

Федеральное агентство по образованию  
Озерский технологический институт (филиал)  
ГОУ ВПО «Московский инженерно-физический институт  
(государственный университет)»

*Кафедра прикладной математики*

# Машинная графика

Учебное пособие

Озерск, 2006

УДК 681.3.06  
П 56

Пономарев В.В.  
Машинная графика. Учебное пособие. Ред. 2.  
Озерск: ОТИ МИФИ, 2006. — 72 с. ил.

Пособие предназначено для изучения дисциплины «Компьютерная графика» студентами-программистами. Во второй части пособия описаны математические основы машинной графики и базовые растровые алгоритмы. Изложение сопровождается множеством рисунков. К каждому разделу приведены контрольные вопросы и задания. Для получения более глубоких знаний рекомендуется изучение указанной в конце пособия литературы.

# Содержание

<b>Основы машинной графики .....</b>	<b>5</b>
Точки и линии на плоскости .....	6
Представление точек.....	6
Преобразование точек.....	6
Преобразование отрезка.....	8
Преобразование середины отрезка.....	8
Параллельные линии.....	9
Пересекающиеся отрезки.....	9
Вращение.....	10
Преобразование единичного квадрата .....	11
Матрица вращения .....	11
Аффинные преобразования .....	12
Однородные координаты .....	13
Точки в бесконечности .....	16
Контрольные вопросы и задания .....	17
Аффинные преобразования в пространстве.....	18
Трёхмерное вращение ( <i>rotation</i> ) .....	18
Растяжение-сжатие ( <i>dilatation</i> ) .....	20
Отражение ( <i>reflection</i> ) .....	20
Перенос ( <i>translation</i> ) .....	21
Пример аффинных преобразований в пространстве .....	21
Контрольные вопросы и задания .....	23
Проективные преобразования .....	24
Параллельные проекции.....	24
Ортография .....	25
Аксонметрические проекции.....	26
Прямоугольные проекции.....	26
Триметрия, диметрия и изометрия .....	28
Косоугольные проекции .....	31
Перспективные преобразования .....	32
Формирование картинки .....	37
Видовое преобразование .....	42
Простое перспективное преобразование .....	45
Автостабилизация размера картинки.....	48
Контрольные вопросы и задания .....	49
<b>Растровые алгоритмы .....</b>	<b>50</b>
Алгоритм Брезенхейма .....	50
Алгоритм отсечения отрезка (Сазерленда-Кохена).....	55
Определение принадлежности точки многоугольнику .....	57
Закраска области.....	59
Триангуляция полигона.....	61
Контрольные вопросы и задания .....	64
<b>Рекомендуемая литература .....</b>	<b>65</b>
<b>Приложения .....</b>	<b>66</b>
Таблицы матриц преобразования .....	66

Элементы матрицы преобразования $4 \times 4$ .....	66
Матрицы отражения, масштабирования и переноса .....	66
Матрицы вращения .....	67
Матрицы параллельных проекций .....	67
Ортогональные проекции .....	67
Прямоугольные проекции .....	68
Аксонметрические проекции .....	69
Косоугольные проекции .....	69
Перспективные проекции .....	69
Видовое преобразование .....	70
Простое перспективное преобразование .....	70
Формулы приведения углов .....	70
Матричные операции .....	71

## Основы машинной графики

Машинная графика — это часть компьютерной графики, занимающаяся математическими аспектами построения изображений. Предполагается, что сами изображения при этом имеют неизобразительную природу. К ним относятся, прежде всего, простые плоские графические примитивы, такие, как отрезки, дуги, окружности, а также более сложные, трехмерные каркасные объекты — сфера, куб, призма и т. п.

Для представления графических объектов используются различные математические модели, их описывающие. Так, для описания отрезка используется запись координат его концевых точек. Для описания круга достаточно записать координаты центральной точки и любой точки на окружности, которая определит радиус.

Для описания многогранных объектов типа куба, призмы используются *полигональные модели*, в которых объект описывается своими гранями. Каждая грань записывается своими вершинами, которые определяют ребра.

Для неграненых объектов, таких, как сфера и цилиндр, математическое описание может состоять из центральной точки и некоторых параметров — центр сферы и радиус, центр основания цилиндра, радиус и высота и т. д. Для изображения таких объектов используется аппроксимация их криволинейной поверхности четырех и треугольными гранями. Объект при этом становится полигональной моделью, которая рассчитывается непосредственно во время вывода, или предварительно, после задания параметров объекта.

Отдельно стоят в ряду графических объектов кривые линии и поверхности. Кривые линии задаются при помощи нескольких точек, а кривые поверхности — несколькими кривыми линиями. Для рисования кривых линий используются сложные алгоритмы, которые вычисляют отрезки прямых, из которых состоит кривая линия. Кривые поверхности выводятся в виде сетки, рисуемой отрезками прямых.

В целом можно сказать, что объекты задаются своими точками. Совокупность объектов для вывода называют *сценой*. В сцену могут также входить дополнительные объекты, такие, как источники света.

Для изменения сцены и ее отдельных объектов используются различные преобразования, — масштабирование, перемещение, вращение, — выполняемые при помощи матриц. Преобразования, с одной стороны, задаются *матрицами преобразования*. Точки объектов сцены, с другой стороны, образуют матрицу точек. Перемножение этих двух матриц дает матрицу преобразованных точек, которая используется для дальнейшего преобразования, или непосредственно для вывода на экран.

При выводе на экран используются различные способы изображения объектов. Так, при конструировании технических объектов используется ускоренный вывод объектов в виде каркасов, ребер и других упрощенных представлений. В других случаях требуется получение полноцветной картинкой для лучшего восприятия.

При программировании графических объектов и их преобразований используются различные структуры данных. Чаще всего для описания точек используются структуры, а для описания матриц — массивы. Используются также классы, описывающие векторы и матрицы. Так, для выполнения практических работ студенты ОТИ МИФИ используют набор классов (объединенных в DLL), имеющий название «ReVoL GREEN Interface».

## Точки и линии на плоскости

В компьютерной графике для преобразований геометрических объектов используются матричные преобразования. Необходимость самих преобразований вытекает из желания установить некоторую точку зрения на наблюдаемые графические объекты, изменить их вид или расположение в пространстве по отношению к исходной системе координат — растянуть, сдвинуть, повернуть, переместить.

### Представление точек

Точки представляются в виде *вектор-строки*, называемой *вектором положения*. Точка на плоскости описывается двумя координатами  $(x, y)$ , или двухмерной вектор-строкой  $|x \ y|$ . Точка в пространстве описывается тремя координатами  $(x, y, z)$  или трехмерной вектор-строкой  $|x \ y \ z|$ .

### Преобразование точек

Рассмотрим простейшие преобразования двухмерной точки  $P(x, y)$ . Будем умножать вектор положения этой точки на квадратную матрицу  $T$  размером 2:

$$|x \ y| \cdot \begin{vmatrix} a & b \\ c & d \end{vmatrix} = |(ax + cy) \ (bx + dy)| = |x^* \ y^*|$$

Начальные координаты  $x$  и  $y$  преобразуются в конечные координаты  $x^*$  и  $y^*$ , причем  $x^* = ax + cy$ ,  $y^* = bx + dy$ . Рассчитаем несколько частных случаев.

1. Нет преобразований, если матрица единичная:

$$|x \ y| \cdot \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix} = |x \ y| = |x^* \ y^*|$$

2. Изменение масштаба вдоль оси  $X$ , если  $b = c = 0$ ,  $d = 1$ ,  $a \neq 1$ . Точка растягивается, если  $a > 1$  и сжимается, если  $a < 1$ :

$$|x \ y| \cdot \begin{vmatrix} a & 0 \\ 0 & 1 \end{vmatrix} = |ax \ y| = |x^* \ y^*|$$

Преобразованные координаты:  $x^* = ax$ ,  $y^* = y$  (Рис. 1, слева).

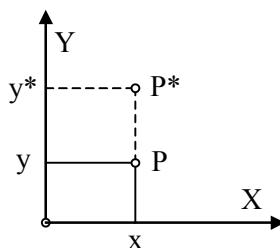
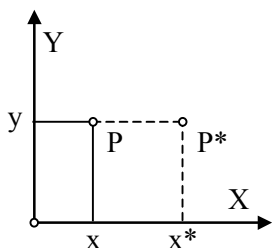


Рис. 1. Изменение масштаба вдоль оси  $X$  (слева) и вдоль оси  $Y$  (справа)

3. Изменение масштаба вдоль оси  $Y$ , если  $b = c = 0$ ,  $a = 1$ ,  $d \neq 1$ . Точка растягивается, если  $d > 1$  и сжимается, если  $d < 1$ :

$$|x \ y| \cdot \begin{vmatrix} 1 & 0 \\ 0 & d \end{vmatrix} = |x \ dy| = |x^* \ y^*|$$

Преобразованные координаты:  $x^* = x$ ,  $y^* = dy$  (Рис. 1, справа).

4. Отображение относительно оси X, если  $a = -1$ ,  $b = c = 0$ ,  $d = 1$ :

$$\begin{vmatrix} x & y \end{vmatrix} \cdot \begin{vmatrix} -1 & 0 \\ 0 & 1 \end{vmatrix} = \begin{vmatrix} -x & y \end{vmatrix} = \begin{vmatrix} x^* & y^* \end{vmatrix}$$

Преобразованные координаты:  $x^* = -x$ ,  $y^* = y$  (Рис. 2, слева).



**Рис. 2. Отображение относительно оси X (слева) и относительно оси Y (справа)**

5. Отображение относительно оси Y, если  $d = -1$ ,  $b = c = 0$ ,  $a = 1$ :

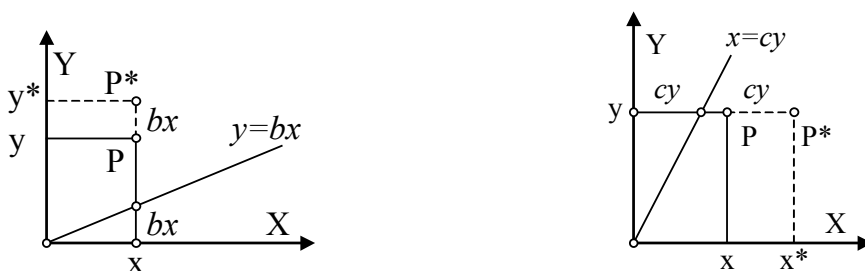
$$\begin{vmatrix} x & y \end{vmatrix} \cdot \begin{vmatrix} 1 & 0 \\ 0 & -1 \end{vmatrix} = \begin{vmatrix} x & -y \end{vmatrix} = \begin{vmatrix} x^* & y^* \end{vmatrix}$$

Преобразованные координаты:  $x^* = x$ ,  $y^* = -y$  (Рис. 2, справа).

6. Сдвиг вдоль оси Y, если  $b \neq 0$ ,  $a = d = 1$ ,  $c = 0$ :

$$\begin{vmatrix} x & y \end{vmatrix} \cdot \begin{vmatrix} 1 & b \\ 0 & 1 \end{vmatrix} = \begin{vmatrix} x & bx + y \end{vmatrix} = \begin{vmatrix} x^* & y^* \end{vmatrix}$$

Преобразованные координаты:  $x^* = x$ ,  $y^* = bx + y$  (Рис. 3, слева). Координата  $x$  точки не меняется, а координата  $y$  линейно зависит от  $x$ .



**Рис. 3. Сдвиг вдоль оси Y (слева) и вдоль оси X (справа)**

7. Сдвиг вдоль оси X, если  $c \neq 0$ ,  $a = d = 1$ ,  $b = 0$ :

$$\begin{vmatrix} x & y \end{vmatrix} \cdot \begin{vmatrix} 1 & 0 \\ c & 1 \end{vmatrix} = \begin{vmatrix} x + cy & y \end{vmatrix} = \begin{vmatrix} x^* & y^* \end{vmatrix}$$

Преобразованные координаты:  $x^* = x + cy$ ,  $y^* = y$  (Рис. 3, справа). Координата  $y$  точки не меняется, а координата  $x$  линейно зависит от  $y$ .

8. Применяя матрицу преобразования общего вида к началу координат, получаем ту же точку — начало координат инвариантно к преобразованию:

$$\begin{vmatrix} 0 & 0 \end{vmatrix} \cdot \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix} = \begin{vmatrix} 0 & 0 \end{vmatrix} = \begin{vmatrix} x^* & y^* \end{vmatrix}$$

**Выводы:** Элементы главной диагонали матрицы преобразования изменяют масштаб точки вдоль координатных осей или выполняют отображения. Элементы вспомогательной диагонали выполняют сдвиг точек вдоль координатных осей.

## Преобразование отрезка

Отрезок прямой линии задается двумя векторами положения, определяющими концевые точки. Здесь не рассматривается, каким образом отрезок может быть изображен на устройстве вывода, и все рассуждения касаются только математических операций с векторами положения.

Рассмотрим отрезок с концевыми точками  $A(0, 1)$  и  $B(2, 3)$ . Выполним преобразование концевых точек при помощи матрицы  $T$ , которая задает сдвиг:

$$T = \begin{vmatrix} 1 & 2 \\ 3 & 1 \end{vmatrix}.$$

Для точки  $A(0, 1)$  получим преобразованную точку  $A^*(3, 1)$ :

$$\begin{vmatrix} 0 & 1 \end{vmatrix} \cdot \begin{vmatrix} 1 & 2 \\ 3 & 1 \end{vmatrix} = \begin{vmatrix} 3 & 1 \end{vmatrix} = A^*.$$

Для точки  $B(2, 3)$  получим преобразованную точку  $B^*(11, 7)$ :

$$\begin{vmatrix} 2 & 3 \end{vmatrix} \cdot \begin{vmatrix} 1 & 2 \\ 3 & 1 \end{vmatrix} = \begin{vmatrix} 11 & 7 \end{vmatrix} = B^*.$$

Результат преобразования приведен на Рис. 4.

