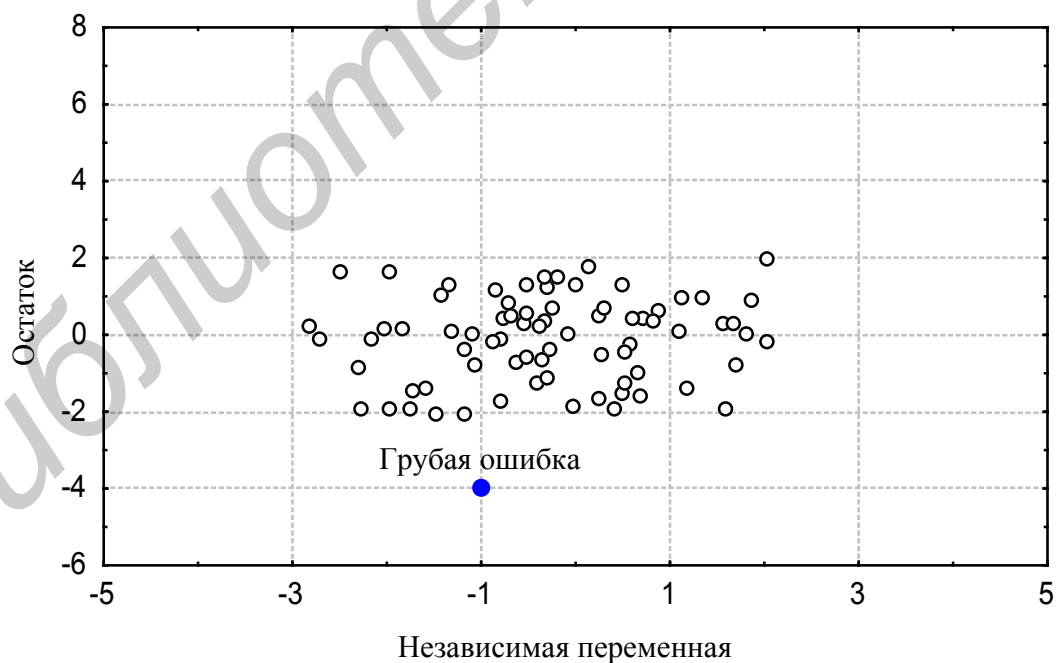


Рис. 3.12. Диаграмма рассеивания остатков предикторной переменной  $x$  при наличии выбросов в остатках

График (см. рис. 3.13) указывает на нелинейную зависимость между остатками  $e$  и  $x_i$ . В этом случае рекомендуется ввести в модель квадратичный член или преобразовать исходные данные для получения линейной зависимости.



Рис. 3.13. Диаграмма рассеивания при нелинейной зависимости остатков от предикторной переменной

График (рис. 3.14) показывает увеличение разброса остатков, связанного с  $x_i$ . Подобный «веерный» эффект отражает непостоянство дисперсии наблюдений. В этом случае рекомендуется преобразование зависимой переменной  $y$  или независимой переменной  $x_i$ , обеспечивающее постоянство дисперсии. Можно также использовать взвешенный метод наименьших квадратов для получения оценок параметров модели.

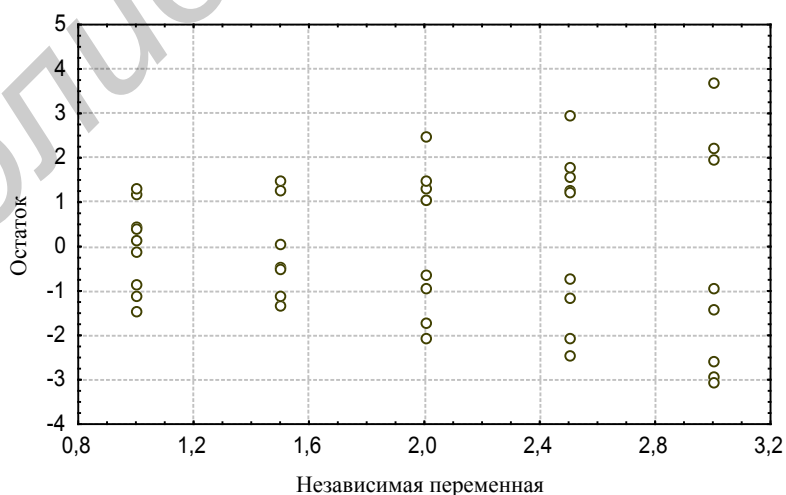


Рис. 3.14. Диаграмма рассеивания остатков и предикторной переменной  $x$  в случае непостоянства дисперсии наблюдения

## Выводы

Для проверки выполнения допущений модели регрессии используется графический анализ остатков, представляющий собой диагностический инструмент установления адекватности модели. Для этого строятся диаграммы рассеивания остатков в зависимости от предсказываемой переменной, независимой переменной, остатка, полученного в предыдущий момент времени. ДР остатков дают возможность:

- установить характерные особенности распределения остатков, свидетельствующие об отклонении от нормальности (наличие асимметрии) или значимого отличия от нуля математического ожидания остатков;
- выявить наличие систематической ошибки в данных или отсутствие существенной независимой переменной в модели;
- идентифицировать аномальные, резко выделяющиеся наблюдения;
- исследовать однородность дисперсии;
- подтвердить линейность или нелинейность регрессии.