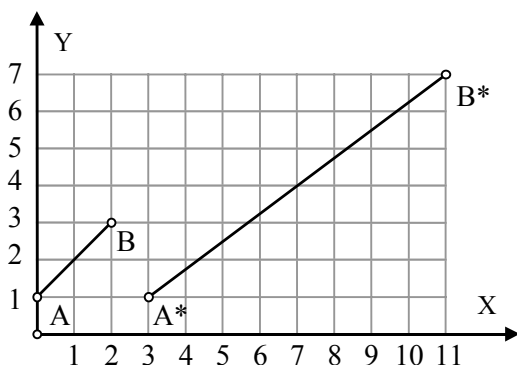


Рис. 4. Преобразование отрезка

Преобразование отрезка можно записать более компактно, если использовать для представления отрезка матрицу  $L$  размером  $2 \times 2$ . Каждая строка этой матрицы представляет вектор положения концевой точки:

$$L = \begin{vmatrix} 0 & 1 \\ 2 & 3 \end{vmatrix}.$$

Перемножая матрицы  $L$  и  $T$ , получаем новую матрицу  $L^*$ , которая содержит векторы положения концевых точек преобразованного отрезка:

$$L \cdot T = \begin{vmatrix} 0 & 1 \\ 2 & 3 \end{vmatrix} \cdot \begin{vmatrix} 1 & 2 \\ 3 & 1 \end{vmatrix} = \begin{vmatrix} 3 & 1 \\ 11 & 7 \end{vmatrix} = L^*.$$

Операция преобразования изменила длину отрезка и ее положение.

## Преобразование середины отрезка

Преобразование отрезка прямой линии приводит к получению нового отрезка. При этом обеспечивается однозначное соответствие между точками исходного и преобразованного отрезков. Это обстоятельство позволяет построить преобразованный отрезок простым соединением преобразованных концевых точек. Рассмотрим точку  $Q$  середины исходного отрезка. Ее координаты можно вычислить как половины сумм координат:



$$Q = \left| \frac{x_1 + x_2}{2} \quad \frac{y_1 + y_2}{2} \right|.$$

Запишем координаты преобразованных точек  $A^*$  и  $B^*$  в общем виде:

$$x_1^* = ax_1 + cy_1, \quad y_1^* = bx_1 + dy_1,$$

$$x_2^* = ax_2 + cy_2, \quad y_2^* = bx_2 + dy_2.$$

Найдем среднюю точку преобразованного отрезка через половины сумм координат:

$$Q^* = \left| \frac{ax_1 + cy_1 + ax_2 + cy_2}{2} \quad \frac{bx_1 + dy_1 + bx_2 + dy_2}{2} \right|.$$

Выполним теперь преобразование точки  $Q((x_1 + x_2)/2, (y_1 + y_2)/2)$ :

$$x^*_Q = \frac{ax_1 + ax_2 + cy_1 + cy_2}{2},$$

$$y^*_Q = \frac{bx_1 + bx_2 + dy_1 + dy_2}{2}.$$

Получили ту же самую точку. Преобразование серединной точки исходного отрезка дает серединную точку преобразованного отрезка (иначе говоря, середина преобразованного отрезка расположена на прямой, соединяющей преобразованные концевые точки отрезка).

## Параллельные линии

Преобразование пары параллельных отрезков приводит к получению другой пары параллельных отрезков. Чтобы убедиться в этом, рассмотрим наклон линии  $k$ , заданной точками  $(x_1, y_1)$  и  $(x_2, y_2)$ :

$$k = \frac{y_2 - y_1}{x_2 - x_1}.$$

Выполняя произвольное преобразование, получим:

$$k^* = \frac{(bx_2 + dy_2) - (bx_1 + dy_1)}{(ax_2 + cy_2) - (ax_1 + cy_1)} = \frac{b(x_2 - x_1) + d(y_2 - y_1)}{a(x_2 - x_1) + c(y_2 - y_1)},$$

или

$$k^* = \frac{b + d(y_2 - y_1)/(x_2 - x_1)}{a + c(y_2 - y_1)/(x_2 - x_1)} = \frac{b + dk}{a + ck}.$$

Поскольку результат преобразования не зависит от координат исходных точек отрезка, а зависит только от наклона исходной линии  $k$  и коэффициентов матрицы преобразования  $a, b, c$  и  $d$ , можно сделать вывод, что наклоны двух параллельных линий после преобразования будут одинаковыми, то есть линии будут также параллельными. Это, в свою очередь, означает, что произвольный параллелограмм преобразуется в другой параллелограмм в результате действия произвольной матрицы преобразования  $2 \times 2$ .

## Пересекающиеся отрезки

Покажем, что точка пересечения двух отрезков преобразуется в точку пересечения двух преобразованных отрезков.

На Рис. 5 показаны два пунктирных пересекающихся отрезка  $AB$  и  $CD$ . Точка пересечения имеет координаты  $(1, 0)$ . Умножив матрицы координат концевых

точек этих отрезков на матрицу преобразования  $a = 1, b = 1, c = 1, d = -1$ , получим два новых отрезка  $A^*B^*$  и  $C^*D^*$ , пересекающихся в точке  $(1, 1)$ :

$$\begin{vmatrix} -1 & 2 \\ 3 & -2 \end{vmatrix} \cdot \begin{vmatrix} 1 & 1 \\ 1 & -1 \end{vmatrix} = \begin{vmatrix} 1 & -3 \\ 1 & 5 \end{vmatrix};$$

$$\begin{vmatrix} -1 & -2 \\ 3 & 2 \end{vmatrix} \cdot \begin{vmatrix} 1 & 1 \\ 1 & -1 \end{vmatrix} = \begin{vmatrix} -3 & 1 \\ 5 & 1 \end{vmatrix}.$$

Умножая точку пересечения исходных отрезков на матрицу преобразования, получаем точку пересечения преобразованных отрезков:

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

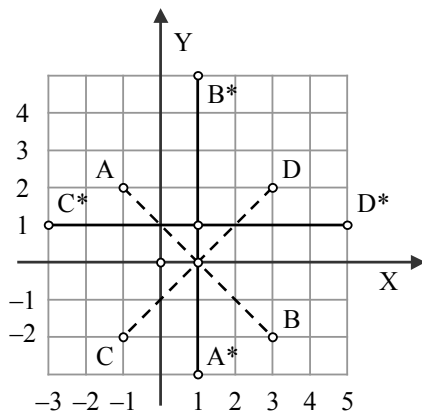


Рис. 5. Преобразование пересекающихся линий

## Вращение

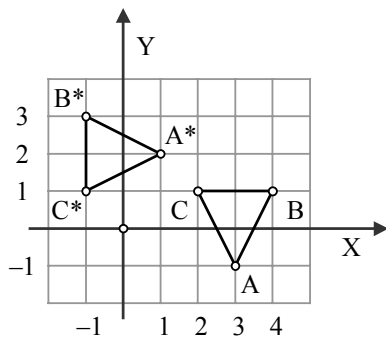


Рис. 6. Вращение

Рассмотрим плоский треугольник ABC (Рис. 6). Умножим матрицу координат вершин треугольника на матрицу  $T_{90} = (a = 0, b = 1, c = -1, d = 0)$ :

$$\begin{vmatrix} 3 & -1 \\ 4 & 1 \\ 2 & 1 \end{vmatrix} \cdot \begin{vmatrix} 0 & 1 \\ -1 & 0 \end{vmatrix} = \begin{vmatrix} 1 & 3 \\ -1 & 4 \\ -1 & 2 \end{vmatrix}.$$

Преобразованный треугольник  $A^*B^*C^*$  повернулся вокруг начала координат на угол  $90^\circ$ . Поворот вокруг начала координат на угол  $180^\circ$  получается при помощи матрицы

$$T_{180} = \begin{vmatrix} -1 & 0 \\ 0 & -1 \end{vmatrix},$$

а поворот на  $270^\circ$  — при помощи матрицы

$$T_{270} = \begin{vmatrix} 0 & -1 \\ 1 & 0 \end{vmatrix}.$$

## Преобразование единичного квадрата

Рассмотрим произвольное преобразование единичного квадрата, векторы положения вершин которого описываются матрицей

$$S = \begin{vmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{vmatrix} \begin{matrix} (A) \\ (B) \\ (C) \\ (D). \end{matrix}$$

Общее преобразование единичного квадрата имеет вид:

$$\begin{vmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{vmatrix} \cdot \begin{vmatrix} a & b \\ c & d \end{vmatrix} = \begin{vmatrix} 0 & 0 \\ a & b \\ a+c & b+d \\ c & d \end{vmatrix} \begin{matrix} (A^*) \\ (B^*) \\ (C^*) \\ (D^*). \end{matrix}$$

Первая строка (точка A) не подвергается преобразованию. Координаты точки B определяются первой, а координаты точки D — второй строкой матрицы преобразования. Учитывая, что фигура является квадратом, тогда если координаты точек B и D известны, то определена и вся фигура, так как параллельные линии преобразуются в параллельные, а вся фигура является параллелограммом. При этом элементы b и c матрицы вызывают сдвиг исходного квадрата в направлениях y и x соответственно, а элементы c и d характеризуют масштабные коэффициенты вдоль осей y и x (Рис. 7).

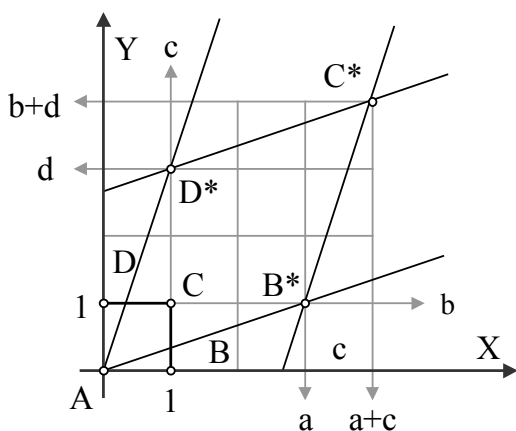


Рис. 7. Преобразование единичного квадрата

## Матрица вращения

Чтобы получить матрицу вращения, рассмотрим вращение единичного квадрата относительно начала координат (Рис. 8).

За *положительное направление вращения* принимается вращение против часовой стрелки, показанное на рисунке. Точка B с координатами (1, 0) преобразуется в точку B\* с координатами  $(1 \cdot \cos\theta, 1 \cdot \sin\theta)$ , а точка D с координатами (0, 1) преобразуется в точку D\* с координатами  $(-1 \cdot \sin\theta, 1 \cdot \cos\theta)$ .

Учитывая, что для единичного квадрата координаты точки B\* определяются первой строкой матрицы преобразования, а координаты точки D\* — второй

строкой, общую матрицу вращения размера  $2 \times 2$  можно записать следующим образом:

$$T = \begin{vmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{vmatrix}.$$

Для частного случая вращения на  $90^\circ$  получаем рассмотренную ранее матрицу поворота:

$$T_{90} = \begin{vmatrix} 0 & 1 \\ -1 & 0 \end{vmatrix}.$$

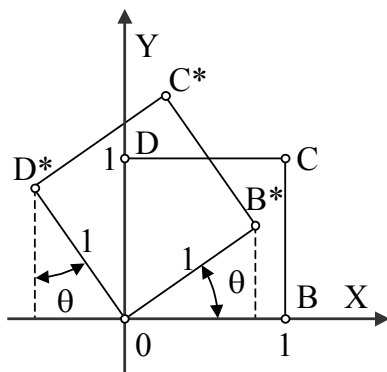


Рис. 8. Вращение единичного квадрата

## Аффинные преобразования

Аффинными преобразования ставят в соответствие одной прямолинейной системе координат другую прямолинейную систему координат. Они описываются следующими соотношениями:

$$\begin{aligned} x^* &= ax + by + m, \\ y^* &= cx + dy + n. \end{aligned}$$

где  $a, b, c, d$  — произвольные числа, связанные неравенством

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} \neq 0.$$

Рассмотренные выше преобразования точки являются аффинными. Среди них можно выделить четыре частных случая, имеющих хорошо прослеживаемую геометрическую интерпретацию:

A. Поворот точки вокруг начала координат на угол  $\theta$ :

$$\begin{aligned} x^* &= x \cos \theta + y \sin \theta, \\ y^* &= -x \sin \theta + y \cos \theta. \end{aligned}$$

B. Растяжение (сжатие) точки вдоль координатных осей:

$$\begin{aligned} x^* &= ax, \\ y^* &= cy. \end{aligned}$$

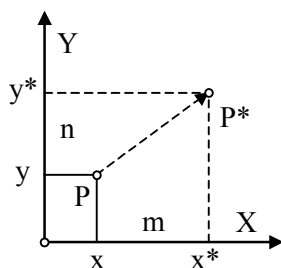
C. Отражение точки относительно оси абсцисс:

$$\begin{aligned} x^* &= x, \\ y^* &= -y. \end{aligned}$$

D. Перенос точки на вектор переноса  $(m, n)$ :

$$\begin{aligned} x^* &= x + m, \\ y^* &= y + n. \end{aligned}$$

Последнее преобразование показано на Рис. 9.



**Рис. 9. Перенос точки на вектор (m, n)**

Данные преобразования являются базовыми преобразованиями в компьютерной графике по следующим причинам:

1. Каждое из преобразований A, B, C и D имеет простой и наглядный геометрический смысл.
2. Произвольное аффинное преобразование может быть описано как последовательность (суперпозиция) базовых.