### 3.2.2. Диаграммы рассеивания для поиска наилучшего разделения классов в дискриминантном анализе

Для представления структуры классов в дискриминантном анализе используются, как правило, матричные диаграммы рассеивания (см. рис. 3.6). Они иллюстрируют разбиение объектов на классы, описывая каждый объект только двумя переменными. Когда размерность вектора наблюдений  $p > 2$ , графическая интерпретация трудновыполнима. В этом случае взаимное пространственное расположение групп объектов можно изучить, получив отображения на канонические направления (или канонические оси), обеспечивающие наилучшее разделение классов.

На канонических осях откладываются значения канонических дискриминантных функций, являющихся линейной комбинацией  $p$  исходных переменных:

$$f_{mk} = u_0 + u_1 x_{1mk} + u_2 x_{2mk} + \dots + u_p x_{pmk},$$

где  $f_{mk}$  – значение канонической дискриминантной функции для  $k$ -го объекта в классе  $m$ ;  $x_{imk}$  – значение  $i$ -й переменной для  $k$ -го объекта в классе  $m$ ;  $u_i$  – коэффициенты канонической функции.

Приведенное выше соотношение задает математическое преобразование  $p$ -мерного пространства исходных переменных в  $q$ -мерное пространство канонических дискриминантных функций. Для данного наблюдения значение  $f_{mk}$  интерпретируется как координата объекта в пространстве канонических дискриминантных функций. Переход к каноническим переменным позволяет снизить размерность и упростить представление многомерных данных.

Диаграмма рассеивания в координатных осях, соответствующих каноническим переменным, является моделью носителя точек как линейного многообразия более низкой размерности, чем исходное.

На рис. 3.15 приведена диаграмма рассеивания двух первых канонических дискриминантных переменных, показывающая структуру классов для данных об ирисах. Четырехмерные исходные точки спроецированы в двумерное пространство таким образом, чтобы обеспечить максимальную разделяемость классов. Результаты отображения на рис. 3.15 можно сопоставить с матричной диаграммой рассеивания (см. рис. 3.6), с помощью которой решалась та же самая задача.

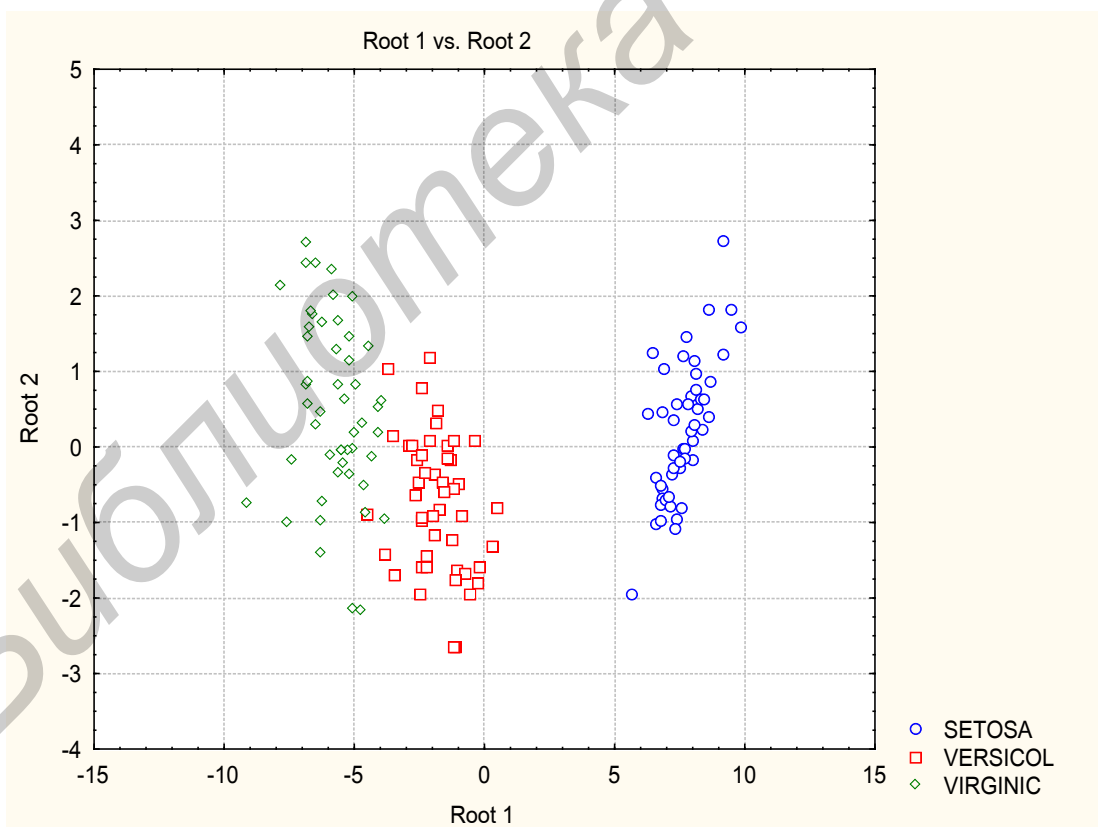


Рис. 3.15. Классы объектов, характеризуемых четырехмерным вектором наблюдений

### 3.2.3. Диаграммы рассеивания в кластерном анализе

Рассмотрим способы визуализации структур, возникающих в процедурах кластерного анализа. Для изучения взаимного пространственного расположения групп объектов может быть использована структурная модель в виде совокупности нескольких «облаков» точек, достаточно далеко отстоящих друг от друга. На основе заданного расстояния между объектами можно установить, какие множества называются группами однородных объектов, или классами. Если использовать в качестве меры близости между объектами, представляемыми  $p$ -мерными точками, евклидово расстояние, то с помощью диаграммы рассеивания любых переменных можно отобразить конфигурацию классов и выдвинуть предположение о типе классов. Выделяются следующие типы классов [3].