Первые три преобразования описываются матричным способом (при помощи соответствующих матриц), что было показано выше. Для однородности вычислений хотелось бы, чтобы перенос также описывался матричным преобразованием. Это можно сделать, если расширить матрицу преобразования до размера  $3 \times 2$ :

$$T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ m & n \end{bmatrix}.$$

Для матрицы переноса нельзя получить обратную, так как она не квадратная. Можно еще расширить матрицу переноса на один столбец следующим образом:

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ m & n & 1 \end{bmatrix}.$$

Тогда для выполнения преобразований в вектор положения точки следует ввести дополнительную третью координату, равную единице:

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ m & n & 1 \end{bmatrix} = \begin{bmatrix} x+m & y+n & 1 \end{bmatrix}.$$

## Однородные координаты

*Однородными* (или *пропорциональными*) называют координаты, которые задают положение точки относительно дополнительной координаты, обозначаемой  $h$  (*homogeneous*). Так, в двухмерном случае, координаты точки  $(x, y)$  преобразуются в однородные умножением на дополнительную координату  $h$ :

$$(X, Y, H) = (x \cdot h, y \cdot h, h).$$

Рассмотрим сначала одномерный случай. Одна координата задает положение точки на оси  $X$ . Вводя дополнительную координату  $H$ , получим вектор положения точки  $(x \cdot h, h)$ , где  $h$  — произвольное значение. Теперь положение точки задается не на оси  $X$ , а на прямой, выходящей из начала координат и проходящей через точку  $P=(x, 1)$ :

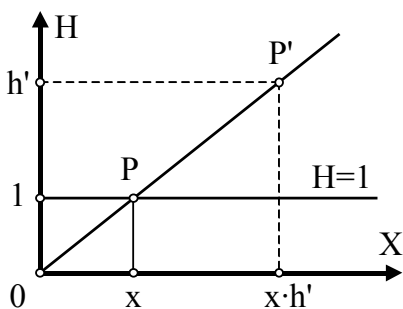


Рис. 10. Одномерные однородные координаты

Точка  $P=(x, 1)$  является исходной точкой, а прямая  $H=1$  — геометрическим местом нормальных координат (Рис. 10). Прямая  $OP$  задает неоднородные координаты точки  $(x, 1)$ . Любая произвольная точка на этой прямой, например, точка  $P'$ , задает одну и ту же нормальную точку  $(x, 1)$ . Нормальные координаты точки могут быть получены из однородных делением всех координат на однородную  $h$ :  $(x, 1) = (x \cdot h/h, h/h)$  (это следует из подобия треугольников). Процесс получения нормальных координат точки называется *нормализацией*.

Однородные координаты задают точку не своими абсолютными значениями, а отношением координат. *Не существует единственного однородного представления точки*. Так, однородные координаты  $(4, 2)$ ,  $(6, 3)$  и  $(8, 4)$  задают одну и ту же нормальную точку  $(2, 1)$ . Нормальные координаты лежат на пересечении прямой, задающей однородную точку, с прямой  $H = 1$ .

Рассмотрим преобразования в одномерном пространстве при помощи однородных координат. Матрица преобразования имеет размерность  $2 \times 2$ :

$$\begin{vmatrix} x & 1 \end{vmatrix} \cdot \begin{vmatrix} a & p \\ m & s \end{vmatrix} = \begin{vmatrix} (ax + m) & (px + s) \end{vmatrix} = \begin{vmatrix} X & H \end{vmatrix}$$

Нормальная координата  $x$  преобразуется в однородную координату  $X=ax+m$ , а нормальная координата  $1$  — в однородную координату  $H=px+s$ . Элемент  $a$  матрицы преобразования задает масштабирование точки, а элемент  $m$  — ее перенос, так же, как и в случае нормальных координат, рассмотренном выше. Если  $p=0$  и  $s=1$ , координата  $H$  равна единице, следовательно, координата  $X$  является нормальной (нормализованной), а результат преобразования лежит на прямой  $H=1$ . Именно такие преобразования являются аффинными.

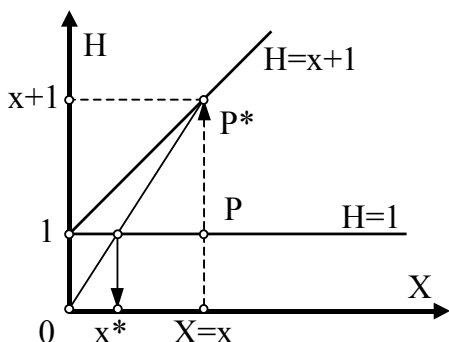


Рис. 11. Преобразование одномерной точки при  $p=1$

Рассмотрим случай, когда матрица преобразования имеет вид

$$T = \begin{vmatrix} 1 & 1 \\ 0 & 1 \end{vmatrix}.$$

Здесь элемент  $p$  матрицы равен не нулю, а единице. Координата  $H$  при этом принимает значение  $x + 1$ , и задает прямую  $H = x + 1$ , на которой находятся преобразованные точки (Рис. 11).

Некоторая произвольная точка  $P$  преобразуется в точку  $P^*$ , которая находится на прямой  $H = x + 1$ . Точка  $P^*$ , будучи заданной в однородных координатах, лежит также на прямой  $OP^*$ , задающей геометрическое место однородных координат точки  $P^*$ . Пересечение этой прямой с прямой  $H = 1$  дает нормальную преобразованную координату  $x^*$  точки  $P$ .

Таким образом, координата  $p$ , не равная нулю, определяет такое преобразование, при котором точки «переходят» с прямой  $H = 1$  на прямую  $H = xp + 1$ , что вносит дополнительное преобразование координаты  $x$ . Можно сказать, что элемент  $p$  вносит непропорциональное масштабирование точки.

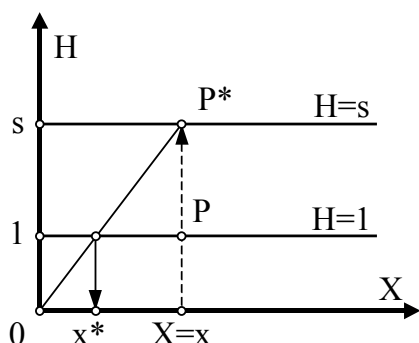


Рис. 12. Преобразование одномерной точки при  $s \neq 1$

Рассмотрим теперь случай, когда матрица преобразования имеет вид

$$T = \begin{vmatrix} 1 & 0 \\ 0 & s \end{vmatrix}.$$

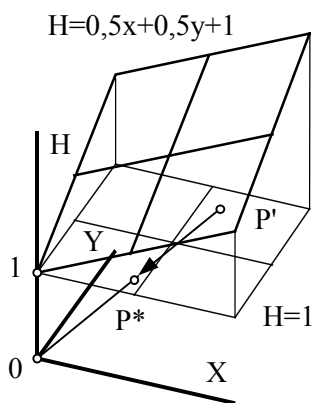
Здесь элемент  $s$  матрицы не равен 1. При этом преобразованные точки «переходят» на плоскость  $H=s$ . Нормальная координата  $x^*$  преобразованной точки принимает значение  $x/s$ . Происходит пропорциональное масштабирование точки, причем, значение  $s > 1$  масштабирует в сторону уменьшения, а  $s < 1$  — в сторону увеличения нормальной координаты  $x$ .

Если элемент  $p \neq 0$  и элемент  $s \neq 1$ , происходит еще более сложное преобразование, являющееся суперпозицией описанных двух. В целом можно сказать, что *однородные координаты позволяют одинаковым способом выполнять произвольные преобразования* (аффинные и не аффинные).

Перейдем к рассмотрению двухмерных однородных координат. Общая матрица преобразования будет теперь иметь вид

$$T = \begin{vmatrix} a & b & p \\ c & d & q \\ m & n & s \end{vmatrix}.$$

После преобразования координаты точки  $(x, y, 1)$  преобразуются в новые координаты  $(ax+cy+m, bx+dy+n, px+qy+s)$ . При этом преобразования являются аффинными, если элементы третьего столбца матрицы равны:  $p=0, q=0, s=1$ . В этом случае преобразованные координаты лежат в плоскости  $H=1$  и нормализации координат не требуется. В противном случае плоскость  $H=1$  преобразуется в плоскость  $H=px+qy+s$ , которая является геометрическим местом преобразованных координат.



**Рис. 13. Преобразование плоскости  $H=1$**

Исходные двумерные однородные координаты располагаются в пространстве на плоскости  $H=1$ . Преобразованные двумерные однородные координаты располагаются в пространстве на плоскости  $H=px+qy+s$ , которая проходит через точку  $(0, 0, s)$ . На Рис. 13 показано преобразование плоскости  $H=1$  в плоскость  $H=0,5x+0,5y+1$ , когда элементы матрицы преобразования имеют значения  $p=0,5$ ,  $q=0,5$ ,  $s=1$ .

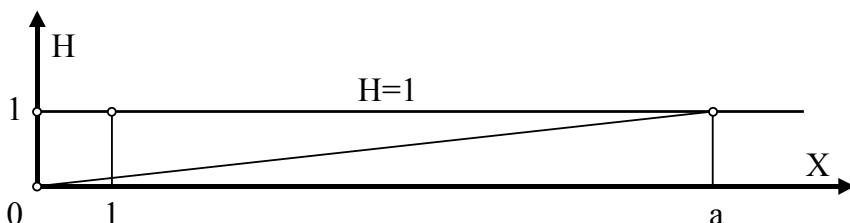
Нормализованные координаты преобразованной точки получаются на пересечении с плоскостью  $H=1$  прямой, проведенной из преобразованной точки в плоскости  $H=px+qy+s$ . На Рис. 13 точка  $P'$  на плоскости  $H=0,5x+0,5y+1$  после нормализации переходит в точку  $P^*$  на плоскости  $H=1$ .

Элементы  $a$ ,  $b$ ,  $c$ ,  $d$  матрицы вращения описывают растяжение, отображение, сдвиг и вращение, элементы  $m$  и  $n$  — перенос, элемент  $s$  — общее изменение масштаба, а элементы  $p$  и  $q$  — проекции.

## Точки в бесконечности

Однородные координаты дают возможность обозначать точки в бесконечности конечными координатами. Рассмотрим для простоты одномерный случай. Когда координата  $x$  стремится к бесконечности, однородная координата  $x$  на прямой  $H=1$  также стремится к бесконечности. В пределе геометрическое место преобразованных бесконечных точек стремится к оси  $X$ , а это означает, что точка с координатами  $(x, 0)$  представляет собой точку в бесконечности.

Рассмотрим точку  $(1, 1)$ , которая перемещается вдоль оси  $X$  в точку  $(a, 1)$  и далее в бесконечность (Рис. 14).



**Рис. 14. Точка в бесконечности**

По мере продвижения нормальная координата  $x$  растет. Если рассматривать, как изменяется точка  $(1, h)$ , получим следующий ряд (в силу множественности однородного представления):



$x^*$	$x=1$	$h$
1	1	1
10	1	1/10
100	1	1/100
1000	1	1/1000
...	...	...

Из таблицы видно, как по мере удаления точки в бесконечность, координата  $h$  точки  $(1, h)$  стремится к нулю. На практике за бесконечную принимают точку  $(1, 0)$  в одномерной системе координат. В двумерной системе координат бесконечная точка на оси  $X$  имеет координаты  $(1, 0, 0)$ , а бесконечная точка на оси  $Y$  —  $(0, 1, 0)$ .

## Контрольные вопросы и задания

1. Какое действие оказывает элемент  $a$  матрицы преобразования  $2 \times 2$  на положение точки? Элемент  $b$ ,  $c$ ,  $d$ ?
2. Какие преобразования называются аффинными?
3. Какие из перечисленных ниже точек являются точками в бесконечности, учитывая, что точки заданы в двумерных однородных координатах:
  - а)  $(0, 0, 0)$ ;
  - б)  $(1, 1, 1)$ ;
  - в)  $(5, 5, 5)$ ;
  - г)  $(1, 1, 1)$ ;
  - д)  $(0, 0, 5)$ ;
  - е)  $(1, 0, 0)$ ?

4. Какие из перечисленных в вопросе 3 точек не лежат в плоскости  $H=1$ ?
5. Какая из матриц выполняет перенос? Куда?

$$T1 = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 1 \end{vmatrix}, \quad T2 = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{vmatrix}.$$

6. Какая из матриц является матрицей вращения:

$$T3 = \begin{vmatrix} \cos & \sin & 0 \\ -\sin & \cos & 0 \\ 0 & 0 & 1 \end{vmatrix}, \quad T4 = \begin{vmatrix} \cos & -\sin & 0 \\ \sin & \cos & 0 \\ 0 & 0 & 1 \end{vmatrix}?$$

7. Какое преобразование выполняет следующая матрица в двумерных однородных координатах:

$$T5 = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{vmatrix}?$$

8. Напишите матрицу отражения объекта сверху вниз.
9. Напишите матрицу поворота объекта на  $180^\circ$ .

## Аффинные преобразования в пространстве

Будем рассматривать преобразования в пространстве в однородных координатах. Пространственная точка описывается четверкой  $(x, y, z, 1)$ , или, в общем случае, четверкой  $(x \cdot h, y \cdot h, z \cdot h, h)$ ,  $h \neq 0$  — произвольное число.

Матрица пространственного преобразования имеет размерность  $4 \times 4$ . В результате умножения пространственной точки  $(x, y, z, 1)$  на матрицу преобразования получим  $|X \ Y \ Z \ H| = |x^* \ y^* \ z^* \ 1|$ :

$$\begin{vmatrix} x & y & z & 1 \end{vmatrix} \cdot \begin{vmatrix} a & b & c & p \\ d & e & f & q \\ g & i & j & r \\ l & m & n & s \end{vmatrix} = \begin{vmatrix} x^* & y^* & z^* & 1 \end{vmatrix}.$$

Ненормализованные координаты после преобразования имеют вид:

$$\begin{aligned} X &= ax + dy + gz + l, \\ Y &= bx + ey + iz + m, \\ Z &= cx + fy + jz + n, \\ H &= px + qy + rz + s. \end{aligned}$$

Нормализованные координаты получаются делением на координату  $H$ :

$$\begin{aligned} x^* &= X/H, \\ y^* &= Y/H, \\ z^* &= Z/H. \end{aligned}$$

Преобразованная точка в общем случае переходит в плоскость, описываемую соотношением  $H = px + qy + rz + s$ . В случае аффинных преобразований точка переходит в плоскость  $H=1$ . Как и в двухмерном случае, матрицу преобразования можно разделить на четыре части по тому действию, которое они оказывают на преобразование:

$$\left| \begin{array}{c|c} 3 \times 3 & 3 \times 1 \\ \hline 1 \times 3 & s \end{array} \right|.$$

Элементы подматрицы  $3 \times 3$  выполняют частичное изменение масштаба (растяжение и сжатие), отображение, сдвиг и вращение, элементы подматрицы  $1 \times 3$  выполняют пространственный перенос, элемент  $s$  выполняет общее преобразование масштаба, а элементы подматрицы  $3 \times 1$  отвечают за проекции.

## Трехмерное вращение (*rotation*)

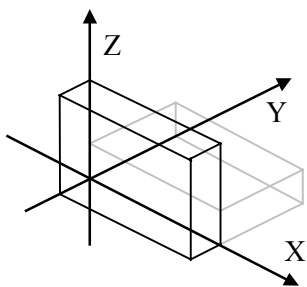
Вращения в пространстве осуществляются вокруг координатных осей.

Матрица вращения вокруг оси  $X$ :

$$R_X = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}.$$

Нормализованные координаты вращения вокруг оси  $X$  получают из следующих соотношений:

$$\begin{aligned} x^* &= x, \\ y^* &= y \cdot \cos\theta - z \cdot \sin\theta, \\ z^* &= y \cdot \sin\theta + z \cdot \cos\theta, \end{aligned}$$



**Рис. 15. Вращение вокруг оси X**

Матрица вращения вокруг оси Y:

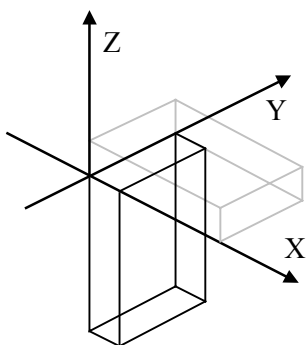
$$R_Y = \begin{vmatrix} \cos\theta & 0 & -\sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}.$$

Нормализованные координаты:

$$x^* = x \cdot \cos\theta + z \cdot \sin\theta,$$

$$y^* = y,$$

$$z^* = z \cdot \cos\theta - x \cdot \sin\theta,$$



**Рис. 16. Вращение вокруг оси Y**

Матрица вращения вокруг оси Z:

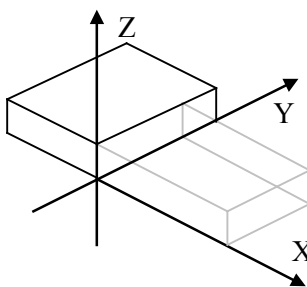
$$R_Z = \begin{vmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}.$$

Нормализованные координаты:

$$x^* = x \cdot \cos\theta - y \cdot \sin\theta,$$

$$y^* = x \cdot \sin\theta + y \cdot \cos\theta,$$

$$z^* = z,$$



**Рис. 17. Вращение вокруг оси Z**

## Растяжение-сжатие (*dilatation*)

Матрица растяжения-сжатия выполняет частичное изменение масштаба вдоль оси  $X$  на коэффициент  $a$ , вдоль оси  $Y$  на коэффициент  $e$  и вдоль оси  $Z$  на коэффициент  $j$ :

$$D = \begin{vmatrix} a & 0 & 0 & 0 \\ 0 & e & 0 & 0 \\ 0 & 0 & j & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}.$$

Нормализованные координаты:

$$x^* = ax,$$

$$y^* = ey,$$

$$z^* = jz.$$

Если коэффициент растяжения-сжатия меньше 1, происходит сжатие, а если больше 1, то растяжение.

## Отражение (*reflection*)

Отражения в пространстве осуществляются относительно трех координатных плоскостей.

Матрица отражения относительно плоскости  $YZ$ :

$$R_{YZ} = \begin{vmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}.$$

Нормализованные координаты:

$$x^* = -x,$$

$$y^* = y,$$

$$z^* = z.$$

Матрица отражения относительно плоскости  $ZX$ :

$$R_{ZX} = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}.$$

Нормализованные координаты:

$$x^* = x,$$

$$y^* = -y,$$

$$z^* = z.$$

Матрица отражения относительно плоскости  $XY$ :

$$R_{XY} = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}.$$

Нормализованные координаты:

$$x^* = x,$$

$$y^* = y,$$

$$z^* = -z.$$

## Перенос (*translation*)

Матрица переноса на вектор  $(l, m, n)$ :

$$T = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ l & m & n & 1 \end{vmatrix}.$$

Нормализованные координаты:

$$\begin{aligned} x^* &= x + l, \\ y^* &= y + m, \\ z^* &= z + n. \end{aligned}$$

## Пример аффинных преобразований в пространстве

Рассмотрим вращение на угол  $\alpha$  вокруг произвольной прямой  $L$ , которая проходит через точку  $A(a, b, c)$  и имеет нормализованный направляющий вектор  $(l, m, n)$ . Преобразование выполняется в несколько шагов.

**Шаг 1.** Перенос точки  $A$  в начало системы координат. Это выполняет матрица переноса  $T$  на вектор  $(-a, -b, -c)$ :

$$T = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -a & -b & -c & 1 \end{vmatrix}.$$

После переноса направляющие векторы  $l, m, n$  прямой расположены на осях координат, образуя вектор направления  $L$  (Рис. 18).

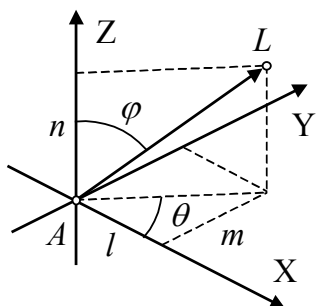


Рис. 18. Направляющий вектор прямой после переноса в начало координат

Для поворота вокруг вектора  $L$  его нужно расположить вдоль одной из координатных осей и выполнить поворот вокруг этой оси. Например, повернем ось  $X$  вокруг оси  $Z$  на угол  $\theta$ , в результате чего вектор направления  $L$  окажется в плоскости  $XZ$ . Далее повернем ось  $Z$  на угол  $\varphi$  относительно оси  $Y$ , в результате чего вектор направления  $L$  совместится с осью  $Z$ .

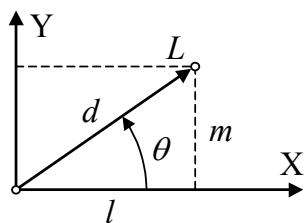
**Шаг 2.** Поворот объекта вокруг оси  $Z$  на угол  $\theta$ . Синус и косинус этого угла определяется из проекции вектора направления  $L$  на плоскость  $XY$  (Рис. 19):

$$\sin\theta = l/d, \cos\theta = m/d.$$

Здесь  $d = \sqrt{l^2 + m^2}$ . Матрица поворота:

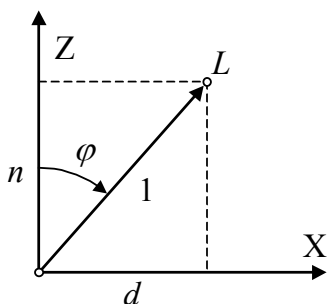
$$R_{ZT} = \begin{vmatrix} l/d & m/d & 0 \\ -m/d & l/d & 0 \\ 0 & 0 & 1 \end{vmatrix}.$$

Так как преобразование аффинное и не затрагивающее перенос, мы сокращаем матрицу, удаляя четвертый столбец и четвертую строку.



**Рис. 19. Расчет угла поворота вокруг оси Z**

**Шаг 3.** Поворот объекта вокруг оси Y на угол  $\varphi$ . После предыдущего поворота вектор направления L находится в плоскости ZX (Рис. 20).



**Рис. 20. Расчет угла поворота вокруг оси Y**

Косинус и синус угла определяются через вектор направления L и его проекции на ось Z и плоскость XY:

$$\sin \varphi = n, \cos \varphi = d.$$

Матрица поворота вокруг оси Y на угол  $\varphi$  будет иметь вид:

$$R_{Y\varphi} = \begin{vmatrix} d & 0 & -n \\ 0 & 1 & 0 \\ n & 0 & d \end{vmatrix}.$$

**Шаг 4.** Собственно поворот вокруг направления L. Выполняется как поворот вокруг оси Z:

$$R_{Z\alpha} = \begin{vmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{vmatrix}.$$

Теперь нужно вернуть объект на место, то есть выполнить шаги 1–3 в обратном порядке (учитывая, что повороты и перенос выполняются в обратную сторону).

**Шаг 5.** Поворот вокруг оси Y на угол  $-\varphi$ :

$$R_{Y-\varphi} = \begin{vmatrix} d & 0 & n \\ 0 & 1 & 0 \\ -n & 0 & d \end{vmatrix}.$$

Здесь учитывается, что  $\sin -\varphi = -\sin \varphi$ ,  $\cos -\varphi = \cos \varphi$ .

**Шаг 6.** Поворот вокруг оси Z на угол  $-\theta$ :

$$R_{Z-\theta} = \begin{vmatrix} l/d & -m/d & 0 \\ m/d & l/d & 0 \\ 0 & 0 & 1 \end{vmatrix}.$$

Здесь учитывается, что  $\sin -\theta = -\sin \theta$ ,  $\cos -\theta = \cos \theta$ .

Шаг 7. Перенос на вектор  $(a, b, c)$ :

$$T_- = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ a & b & c & 1 \end{vmatrix}.$$

Для выполнения преобразования полученные матрицы можно перемножить и получить обобщенную матрицу преобразования. Однако в большинстве случаев выполнение последовательности отдельных преобразований оказывается более экономичным в смысле количества элементарных арифметических операций, так как коэффициенты общей матрицы преобразования оказываются сложными.

## Контрольные вопросы и задания

1. Какая из матриц является матрицей вращения вокруг оси Y:

$$T6 = \begin{vmatrix} \cos & 0 & \sin & 0 \\ 0 & 1 & 0 & 0 \\ -\sin & 0 & \cos & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}, \quad T7 = \begin{vmatrix} \cos & 0 & -\sin & 0 \\ 0 & 1 & 0 & 0 \\ \sin & 0 & \cos & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} ?$$

2. Какое преобразование выполняет следующая матрица:

$$T8 = \begin{vmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} ?$$

3. Напишите матрицу трехмерного переноса из начала системы координат в на расстояние 100 вдоль оси X.
4. Составьте матрицу трехмерного поворота объекта вокруг оси Z на угол  $90^\circ$  относительно точки A с координатами  $(a, b, c)$ .

## Проективные преобразования

До сих пор мы рассматривали преобразования, которые приводят к изменению объекта — растяжению, отражению, вращению, сдвигу и переносу. Часто вместо этого нужно получить изображение объекта с некоторой точки зрения. Преобразования, которые устанавливают определенную точку зрения на объект, называют *проективными преобразованиями*, а сам процесс — *проектированием* или *проецированием*. Результат проецирования будем называть *проекцией*.

Суть проективных преобразований заключается в «просвечивании» объекта проективными (проектирующими) лучами на плоскость, в которой формируется изображение — *плоскость проекции* (ее также называют *картинной* или *экранной*). Проективные лучи обычно пропускаются через определенные точки — вершины многоугольных граней. Так как при преобразовании прямых их линейность не нарушается, результирующее изображение можно получить простым соединением преобразованных вершин (Рис. 21).

В зависимости от того, как выпускаются проектирующие лучи, различают *параллельные* и *перспективные* проекции.

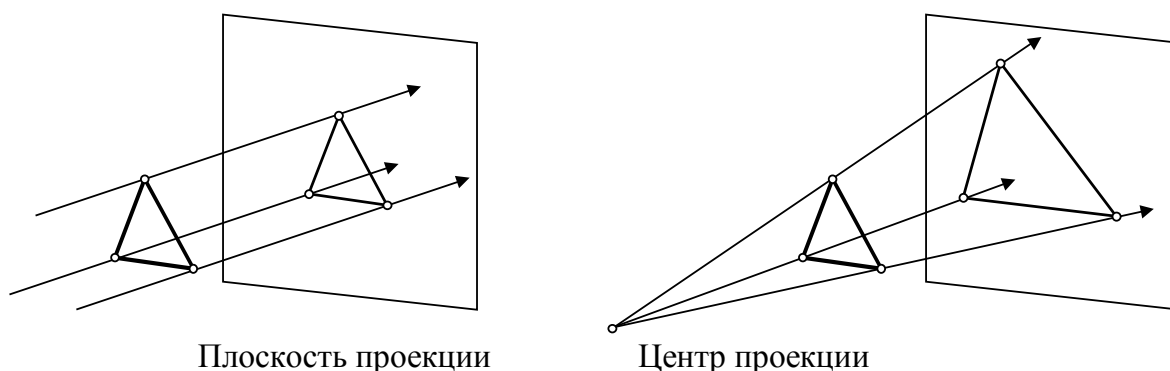


Рис. 21. Параллельное (слева) и перспективное (справа) проецирование

В параллельных проекциях проектирующие лучи выпускаются параллельно друг другу. В перспективных проекциях лучи исходят из одной точки — центра проекции. Параллельные проекции можно рассматривать как частный случай перспективных, когда центр проекции находится в бесконечности.