1. **Класс типа ядра.** Такой класс иногда называют сгущением. Все расстояния между объектами внутри класса меньше любого из расстояний между объектами класса и остальной частью множества объектов. На рис. 3.16 сгущениями являются  $A$  и  $B$ . Остальные пары множеств не разделяются с помощью этого определения.

2. **Кластер (сгущение в среднем).** Среднее расстояние внутри класса меньше расстояния объектов класса до всех остальных. Множества  $C$  и  $D$  теперь разделяются, но у  $E$  ( $G$ ) среднее внутреннее расстояние, больше чем  $E$  и  $F$  ( $G$  и  $H$ ).

3. **Класс типа ленты (слабое сгущение).** Существует  $\tau > 0$ , такое, что для любого  $x_i$  из класса  $S$  найдется такой объект  $x_j \in S$ , что расстояние  $d_{ij} \leq \tau$ , а для всех  $x_k \notin S$  справедливо неравенство  $d_{ik} > \tau$ . В смысле этого определения на рис. 3.16 разделяются все пары множеств, кроме  $I$  и  $J$ ,  $K$  и  $L$ .

4. **Класс с центром.** Существует порог  $R > 0$  и некоторая точка  $x^*$  в пространстве, занимаемом объектами класса  $S$ , такие, что все объекты из  $S$  и только они содержатся в шаре радиусом  $R$  с центром в  $x^*$ . Часто в качестве  $x^*$  выступает центр масс класса  $S$ , т. е. координаты центра определяются как средние значения признаков у объектов класса. Множества  $I$  и  $J$  являются классами с центром, а  $E$ ,  $F$  и  $G$  – нет.



## Выводы

Изучение взаимного пространственного расположения групп объектов и выделение любой структуры в дискриминантном анализе может быть выполнено с помощью диаграмм рассеивания.

Эффективным способом получения отображения структуры классов в дискриминантном анализе помимо диаграмм рассеивания является проецирование на канонические направления, что позволяет уменьшить размерность информации и выбрать наилучшие разделяющие поверхности.

Для выделения и изучения типов кластеров с целью получения содержательной трактовки задачи, обоснования выбора метода и алгоритмов классификации применяются диаграммы рассеивания.