Основным назначением проекций является получение двумерных координат для вывода на устройство отображения.

## Параллельные проекции

Параллельные проекции классифицируются по взаимному расположению проектирующих лучей, плоскости проекции и системы координат объекта.

На Рис. 22 приведен пример классификации параллельных проекций. На первом уровне их делят на ортографические, аксонометрические и косоугольные. Аксонометрические проекции делят на прямоугольные, триметрические, диметрические и изометрические по положению системы координат объекта относительно проектирующих лучей. Косоугольные проекции делятся на свободные и кабинетные проекции, различающиеся разным масштабом вдоль одной из осей. Рассмотрим перечисленные виды проекций более подробно.



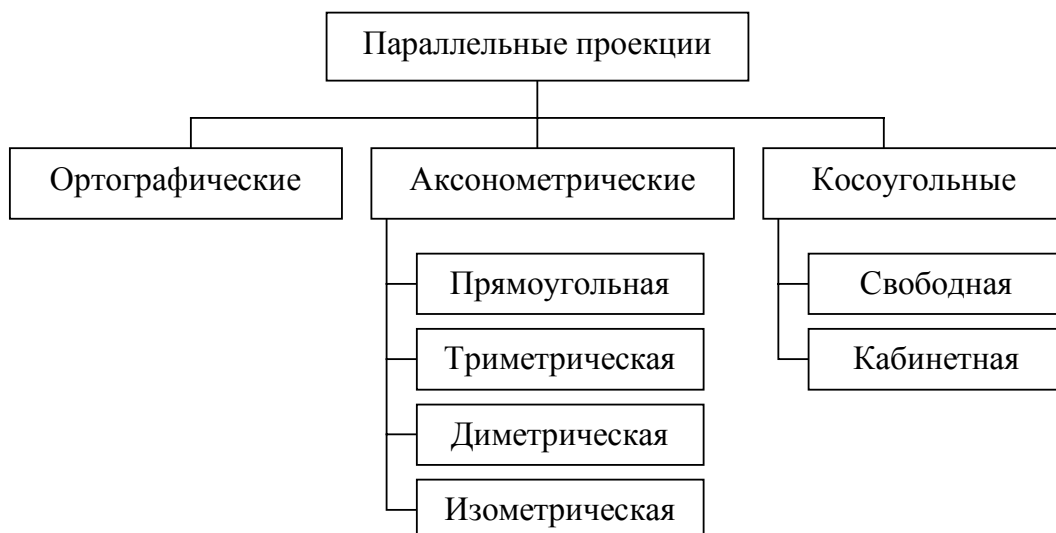


Рис. 22. Классификация параллельных проекций

### Ортография

Ортографические проекции получаются, когда проектирующие лучи перпендикулярны плоскости проекции, а плоскость проекции является одной из координатных плоскостей, или плоскостью, параллельной одной из координатных. Это самые простые проекции, которые никак не преобразуют объект, а просто получают его вид с той или иной стороны. Эти проекции далее можно использовать для дальнейшего проецирования с целью установить точку зрения так, чтобы проекция была видна под подходящим углом. Ортографические проекции иногда относят к аксонометрическим и называют *ортогональными*. На мой взгляд, ортографические проекции отличаются от аксонометрических тем, что не изменяют систему координат объекта. В аксонометрических проекциях участвуют элементы подматрицы преобразования  $3 \times 3$ , в то время как в ортографических проекциях участвуют только элементы подматрицы  $1 \times 3$ .

Ортографическая проекция вдоль оси  $Z$  выполняется при помощи единичной матрицы (нет преобразования). Так как координата  $Z$  обычно не требуется, третий столбец матрицы преобразования может быть принят равным нулю, что приведет к обнулению координат  $Z$ , и фактическому расположению проекции в плоскости  $Z=0$  (чистая ортография). Однако иногда требуется, чтобы проекция располагалась в плоскости  $Z=n$ . Тогда используется матрица переноса, в которой координата  $z$  в третьем столбце исключена. Матрица ортографической проекции получается в результате перемножения матрицы чистой ортографии и матрицы переноса:

$$\text{ORTHO}_Z = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \cdot \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & n & 1 \end{vmatrix} = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & n & 1 \end{vmatrix}.$$

Преобразование может быть выполнено по формулам

$$\begin{aligned} x^* &= x, \\ y^* &= y, \\ z^* &= n. \end{aligned}$$

Аналогично выполняется ортография на плоскости  $X=0$  и  $Y=0$ , или на плоскости, им параллельные ( $X=l$  или  $Y=m$ ):

$$\text{ORTHO}_X = \begin{vmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \cdot \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ l & 0 & 0 & 1 \end{vmatrix} = \begin{vmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ l & 0 & 0 & 1 \end{vmatrix}.$$

$$\text{ORTHO}_Y = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \cdot \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & m & 0 & 1 \end{vmatrix} = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & m & 0 & 1 \end{vmatrix}.$$

### АксонOMETрические проекции

АксонOMETрические проекции получаются, когда проектирующие лучи перпендикулярны плоскости проецирования. В зависимости от положения системы координат объекта по отношению к плоскости проектирования их делят на прямоугольные, триметрические, диметрические и изометрические.

В аксонOMETрических преобразованиях используются матрицы вращения.

### Прямоугольные проекции

Прямоугольные или ортогональные проекции являются чрезвычайно важными — на их основе выполняются чертежи деталей для их изготовления. Они показывают вид на деталь с той или иной стороны, направленной вдоль одной из осей координат. Эти проекции получаются, когда система координат объекта расположена ортогонально по отношению к плоскости проекции.

Всего получается 6 проекций (Рис. 23): виды спереди и сзади (со стороны положительного и отрицательного направления оси  $Z$  соответственно), справа и слева (со стороны положительного и отрицательного направления оси  $X$ ), и сверху и снизу (со стороны положительного и отрицательного направления оси  $Y$ ).

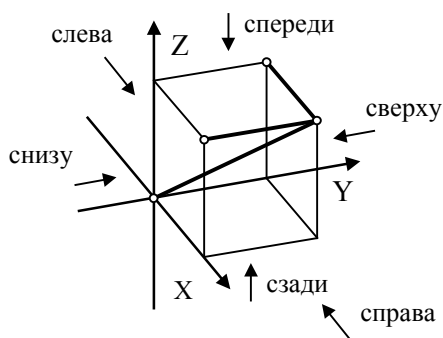


Рис. 23. Ортогональные виды

Следует обратить внимание на то, что за вид спереди принимается вид со стороны положительного направления оси  $Z$ . Это связано, видимо, с тем, что при черчении детали плоскостью чертежа является плоскость  $XY$ . Именно в этой плоскости чертится основной вид на деталь, который принимается за вид спереди.

Этот факт следует учитывать при разработке графических объектов.

Проекция «Вид спереди» — это ортографическая проекция на плоскость  $Z=0$ . Для ее получения не нужно выполнять никакого преобразования или применить преобразование «чистая ортография Z», обнуляющее координату Z.



**Рис. 24. Вид спереди (слева) и вид сзади (справа)**

Проекция «Вид сзади» — это проекция «Вид спереди», повернутая на  $180^\circ$  вокруг оси X. Учитывая, что  $\sin 180^\circ = 0$ , а  $\cos 180^\circ = -1$ , матрица преобразования имеет вид:

$$\text{BACK} = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}.$$

Преобразование выполняется по формулам

$$\begin{aligned} x^* &= x, \\ y^* &= -y, \\ z^* &= -z. \end{aligned}$$



**Рис. 25. Вид слева (слева) и вид справа (справа)**

Проекция «Вид слева» — это поворот вокруг оси Y на угол  $90^\circ$ :

$$\text{LEFT} = \begin{vmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}.$$

Преобразование выполняется по формулам

$$\begin{aligned} x^* &= z, \\ y^* &= y, \\ z^* &= -x. \end{aligned}$$

Проекция «Вид справа» — это вид с положительного направления оси X. Чтобы получить этот вид, нужно повернуть объект на угол  $-90^\circ$  вокруг оси Y. Учитывая, что  $\sin -90^\circ = -1$ , а  $\cos -90^\circ = 0$ , матрица преобразования имеет вид:

$$\text{RIGHT} = \begin{vmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}.$$

Преобразование выполняется по формулам

$$\begin{aligned}x^* &= -z, \\y^* &= y, \\z^* &= x.\end{aligned}$$



Рис. 26. Вид сверху (слева) и вид снизу (справа)

Проекция «Вид сверху» получается при повороте на угол  $90^\circ$  вокруг оси  $X$ :

$$\text{TOP} = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & \mathbf{0} & \mathbf{1} & 0 \\ 0 & -\mathbf{1} & \mathbf{0} & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}.$$

Преобразование выполняется по формулам

$$\begin{aligned}x^* &= x, \\y^* &= -z, \\z^* &= y.\end{aligned}$$

Наконец, проекция «Вид снизу» — это поворот вокруг оси  $X$  на угол  $-90^\circ$ :

$$\text{BOTTOM} = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & \mathbf{0} & -\mathbf{1} & 0 \\ 0 & \mathbf{1} & \mathbf{0} & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}.$$

Преобразование выполняется по формулам

$$\begin{aligned}x^* &= x, \\y^* &= z, \\z^* &= -y.\end{aligned}$$

### Триметрия, диметрия и изометрия

В триметрических, диметрических и изометрических проекциях оси координат объекта наклонены к плоскости проекции с целью получения трехмерного изображения объекта. Эти проекции получают двумя вращениями.

В триметрической проекции углы между осями координат и плоскостью проекции попарно различны. Для получения проекции объект поворачивается вокруг оси  $Y$  на угол  $\alpha$ , а затем вокруг оси  $X$  на угол  $\beta$ :

$$T = \begin{vmatrix} \cos\alpha & 0 & -\sin\alpha & 0 \\ 0 & 1 & 0 & 0 \\ \sin\alpha & 0 & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \cdot \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\beta & \sin\beta & 0 \\ 0 & -\sin\beta & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}.$$

В результате этих преобразований единичные орты координатных осей принимают следующие значения:

$$\begin{aligned}e_X &= |1, 0, 0, 1| \cdot T = | \cos\alpha & \sin\alpha \cdot \sin\beta & -\sin\alpha \cdot \cos\beta & 1 |, \\e_Y &= |0, 1, 0, 1| \cdot T = | 0 & \cos\beta & \sin\beta & 1 |, \\e_Z &= |0, 0, 1, 1| \cdot T = | \sin\alpha & -\cos\alpha \cdot \sin\beta & \cos\alpha \cdot \cos\beta & 1 |.\end{aligned}$$

Обобщенная матрица триметрического преобразования имеет вид:

$$T = \begin{vmatrix} \cos\alpha & \sin\alpha \cdot \sin\beta & -\sin\alpha \cdot \cos\beta \\ 0 & \cos\beta & \sin\beta \\ \sin\alpha & -\cos\alpha \cdot \sin\beta & \cos\alpha \cdot \cos\beta \end{vmatrix}.$$

Преобразованные координаты определяются соотношениями

$$\begin{aligned} x^* &= x \cdot \cos\alpha + z \cdot \sin\alpha, \\ y^* &= x \cdot \sin\alpha \cdot \sin\beta + y \cdot \cos\beta - z \cdot \cos\alpha \cdot \sin\beta, \\ z^* &= -x \cdot \sin\alpha \cdot \cos\beta + y \cdot \sin\beta + z \cdot \cos\alpha \cdot \cos\beta. \end{aligned}$$

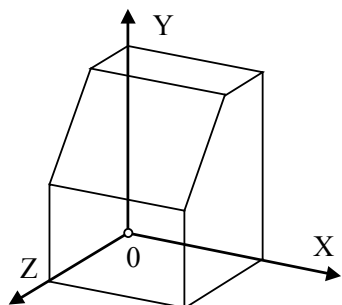
Если преобразование является окончательным и используется непосредственно для вывода, то координату  $z$  можно не считать. При этом к матрице  $T$  дополнительно применяется преобразование «чистая ортография  $Z$ », в результате которого единичные орты получают значения

$$\begin{aligned} e_X &= |1, 0, 0, 1| \cdot T = | \cos\alpha & \sin\alpha \cdot \sin\beta & 0 & 1 |, \\ e_Y &= |0, 1, 0, 1| \cdot T = | 0 & \cos\beta & 0 & 1 |, \\ e_Z &= |0, 0, 1, 1| \cdot T = | \sin\alpha & -\cos\alpha \cdot \sin\beta & 0 & 1 |, \end{aligned}$$

а преобразованные координаты определяются соотношениями

$$\begin{aligned} x^* &= x \cdot \cos\alpha + z \cdot \sin\alpha, \\ y^* &= x \cdot \sin\alpha \cdot \sin\beta + y \cdot \cos\beta - z \cdot \cos\alpha \cdot \sin\beta, \\ z^* &= 0. \end{aligned}$$

Результат триметрического преобразования зависит от выбранных углов. Для случая, когда  $\alpha = -30^\circ$  а  $\beta = 20^\circ$ , проекция имеет вид, показанный на Рис. 27.



**Рис. 27. Триметрическое преобразование**

В диметрическом преобразовании длины единичных векторов осей  $X$  и  $Y$  сокращают в равное число раз. Приравнивая длины орт  $e_X$  и  $e_Y$ , получаем

$$\cos^2\alpha + \sin^2\alpha \cdot \sin^2\beta = \cos^2\beta.$$

С учетом тождеств  $1 - \sin^2\alpha = \cos^2\alpha$  и  $1 - \sin^2\beta = \cos^2\beta$  получаем

$$(1) \quad \sin^2\alpha = \sin^2\beta / (1 - \sin^2\beta).$$

Для определения  $\sin\beta$  применяют сокращение длины оси  $Z$  в фиксированное число раз. Например, сократив единичный вектор оси  $Z$  в два раза, получим

$$\sin^2\alpha + \cos^2\alpha \cdot \sin^2\beta = (1/2)^2.$$

Заменим  $\cos^2\alpha$ , получим:

$$\sin^2\alpha + (1 - \sin^2\alpha) \cdot \sin^2\beta = 1/4.$$

С учетом соотношения (1) и заменив  $\sin^2\beta$  переменной  $\omega$ , получаем квадратное относительно  $\omega$  уравнение:

$$\omega / (1 - \omega) + (1 - \omega / (1 - \omega)) \cdot \omega = 1/4;$$

$$\begin{aligned}\omega/(1-\omega)+\omega-\omega^2/(1-\omega) &= 1/4; \\ 4\omega+4\omega(1-\omega)-4\omega^2 &= 1-\omega; \\ 4\omega+4\omega-4\omega^2-4\omega^2-1+\omega &= 0; \\ 8\omega^2-9\omega+1 &= 0.\end{aligned}$$

Решая уравнение, получаем два корня  $\omega=1$  и  $\omega=1/8$ . Принимая  $\omega=\sin^2\beta=1/8$ , получаем  $\beta=20,704811054635429669240251093463^\circ$  (вычисления выполнены на калькуляторе Windows<sup>®</sup> XP). Теперь из соотношения (1) можно найти  $\sin^2\alpha$ :

$$\sin 2\alpha = 1/8/(1-1/8) = 1/7,$$

откуда

$$\alpha = 22,207654298596487487055620026178^\circ.$$

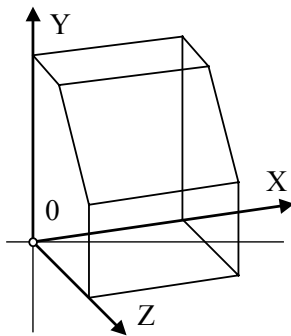
Матрица диметрического преобразования:

$$\text{Dimet} = \begin{vmatrix} 0,935414346693485 & 0,133630620956212 & -0,327326835353989 \\ 0 & 0,925820099772551 & 0,377964473009227 \\ 0,353553390593274 & -0,353553390593274 & 0,866025403784439 \end{vmatrix}.$$

Преобразованные координаты диметрического преобразования:

$$\begin{aligned}x^* &= x \cdot 0,935414346693485 + z \cdot 0,353553390593274, \\ y^* &= x \cdot 0,133630620956212 + y \cdot 0,925820099772551 - z \cdot 0,353553390593274, \\ z^* &= z \cdot 0,866025403784439 - x \cdot 0,327326835353989 + y \cdot 0,377964473009227.\end{aligned}$$

Результат диметрического преобразования приведен на Рис. 28.



**Рис. 28. Диметрическое преобразование**

Изометрическая проекция получается в результате сокращения всех координатных осей в равное число раз. Попарно приравнявая длины орт  $e_X$  и  $e_Y$  а также  $e_Z$  и  $e_Y$ , имеем равенства:

$$\begin{aligned}\cos^2\alpha + \sin^2\alpha \cdot \sin^2\beta &= \cos^2\beta, \\ \sin^2\alpha + \cos^2\alpha \cdot \sin^2\beta &= \cos^2\beta.\end{aligned}$$

Из первого равенства получаем

$$\begin{aligned}1 - \sin^2\alpha + \sin^2\alpha \cdot \sin^2\beta &= 1 - \sin^2\beta; \\ \sin^2\alpha(1 - \sin^2\beta) &= \sin^2\beta;\end{aligned}$$

$$(2) \quad \sin^2\alpha = \sin^2\beta/(1 - \sin^2\beta),$$

а из второго

$$\begin{aligned}\sin^2\alpha + (1 - \sin^2\alpha) \cdot \sin^2\beta &= 1 - \sin^2\beta; \\ \sin^2\alpha + \sin^2\beta - \sin^2\alpha \cdot \sin^2\beta &= 1 - \sin^2\beta; \\ \sin^2\alpha(1 - \sin^2\beta) &= 1 - 2\sin^2\beta; \\ \sin^2\alpha &= 1 - 2\sin^2\beta/(1 - \sin^2\beta).\end{aligned}$$

Приравнявая правые части уравнений, получим

$$\sin^2\beta = 1 - 2\sin^2\alpha,$$

откуда  $\sin^2\beta = 1/3$ , и угол  $\beta = 35,264389682754654315377000330019^\circ$ .

Угол  $\alpha$  найдем из формулы (2):

$$\sin^2\alpha = 1/3 / (1 - 1/3) = 1/2, \alpha = 45^\circ.$$

Матрица изометрического преобразования:

$$\text{Isomet} = \begin{vmatrix} a & b & -c \\ 0 & d & c \\ a & -b & c \end{vmatrix},$$

здесь:

$a = 0,707106781186548$ ,  $b = 0,408248290463863$ ,

$c = 0,577350269189626$ ,  $d = 0,816496580927726$ .

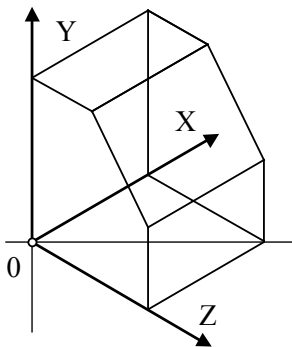
Учитывая равенство коэффициентов матрицы, вычислить преобразованные координаты изометрического преобразования можно по формулам:

$$x^* = (x+z) \cdot 0,707106781186548,$$

$$y^* = (x-z) \cdot 0,408248290463863 + y \cdot 0,816496580927726,$$

$$z^* = (z-x+y) \cdot 0,577350269189626.$$

Результат изометрического преобразования приведен на Рис. 29.



**Рис. 29. Изометрическое преобразование**

Угол, который составляет преобразованная ось  $X$  с горизонталью (экранной осью  $X$ ), можно найти из соотношения  $\text{tg}\gamma = \sin\alpha \cdot \sin\beta / \cos\alpha$  (отношение проекций орта оси  $X$ ). Учитывая, что  $\sin\alpha = \cos\alpha$ , получим  $\text{tg}\gamma = \sin\beta$ , откуда  $\gamma = 30^\circ$ . Именно поэтому для построения изометрических проекций вручную используется угольник  $30^\circ \times 60^\circ \times 90^\circ$ .

### **Косоугольные проекции**

Косоугольные проекции получаются, когда проектирующие лучи наклонены к плоскости проецирования. Наиболее известны проекции *Кавалье* (*Cavalier*), или свободная проекция, и *Кабине* (*Cabinet*), или кабинетная проекция.

Проектирующие лучи в этих проекциях наклонены под углом  $45^\circ$  к плоскости проекции. В проекции *Кавалье* масштаб оси  $Z$  не изменяется, а в проекции *Кабине* масштаб оси  $Z$  сокращен в два раза для лучшего восприятия.

Матрицы преобразования этих проекций имеют одинаковый вид:

$$T = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -a & -a & 1 \end{vmatrix},$$

Для проекции *Кавалье*  $a = \cos 45^\circ = 0,707106781186548$ , а для проекции *Кабине*  $a = \frac{1}{2} \cdot \cos 45^\circ = 0,353553390593274$ . Результат проектирования для свободной и кабинетной проекций приведен на Рис. 30.

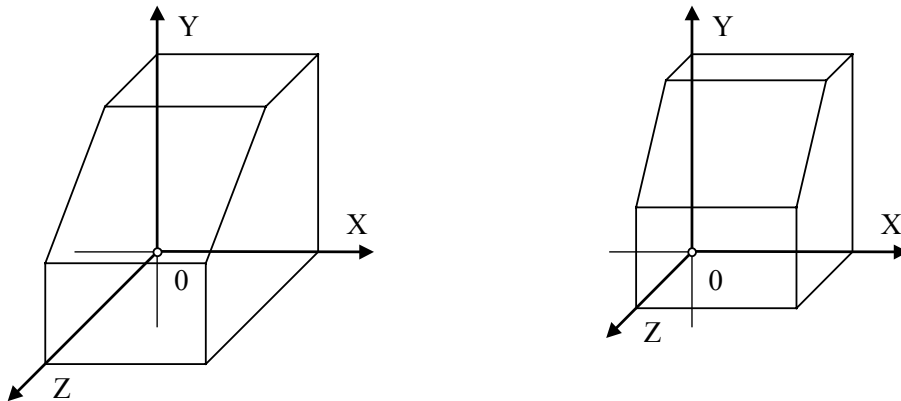


Рис. 30. Проекция Кавалье (слева) и Кабине (справа)

## Перспективные преобразования

Перспективным будем называться изображение, которое соответствует изображению, получаемому на сетчатке глаза. Рассмотрим, как формируется изображение двух одинаковых по длине отрезков А и В, расположенных на разном удалении от точки зрения Е (Рис. 31):

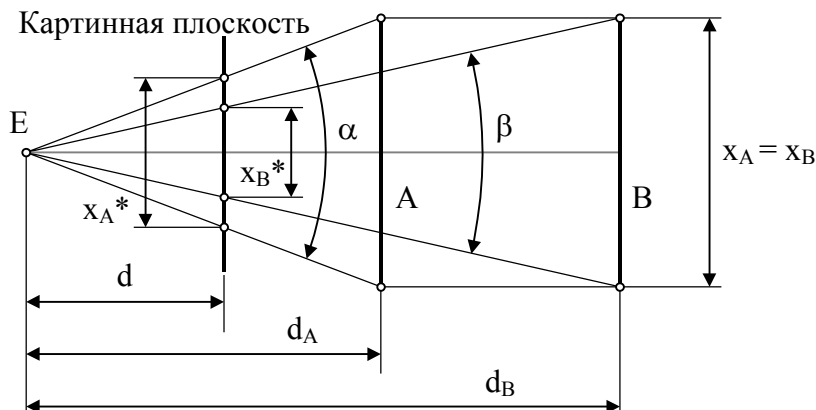


Рис. 31. Формирование изображение на сетчатке глаза

Глаз воспринимает длину отрезка через его угловой размер. Для отрезка А угловой размер равен углу  $\alpha$ , для отрезка В угловой размер равен  $\beta$ . Определить угловые размеры отрезка А длиной  $x_A$  через расстояние  $d_A$  до него можно через тангенс:

$$\operatorname{tg} \alpha = \frac{x_A}{d_A}.$$

Из этого соотношения следует, что чем дальше объект расположен от глаза, тем меньше его угловой размер, и тем меньше воспринимаемый глазом его линейный размер.

Для того, чтобы изображение на плоскости обладало необходимой перспективной, нужно обеспечить разность линейных размеров одинаковых объектов, расположенных в пространстве на разном удалении от картинной плоскости. При этом нужно выдержать правильные угловые размеры объектов.



Чтобы получить перспективное изображение, нужно вычислить линейный размер проекции объекта в соответствии с его удалением от точки зрения. Рассматривая подобные треугольники на Рис. 31, можно найти размер объекта в картинной плоскости из следующей пропорции:

$$\frac{x_A^*}{d} = \frac{x_A}{d_A}.$$

Из нее следует

$$x_A^* = x_A \frac{d}{d_A}.$$

Аналогично получим

$$x_B^* = x_B \frac{d}{d_B}.$$

Найдем отношение проекций, учитывая, что  $x_B = x_A$ :

$$\frac{x_B^*}{x_A^*} = \frac{d_A}{d_B}.$$

Из него следует важный вывод — отношение проекций зависит от расстояний до объектов и не зависит от положения картинной плоскости. Позднее, при рассмотрении простого перспективного преобразования, мы вернемся к этому выводу и рассмотрим его более детально.

Перспективное изображение обладает большей реальностью. Проекция — это плоское изображение. Все отрезки равной длины при параллельном проектировании, будут иметь равную длину на картинной плоскости. Для глаза эти равные по длине отрезки будут восприниматься как имеющие равную длину, так как картинка находится на одном и том же расстоянии от глаза. Именно поэтому изображение должно быть немного искажено, чтобы на плоском изображении отобразить разность в удалении объектов от точки зрения. На Рис. 32 изображены два куба. Слева изображение куба получено при помощи параллельной проекции, а справа — при помощи перспективной.



**Рис. 32. Неперспективное и перспективное изображение куба**

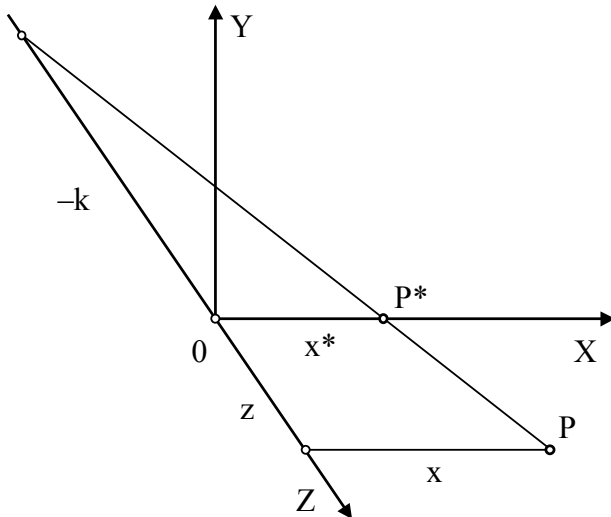
Куб слева кажется неестественным потому, что его задняя грань имеет такой же размер, как и передняя. У куба справа ребра, направленные к задней грани, не параллельны друг другу, и именно это несоответствие создает впечатление правильности. Этим приемом издавна пользуются художники и архитекторы при создании трехмерных, пространственных изображений.