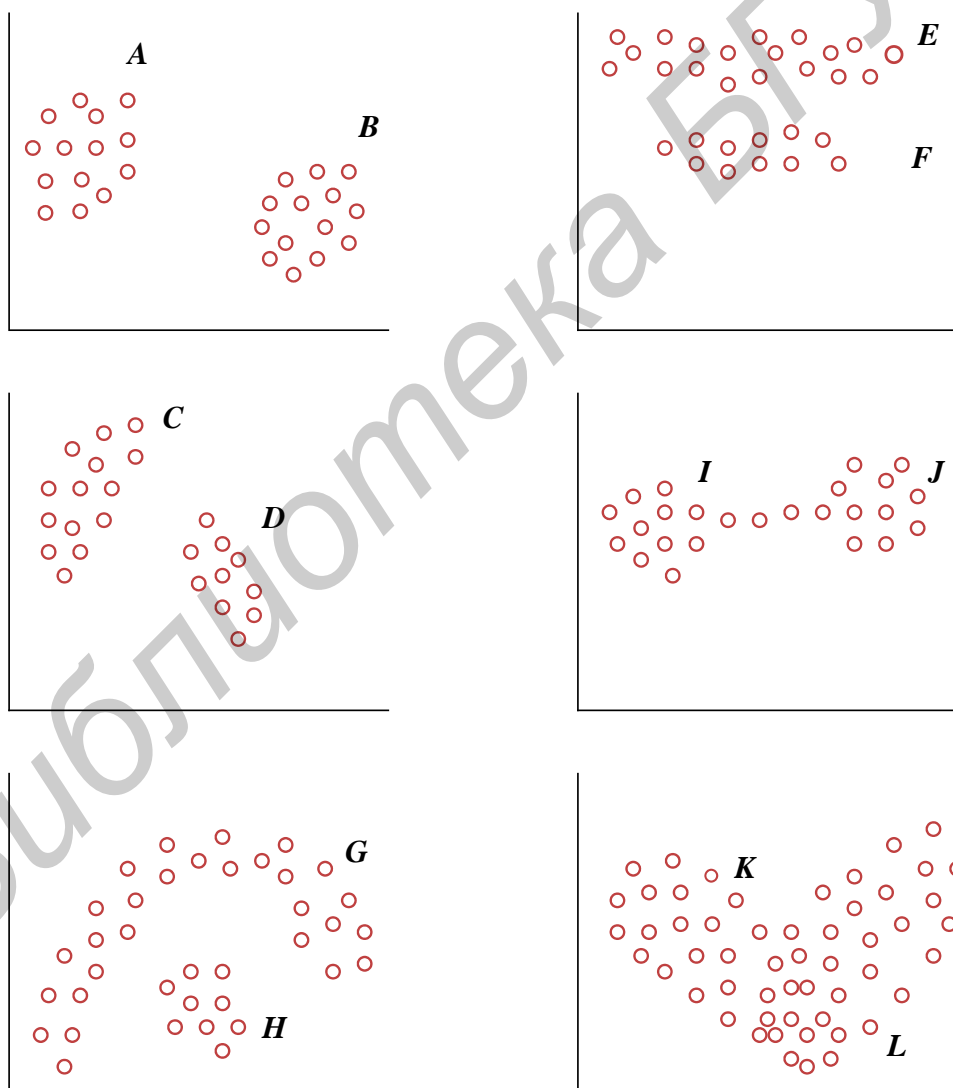


Рис. 3.16. Типы классов, отображаемых диаграммой рассеивания

### 3.3. Лица Чернова

Графики, рассмотренные в предыдущих разделах, являются средствами числового отображения данных: каждое наблюдение изображалось в виде точки в двухмерной или трехмерной системе координат. Координаты точки были равны числовым значениям двух или трех признаков, характеризующих объект. Поэтому числовое отображение структуры данных с использованием координат не может быть использовано для размерности пространства признаков, превышающей три.

Если мы заинтересованы в сжатом представлении структуры данных в виде двухмерных элементов, необходимо использовать иные графические методы, например, визуализацию с помощью пиктографиков.

На статистических пиктографиках наблюдения (объекты) представлены в виде графических символов со многими элементами. На каждом таком графическом объекте значениям переменных соответствуют определенные свойства или размеры этих объектов (как правило, одно наблюдение изображает один объект). Это соответствие таково, что внешний вид объекта изменяется в зависимости от набора значений признаков. Изучение пиктограмм помогает обнаружить характерные зависимости или группы наблюдений и исследовать взаимосвязи между несколькими переменными.

**Лица Чернова.** Вектор  $X_j$  из многомерного пространства признаков представляется на дисплее отдельным рисунком человеческого лица<sup>1</sup>. Черты лица определяются компонентами  $X_1, X_2, \dots, X_p$ . Например, признак  $X_1$  задает длину глаз,  $X_2$  – раскрытие глаз,  $X_3$  – положение носа и т. д. Относительная величина признака  $X_i$  связана с формой и размером индивидуальной характеристики лица (нос, рот, глаза, брови, уши и т. д.). Таким образом, синтезируемое лицо является результатом применения многомерного кодирования, поскольку кодовый алфавит включает в себя как линейные размеры, так и кривизну и углы наклона линий. В системе STATISTICA© для построения лица используется до 20 характеристик, т. е. можно без преобразований визуализировать исходные данные, имеющие размерность не более 20 (рис. 3.17).

---

<sup>1</sup>Chernoff H. The use of faces to represent points in  $k$ -dimensional space graphically//Journal of American Statistical Association. – 1973. – № 68. –С. 361–388.