Переходя к проецированию, используем аналогию в формировании изображения при помощи лучей, исходящих из точки зрения. На Рис. 33 показана модель получения перспективной проекции  $P^*$  точки  $P$ . Проектирующий луч исходит из точки, расположенной на расстоянии  $k$  от центра. Преобразованная координата  $x^*$  получают в плоскости проекции так же, как и в параллельной проекции — при пересечении плоскости проектирующим лучом. Рассматривая подобные треугольники в плоскости  $XZ$ , получим соотношение

$$\frac{x^*}{k} = \frac{x}{k+z},$$

из которого следует

$$x^* = k \frac{x_A}{k+z} = \frac{x_A}{1+z/k}.$$



**Рис. 33. Перспективная проекция точки в плоскости XZ**

Рассмотрим теперь следующее преобразование:

$$\begin{vmatrix} x & y & z & 1 \end{vmatrix} \cdot \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & r \\ 0 & 0 & 0 & 1 \end{vmatrix} = \begin{vmatrix} X & Y & Z & H \end{vmatrix}.$$

Так как четвертый столбец матрицы содержит ненулевой элемент  $r$ , преобразованный объект располагается в пространстве  $H = zr + 1$ . Нормализованные координаты будут получены из соотношений

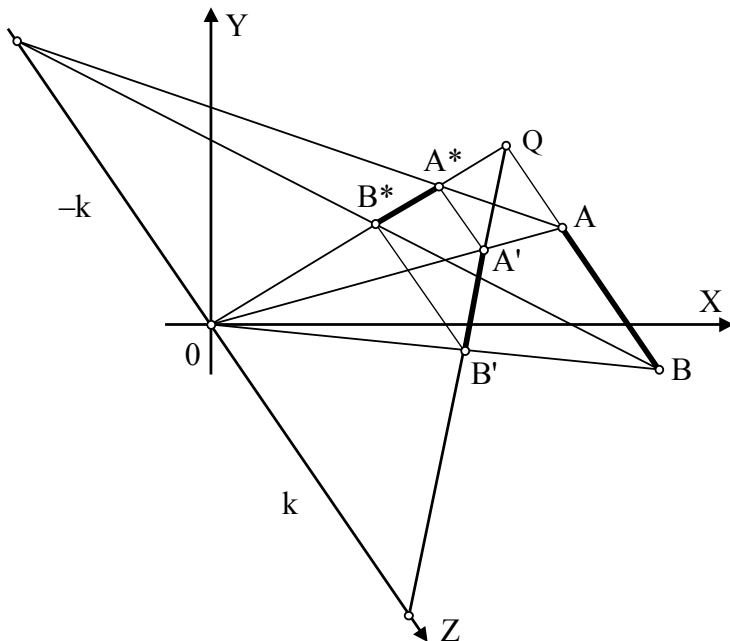
$$x^* = \frac{x}{zr+1}, \quad y^* = \frac{y}{zr+1}.$$

Нетрудно заметить, что здесь  $r = 1/k$ , или  $k = 1/r$ . Точка  $z = -1/r$  называется *центром проекции*. Точка в бесконечности на оси  $Z$ , имеющая нормальные координаты  $(0, 0, 1, 0)$ , преобразуется в точку  $(0, 0, 1, r)$ , расположенную на оси  $Z$  расстоянии  $1/r$  в положительном направлении. Иначе говоря, положительное полупространство  $z \geq 0$  проецируется в ограниченную область  $0 \leq z \leq 1/r$ . Линии, параллельные оси  $Z$ , будут сходиться в точке  $(0, 0, 1/r, 1)$ , которую называют *точкой схода перспективного преобразования*.

Сказанное иллюстрирует пример, приведенный на Рис. 34. Здесь рассматривается проекция отрезка  $AB$ , расположенного на прямой, параллельной оси  $Z$ . Эта прямая «входит» в плоскость  $z=0$  в точке  $Q$ , которую мы будем называть *точкой входа*. Перспективная проекция отрезка  $A^*B^*$  лежит на прямой, соединяющей точку входа с центром, а спроецированный отрезок  $A'B'$  располагается на прямой от точки входа до точки схода, которая является проекцией прямой, параллельной оси  $Z$  (той, на которой расположен отрезок  $AB$ ).

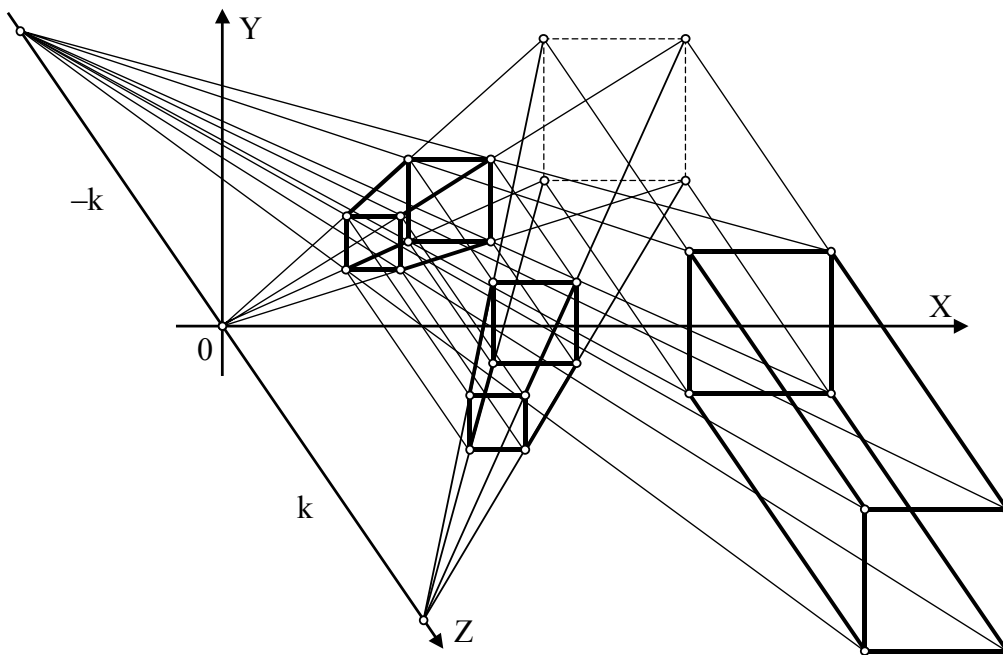
Проекция отрезка в плоскости  $z=0$  может быть получена при помощи проектирующих лучей (обычным образом). С другой стороны, проекция  $A^*B^*$  есть не что иное, как спроецированный (преобразованный) объект  $A'B'$ , у которо

го координата  $z=0$ . То есть перспективная проекция есть ортогональная проекция ORTHOZ преобразованного объекта  $A'B'$ . Это обстоятельство можно использовать для построения преобразованного объекта  $A'B'$  посредством восстановления перпендикуляров к плоскости проекции из точек перспективной проекции  $A^*$  и  $B^*$  до пересечения с преобразованной прямой  $QB$ .



**Рис. 34. Проекция отрезка, параллельного оси Z**

Заметим, что геометрически получить преобразованные точки можно также, проведя прямые из исходных точек  $A$  и  $B$  до центра. Так как все построения лежат в одной плоскости, эти прямые пересекут спроектированную прямую  $QB$  в точках  $A'B'$ .



**Рис. 35. Пример перспективного преобразования**

Чтобы лучше понять, как происходит перспективное преобразование, следует рассмотреть преобразование объемного объекта, приведенное на Рис. 35.

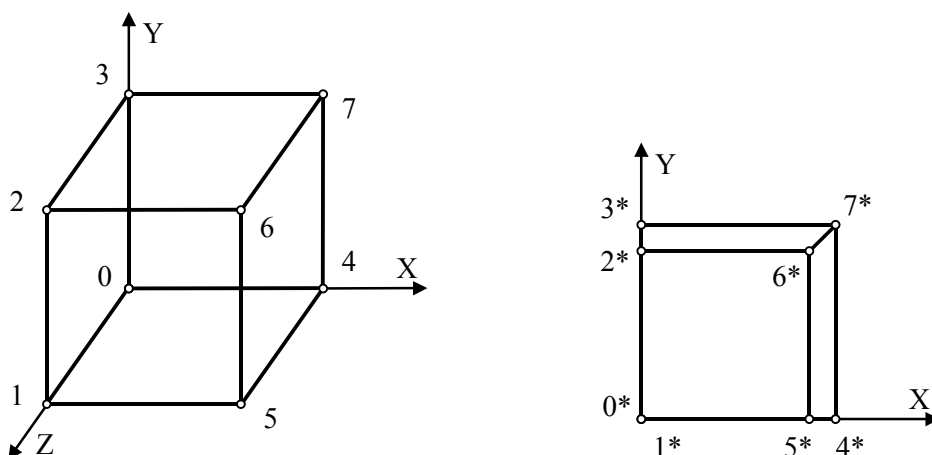
Здесь преобразуется прямоугольный брусок, для простоты расположенный перпендикулярно оси  $Z$ . Это дает четыре точки входа. Перспективная проекция объекта находится в плоскости  $XY$ , а преобразованный объект — на прямых, проведенных из точек входа в точку схода. Построить изображение можно, последовательно применяя прием, показанный на Рис. 34, ко всем восьми точкам объекта.

Аналогичные преобразования получим, если элементы  $p$  или  $q$  матрицы преобразования не равны нулю. Если  $p \neq 0$ , центр проекции находится в точке  $x = -1/p$ , точка схода в  $x = 1/p$ , и положительное полупространство  $x \geq 0$ , сжимается к точке схода. Проекция находится в плоскости  $x = 0$ . Преобразованные координаты  $y$  и  $z$  получают из соотношений

$$y^* = \frac{y}{xp+1}, \quad z^* = \frac{z}{xp+1}.$$

Если  $q \neq 0$ , получим центр проекции  $y = -1/q$ , и точку схода  $y = 1/q$ . Положительное полупространство  $y \geq 0$  сжимается к точке схода. Проекция получается в плоскости  $y = 0$ , а преобразованные координаты имеют вид

$$x^* = \frac{x}{yq+1}, \quad z^* = \frac{z}{yq+1}.$$



**Рис. 36. Одноточечное преобразование куба**

Рассмотренные преобразования называются *одноточечными* или *параллельными* перспективными преобразованиями.

На Рис. 36 (справа) приведена одноточечная перспективная проекция куба. Центр проекции расположен на оси  $Z$  в точке  $z = -1/r$ . Плоскость проекции  $z = 0$ . Грань куба 0-3-7-4 не искажена, так как она расположена в плоскости проекции. Грань 1-2-6-5 уменьшилась в размерах, ребра 2-3, 6-7 и 5-4 теперь видны. Точка наблюдения находится в центре проекции.

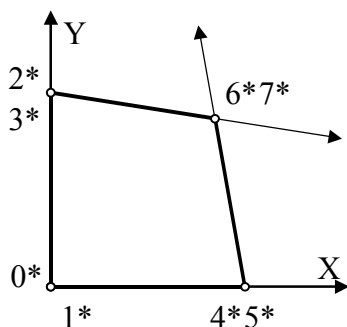
Если два элемента четвертого столбца в матрице преобразования одновременно не равны нулю, получается *двухточечная*, или *угловая* перспектива. При таком преобразовании получается два центра проекции и две точки схода, расположенные на двух координатных осях. Рассмотрим, например, преобразование:

$$\begin{vmatrix} x & y & z & 1 \end{vmatrix} \cdot \begin{vmatrix} 1 & 0 & 0 & p \\ 0 & 1 & 0 & q \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} = \begin{vmatrix} X & Y & Z & H \end{vmatrix}.$$

Обычные координаты  $(x, y, z, 1)$  преобразуются в нормальные координаты

$$x^* = \frac{x}{xp+yq+1}, \quad y^* = \frac{y}{xp+yq+1}, \quad z^* = \frac{z}{xp+yq+1}.$$

Один центр проекции находится на оси  $X$  в точке  $x = -1/p$ , второй расположен на оси  $Y$  в точке  $y = -1/q$ . Соответственно, точки схода на осях  $X$  и  $Y$  имеют координаты  $x = 1/p$  и  $y = 1/q$ . Результат двухточечного преобразования куба показан на Рис. 37.



**Рис. 37. Двухточечная перспектива куба**

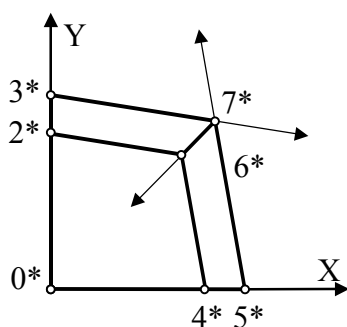
В случае, когда три элемента четвертого столбца матрицы преобразования одновременно не равны нулю, получаем *трехточечную*, или *косую* перспективу:

$$\begin{vmatrix} x & y & z & 1 \end{vmatrix} \cdot \begin{vmatrix} 1 & 0 & 0 & p \\ 0 & 1 & 0 & q \\ 0 & 0 & 1 & r \\ 0 & 0 & 0 & 1 \end{vmatrix} = \begin{vmatrix} X & Y & Z & H \end{vmatrix}.$$

Обычные координаты  $(x, y, z, 1)$  преобразуются в нормальные координаты

$$x^* = \frac{x}{xp+yq+rz+1}, \quad y^* = \frac{y}{xp+yq+rz+1}, \quad z^* = \frac{z}{xp+yq+rz+1}.$$

Центры проекций находятся в точках  $x = -1/p$ ,  $y = -1/q$ , и  $z = -1/r$ . Результат трехточечного перспективного преобразования куба показан на Рис. 38.