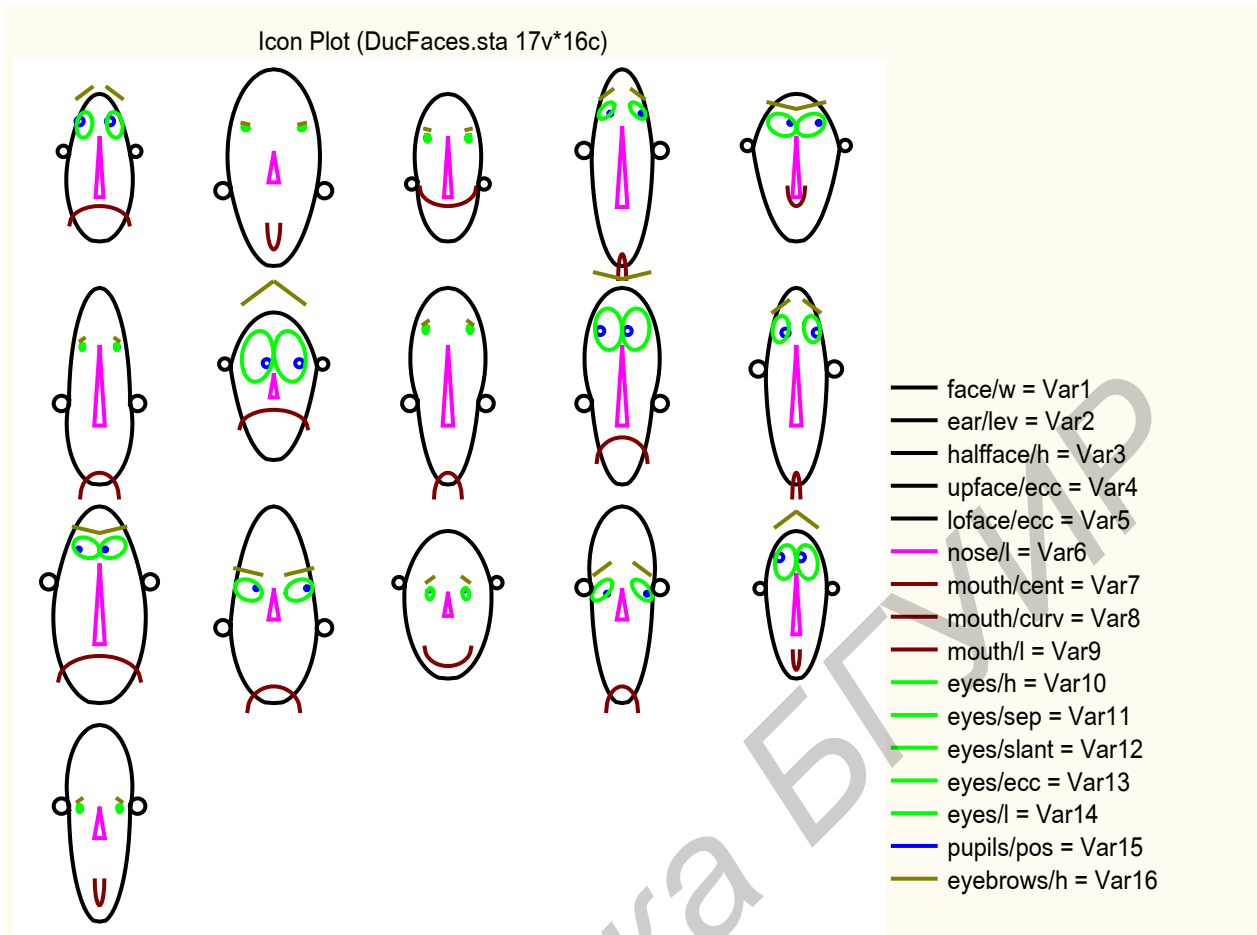


Рис. 3.17. Лица Чернова (16 признаков, 16 наблюдений)

На рис. 3.17 дается графическая иллюстрация 16 наблюдений вектора  $X$ , имеющего  $p = 16$  компонент:  $X = (x_1, x_2, \dots, x_{16})$ . Каждому наблюдению (объекту) соответствует свое лицо, а каждой переменной – компоненте  $x_i$  ( $i = 1, 2, \dots, n$ ) вектора  $X$  – определенная характеристика лица:  $x_1$  – ширина лица,  $x_2$  – уровень ушей,  $x_3$  – половина высоты лица,  $x_4$  – эксцентриситет верхней половины лица,  $x_5$  – эксцентриситет нижней половины лица,  $x_6$  – длина носа,  $x_7$  – положение центра рта,  $x_8$  – кривизна рта,  $x_9$  – длина рта,  $x_{10}$  – высота центра глаз,  $x_{11}$  – расстояние между глазами,  $x_{12}$  – наклон глаз,  $x_{13}$  – эксцентриситет глаз,  $x_{14}$  – половина длины глаз,  $x_{15}$  – положение зрачков,  $x_{16}$  – высота бровей,  $x_{17}$  – угол наклона бровей,  $x_{18}$  – длина брови,  $x_{19}$  – радиус уха,  $x_{20}$  – ширина носа.

Лица Чернова могут быть полезны не только для представления многомерных данных, но и для поиска однородных групп данных при классификации, при решении задач сравнения или изучения парных наблюдений, а также при идентифика-

ции аномальных наблюдений. Для этой цели множество лиц, характеризующих каждое наблюдение, дополняется новыми лицами-эталоном, с которыми производится сравнение. Новое лицо может быть построено как симметричное лицо для каждой пары наблюдений (двух объектов) или для одного изучаемого объекта.

В процессе анализа лиц Чернова производится поиск любых возможных закономерностей, таких, как сходство между группами изображений, выбросы или специфические соотношения между элементами лиц. Обнаруженные закономерности описываются в терминах используемых переменных. Для проверки найденной структуры соотношений переменные сопоставляются с другими элементами изображений. Например, можно попытаться переместить лица, чтобы упростить дальнейшее сравнение. В некоторых случаях рекомендуется исключить из рассмотрения переменные, не вносящие заметного вклада в исследуемую структуру.

Для проверки и количественной оценки обнаруженной зависимости или хотя бы некоторых ее параметров используется, например, регрессионный анализ, нелинейное оценивание, дискриминантный или кластерный анализ.

Рассмотрим пример использования лиц Чернова для представления образцов эфирного масла лимона<sup>1</sup>.

Лица Чернова позволяют обосновать принятие решения о принадлежности объекта какому-либо классу эквивалентности, получая ответы на вопросы: «Что общего у данного объекта с другим объектом или группой объектов (визуально ближайших или удаленных) с известной классификацией?», «Чем отличается данный объект от другого объекта или группы объектов с известной классификацией?» и т. п.

Были проанализированы хроматографические образцы эфирного масла лимона: L3 – фальсификат лимонного масла на основе терпеновой фракции; L2, L4, L5, L7 – образцы натурального эфирного масла лимона разного происхождения; IRIS, HADEK – образцы эфирного масла лимона различных фирм. Для сравнения были взяты усредненные хроматографические данные, приведенные в ISO 855:2003(E):

ISO-COAST – американский тип (прибрежный, влажный);

ISO-DES – американский тип (сухой, пустынный);

ISO-SPAIN – испанский тип;

---

<sup>1</sup>[viness.narod.ru/lemon\\_analysis.htm](http://viness.narod.ru/lemon_analysis.htm)

ISO-ITAL – итальянский тип;

ISO-IVORY – центрально-африканский тип.

Для портретного представления данных хроматографического анализа можно использовать лица Чернова. Метод основан на построении с помощью компьютерной программы черт человеческого лица из данных хроматографического анализа, т.е. содержания отдельных компонентов. Узнаваемость и похожесть лица и его сходство со стандартом позволяют легко сравнивать различные эфирные масла после проведения хроматографического анализа.

Оценка мимики лиц, характеризующих отдельные образцы эфирных масел, позволяет визуально сравнивать их качество. На рис. 3.18 представлены результаты хроматографического анализа образцов эфирного масла в виде лиц Чернова. Образец L3 (фальсификат) полностью «потерял лицо», несколько ушел от всех образцов L7 (африканский тип, отличный от европейского и американского). Образец L2 очень близок к стандарту ISO, образцы L4 и L5 очень сходны и близки к стандарту. Образец IRIS близок к испанскому типу (SPAIN) или итальянскому (ITAL), а HADEK, возможно, имеет американское (DES) происхождение.

Полезную информацию о близости объектов и наличии однородных групп в матрице исходных данных или матрице расстояний удастся извлечь с помощью методов иерархической классификации. Анализ выделенных группировок на каждом шаге алгоритма дает возможность установить, что общего (различного) имеется в каждой группировке объектов.

Для выявления сходства и различий образцов эфирного масла лимона наряду с лицами Чернова используем кластерный анализ, позволяющий группировать результаты хроматографического анализа в графический образ – дерево классификации.

На рис. 3.18 представлена кластерная диаграмма образцов эфирного масла лимона различного качества.

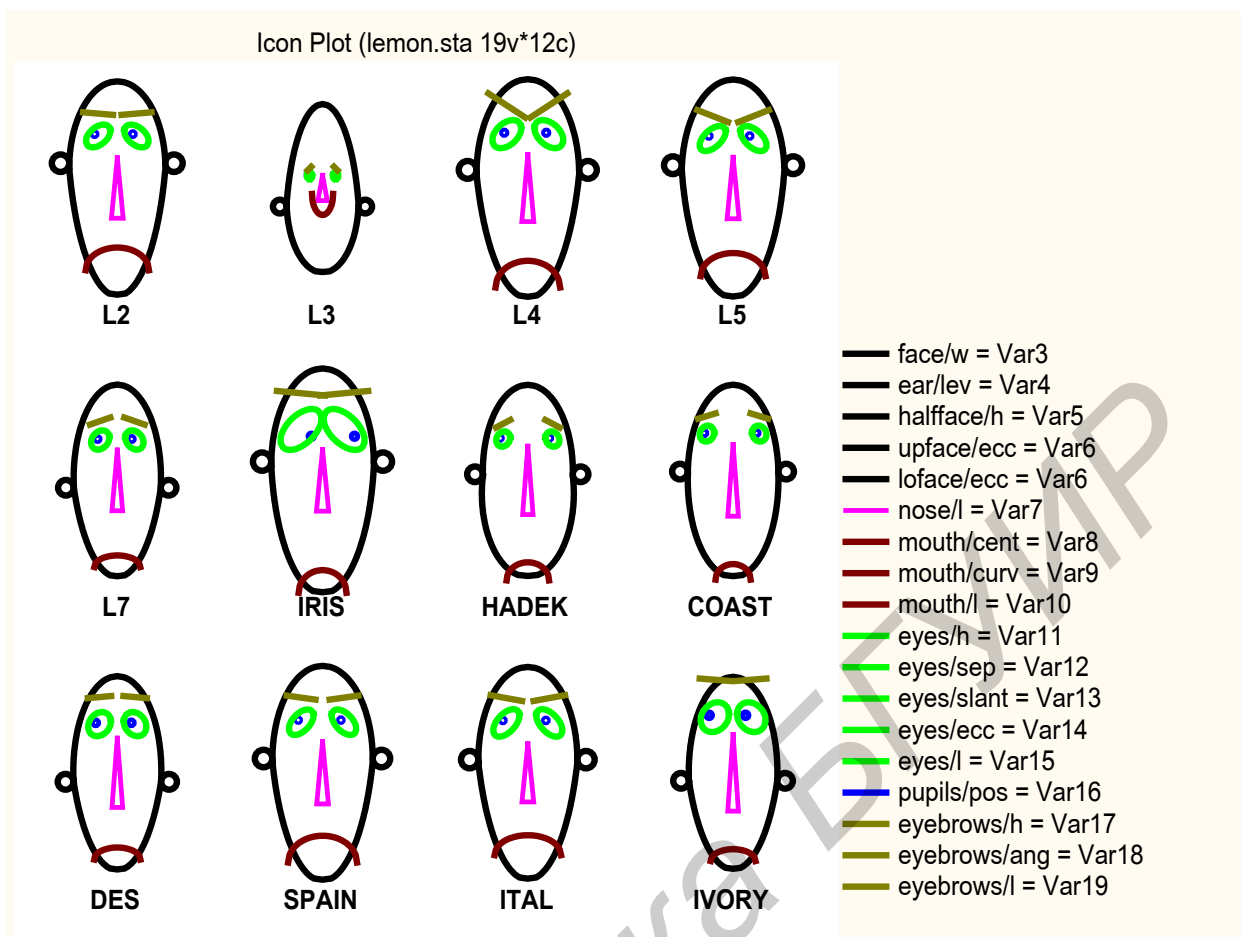


Рис. 3.18. Результаты хроматографического анализа образцов эфирного масла в виде лиц Чернова

Из дерева классификации видно, что образец L3 лежит очень далеко от кластера качественных эфирных масел лимона. Образцы L2 и L5 очень близки между собой и вместе с образцом L4 образуют кластер эфирных масел, сходных с лимонным маслом испанского происхождения, менее сходных с маслом итальянского происхождения и менее всего похожих на масло лимона американского типа, выращенного на океанском побережье. Образец L7 близок по качеству к маслу африканского экваториального происхождения (Берег Слоновой Кости – IVORY) и менее близок к маслу лимона американского типа (DES), выращенного в сухих условиях, и вряд ли получен в Испании (SPAIN) или Италии (ITAL).

Создана матрица расстояний euclid.smx. Исходные данные в файле lemon.sta (faces).

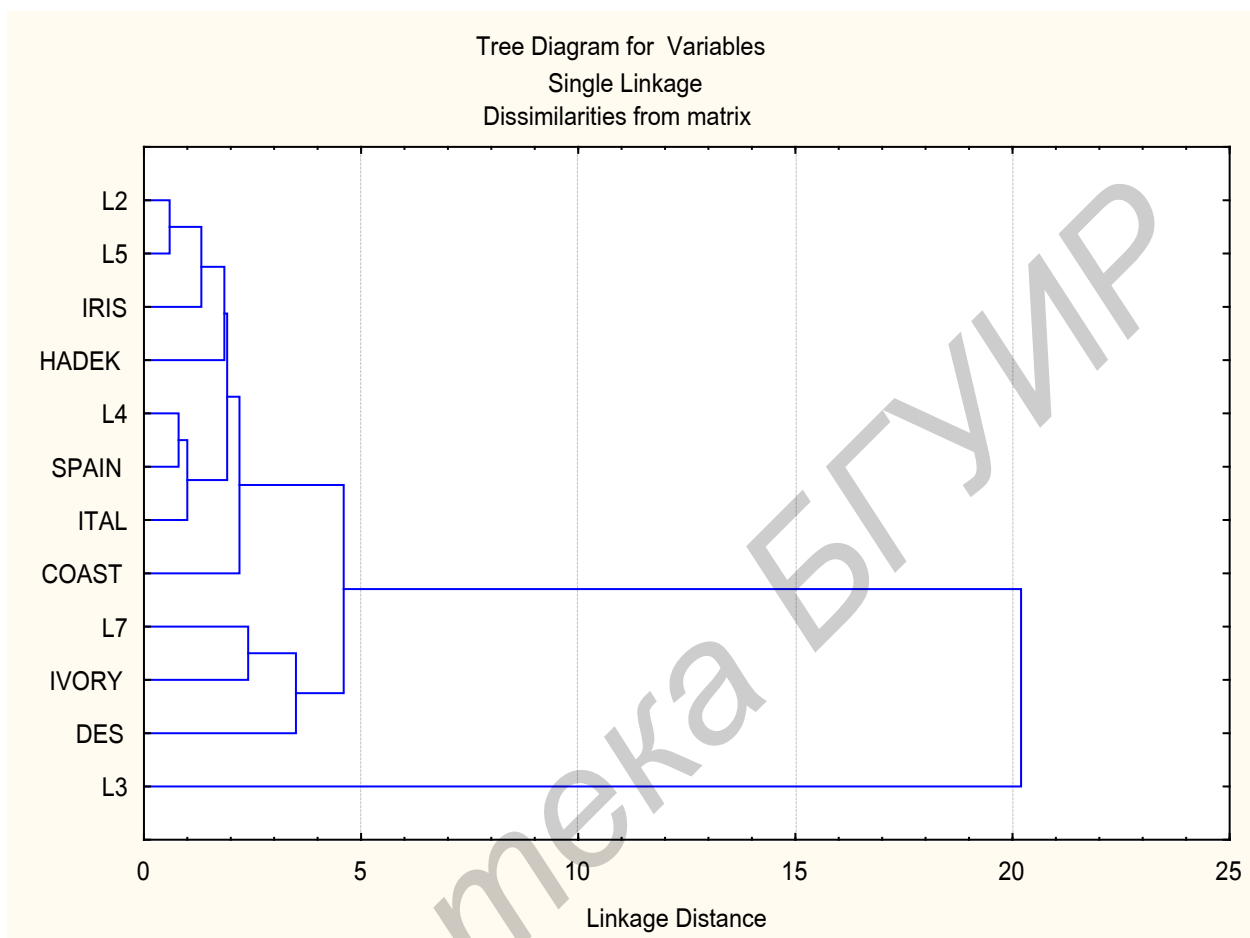


Рис. 3.19. Дерево классификации результатов хроматографического анализа образцов эфирного масла

Таблица 3.1

## Основные виды графиков, используемых в анализе данных

Назначение	Тип графика
1	2
<b>Одномерные данные</b>	
Представление выборки	Гистограмма Висячая гистограмма Полигон накопленных частот Эмпирическая функция распределения Стебель-и-листья «Ящик с усами» Столбиковые диаграммы Секторные диаграммы Линейные графики (переменные) Профили (реализации) Графики для идентификации выбросов Графики для представления размаха
Сопоставление характеристик по группам данных	Столбиковые диаграммы на плоскости Секторные диаграммы «Ящик с усами» Линейные графики (переменные) Профили (реализации) Графики для идентификации выбросов Графики для представления размаха Графики нормального распределения Графики квантиль–квантиль Графики вероятность–вероятность



1	2	
Подбор модели данных	Графики нормального распределения Графики квантиль–квантиль Графики вероятность–вероятность Графики функций распределения Графики плотностей распределения Графики, иллюстрирующие проверку гипотезы согласия	
<b>Двухмерные данные</b>		
Представление выборки	Столбиковые диаграммы Секторные диаграммы Диаграммы разброса (скаттерограммы)	
Сопоставление характеристик по группам данных	Двухмерные гистограммы Столбиковые диаграммы Секторные диаграммы Графики для представления размаха	
Подбор модели данных	Графики двухмерных плотностей распределения Поверхности Контуры, следы	
<b>Многомерные данные</b>		
Представление выборки	Лучи Звезды Проекции Эндрюса Секторные диаграммы	Профили Лица Чернова Столбиковые диаграммы
Исследование зависимостей	Матрица корреляций Трехмерные диаграммы рассеивания	
<b>Методы анализа данных</b>		
Проверка гипотез	Графики области принятия (отклонения) гипотезы Доверительный интервал	

1	2
Регрессионный анализ	<p>Графики для проверки допущений модели</p> <p>Диагностические графики анализа остатков (линейная регрессия)</p> <p>Графики для выбора наилучшего подмножества предикторов (линейная регрессия)</p> <p>Доверительный интервал для уравнения регрессии</p> <p>Эллипс рассеяния</p>
Анализ временных рядов	<p>Визуализация временного ряда (ВР)</p> <p>График тренда</p> <p>Графики компонент ВР</p> <p>График автокорреляционной функции</p> <p>График взаимной корреляционной функции</p>
Кластерный анализ	<p>Графики, иллюстрирующие процесс классификации (древовидные схемы, контуры, положение центров классов, расстояние объединения в зависимости от номера шага)</p> <p>Диаграммы рассеяния для выявления групп однородных данных и формы кластеров</p>
Факторный анализ	<p>Графики нагрузок в факторном пространстве</p> <p>График собственных значений матрицы корреляции</p>

1	2
Деревья классификации	<p>Дерево классификации</p> <p>Контурный график зависимости предсказанного номера класса от наблюдаемого номера</p> <p>График стоимость потерь – сложность дерева</p>
Дискриминантный анализ	<p>Графики для представления классов в пространстве канонических переменных</p> <p>Диаграммы рассеяния для выявления структуры обучающей выборки</p> <p>Диагностические графики для проверки выполнения допущений модели</p>
Таблицы выживаемости	<p>График функции риска</p> <p>График функции выживаемости</p> <p>График плотности распределения</p>
Дисперсионный анализ	<p>График средних значений</p> <p>Графики для проверки допущений модели</p>

## Литература

1. Анализ геоинформационных данных. Компьютерный практикум: учеб. пособие / М. Д. Степанова [и др.]; под науч. ред. В. В. Голенкова. – Минск : БГУИР, 2009.
2. Блинова, Т. А. Компьютерная графика : учеб. пособие / Т. А. Блинова, В. Н. Порев; под ред. В. Н. Порева. – Киев : Юниор, 2006. — 520 с.
3. Зенкин, А. А. Когнитивная компьютерная графика / А. А. Зенкин; под науч. ред. Д. А. Поспелова. – М.: Наука, 1991.
4. Лагутин, М. Б. Наглядная математическая статистика: учеб. пособие / М. Б. Лагутин.– М.: БИНОМ. Лаборатория знаний, 2007.
5. Прикладная статистика: Классификация и снижение размерности: справ. изд. / С. А. Айвазян [и др.]. – М.: Финансы и статистика, 1989.
6. Роджерс, Д. Алгоритмические основы машинной графики / Д. Роджерс : пер. с англ. – М. : Мир, 1989. – 512 с.
7. Роджерс, Д. Математические основы машинной графики / Д. Роджерс, Дж. Адамс : пер. с англ.– М. : Мир, 2001. – 604 с.
8. Статистические основы индуктивного вывода: учеб. пособие / М. Д. Степанова [и др.]; под науч. ред. В. В. Голенкова. – Минск : БГУИР, 2009.

*Учебное издание*

**Самодумкин Сергей Александрович**  
**Степанова Маргарита Дмитриевна**  
**Колб Дмитрий Григорьевич**

## ***ПРАКТИКУМ ПО КОМПЬЮТЕРНОЙ ГРАФИКЕ***

В 3-х частях

Часть 3

УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ

Редактор *Е. Н. Батурчик*  
Компьютерная правка, оригинал-макет *В. М. Задоя*

Подписано в печать . Формат 60x84 1/16. Бумага офсетная. Гарнитура «Таймс».  
Печать ризографическая. Усл. печ. л. . Уч.-изд. л. 5,0. Тираж 100 экз. Заказ 308.

Издатель и полиграфическое исполнение: учреждение образования  
«Белорусский государственный университет информатики и радиоэлектроники».  
Свидетельство о государственной регистрации издателя, изготовителя,  
распространителя печатных изданий №1/238 от 24.03.2014, №2/113 от 07.04.2014.  
ЛП №02330/264 от 14.04.2014.  
2200013, Минск, П. Бровки, 6