**Рис. 38. Трехточечная перспектива куба**

## Формирование картинки

Конечным итогом всех преобразований является получение картинки в плоскости. Этот процесс состоит из трех основных этапов:

1. Формирование сцены;
2. Получение точки зрения (видовое преобразование);
3. Получение проекции на плоскость экрана.

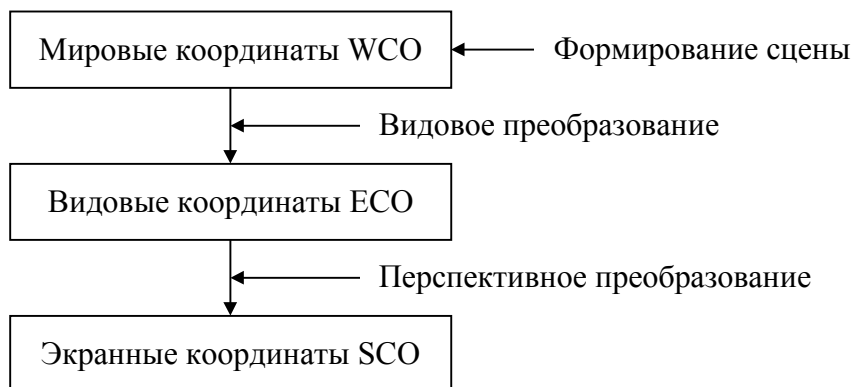
*Формирование сцены* — это размещение объектов в *мировой системе координат*. При этом объекты подвергаются аффинным преобразованиям — перемещению, повороту, изменению масштаба. Цель этого этапа — задать точки объектов в *мировой системе координат*.

*Получение точки зрения* — это проективное преобразование, которое разворачивает сцену так, чтобы получить определенное *направление визирования* (взгляда) на сцену. Иначе говоря, после формирования сцены нужно получить определенный вид на нее. При этом происходит преобразование, результатом которого являются *видовые координаты*.

*Получение проекции на плоскость экрана* — это проективное преобразование, результатом которого является проекция видовых координат на плоскость. Результирующие координаты называются *экранными*.

При получении проекции, как правило, выполняется перспективное преобразование или ортография.

Процесс получения картинки в целом отображен на Рис. 39.



**Рис. 39. Процесс получения картинки**

На практике любая часть процесса может отсутствовать при соблюдении условия, что объект(ы) задан(ы). Если, например, задан один объект и к нему не применено никакого преобразования, то при выводе изображения автоматически получится преобразование «чистая ортография Z». Попросту говоря, координата *Z* будет проигнорирована, и мы получим вид спереди. Если будет проигнорирована координата *X*, получим вид сбоку (слева или справа), при этом координата *Z* будет выводиться как координата *X*. Аналогично, если игнорировать координату *Y*, получим вид сверху (снизу), при этом координата *Z* будет выводиться как координата *Y*.

Если в процессе отсутствует видовое преобразование, получим перспективную картинку вида спереди или опять же чистую ортографию. Если отсутствует перспективное преобразование, получим вид на сцену без перспективы.

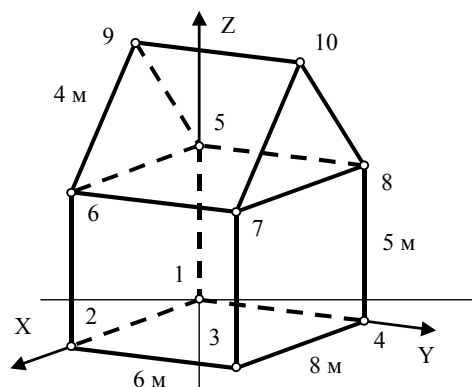
Рассмотренные случаи являются скорее исключениями. Как правило, требуется определенный вид на сцену, и желательно перспективный. Тогда для выполнения всех преобразований нужно иметь три матрицы координат, которые принято называть *мировыми* (*World*), *видовыми* (*Eye*) и *экранными* (*Screen*). Несмотря на то, что экранные координаты могут содержать третью координату (обычно *Z*), используются только две (*X* и *Y*).

Важным аспектом получения картинки является также подбор масштаба изображения. Объекты, предназначенные для вывода на экран, задаются в своих, реальных координатах. Например, для получения изображения железнодорожного моста длиной в 6 км понадобится задать точки, имеющие размер

ность порядка 1000, если измерять размеры в метрах. Экран, в который выводится изображение, как правило имеет ограничение на максимальную величину координат. Для лучших графических дисплеев максимальные значения составляют величину не более 2048, а практически эта величина имеет порядок 1024–1600. Следовательно, требуется преобразование масштаба, цель которого — приведение реальных координат к диапазону, в котором объект можно отобразить на устройстве вывода. Масштабирование можно выполнить на этапе формирования сцены и на этапе перспективного преобразования.

Видовое преобразование является одним из важнейших. Оно устанавливает точку зрения на сцену. Видовое преобразование может быть простым, например, изометрия или кабинетная проекция, но может быть и достаточно сложным, например, два-три поворота сцены и один-два переноса.

В качестве примера рассмотрим сцену, состоящую из утилизированного домика — параллелепипед и двухскатная крыша. Размер основания домика 8×6 м, высота стен 5 м, высота крыши 4 м. На Рис. 40 показано, как задан домик в мировой системе координат (кстати, для получения этого изображения было использовано видовое преобразование и интерфейс GREEN).



**Рис. 40. Сцена из одного объекта**

Сначала должна быть задана матрица мировых координат из 10-ти векторов положения (Рис. 41).

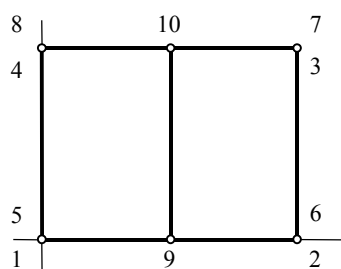
Точка	$X_w$	$Y_w$	$Z_w$
1	0	0	0
2	8	0	0
3	8	6	0
4	0	6	0
5	0	0	5
6	8	0	5
7	8	6	5
8	0	6	5
9	4	0	9
10	4	6	9

**Рис. 41. Матрица мировых координат точек домика**

Эту матрицу можно непосредственно использовать для вывода изображения, однако размеры объекта очень малы. Поэтому мы применяем масштабирование матрицы в 16 раз с получением новых мировых координат.

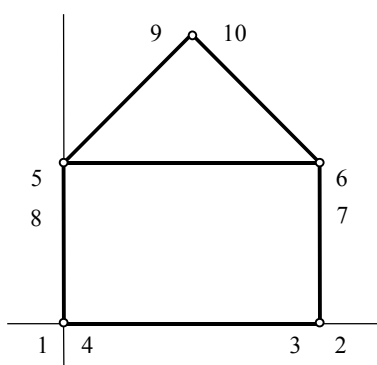
Для получения изображения нужно нарисовать отрезки прямых линий между точками 1-2, 2-3, 3-4 и т. д.

Вывод ребер с увеличенными координатами дает плоское изображение, показывающее вид со стороны оси Z (Рис. 42).



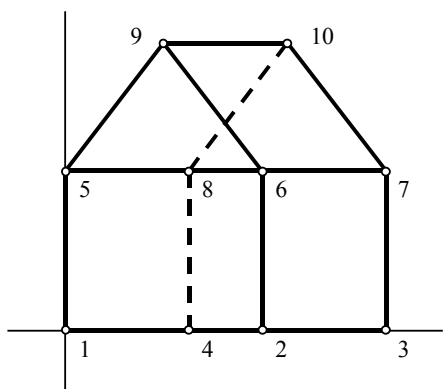
**Рис. 42. Проекция ORTHOZ домика**

Для получения пространственной картинки поворачиваем сцену вокруг оси X на угол  $-90^\circ$ . Получаем новые координаты, которые являются видовыми. Результат преобразования показан на Рис. 43.



**Рис. 43. Поворот домика вокруг оси X на  $-90^\circ$**

Далее выполняем еще два поворота. Первый вокруг оси Y на угол  $-40^\circ$ . Результат преобразования показан Рис. 44.

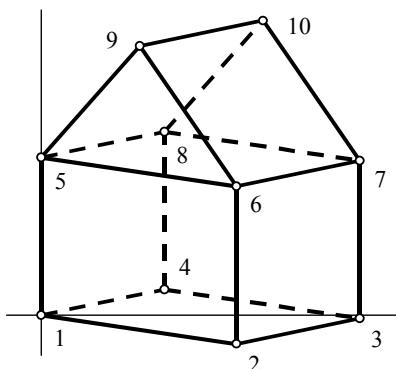


**Рис. 44. Поворот домика вокруг оси Y на  $-40^\circ$**

Второй поворот — вокруг оси X на угол  $10^\circ$ . Результат этого преобразования показан на Рис. 45.

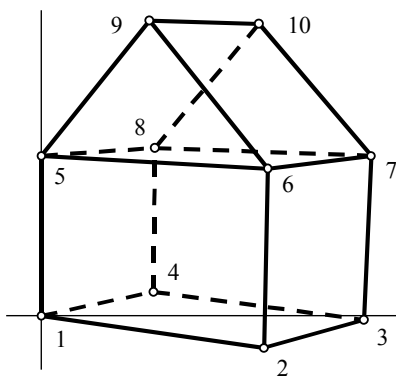
Таким образом мы получили трехмерное изображение и сформировали матрицу видовых координат. Можно было сделать еще несколько преобразований, как в матрице мировых координат, так и в матрице видовых. В мировых координатах можно было, например, сместить объект к центру сцены, или развернуть вокруг оси Z. В видовых координатах также можно выполнить перемещение.





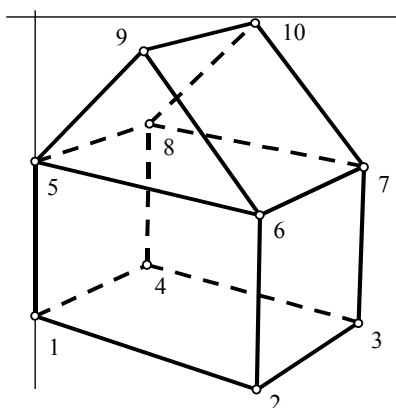
**Рис. 45. Второй поворот домика вокруг оси X на  $10^\circ$**

Далее для получения реалистичной картинку применим перспективное преобразование, например, простое перспективное преобразование (см. далее) или одноточечную перспективу вдоль оси Z. Результат запишем в матрицу экранных координат, которая используется для вывода окончательного изображения. Вот что получится, если применить одноточечное перспективное преобразование с  $r = -1/700$  (Рис. 46).



**Рис. 46. Одноточечная перспектива домика**

Наконец, для получения лучшего ракурса перед перспективным преобразованием дополнительно сместим объект на  $-150$  единиц вдоль оси Y. Окончательно получим картинку Рис. 47.

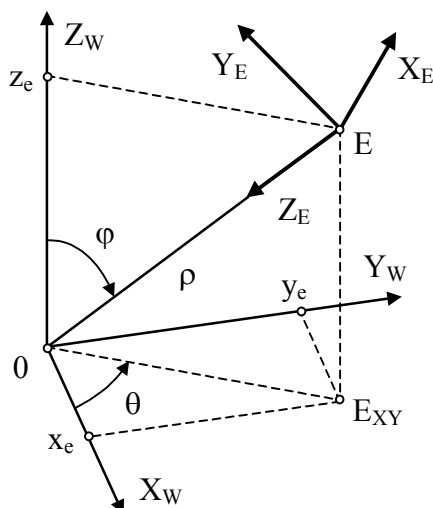


**Рис. 47. Окончательное изображение домика**

Как видим, процесс получения окончательного изображения представляет собой последовательность множества отдельных преобразований, каждое из которых имеет ясную направленность на достижение того или иного положения объектов сцены и точки зрения в пространстве.

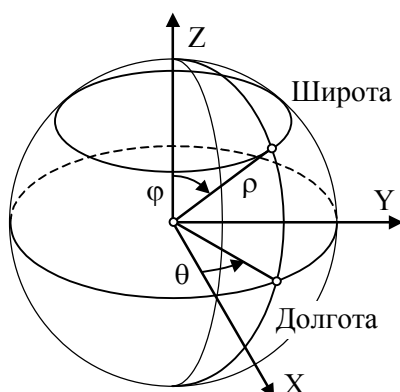
## Видовое преобразование

Получение определенного вида на сцену, как было показано ранее, является важной составной частью процесса получения картинки на плоскости. Будем рассматривать видовое преобразование как самостоятельное (несмотря на то, что оно состоит из множества отдельных). Целью видового преобразования поставим задание видовой системы координат так, чтобы ее центр находился бы в точке зрения, плоскость  $XY$  представляла бы собой плоскость проекции, а ось  $Z$  была бы направлена на центр мировой системы координат. Такое положение видовой системы координат показано на Рис. 48.



**Рис. 48. Видовая система координат**

Как следует из рисунка, видовая система координат при этом является левосторонней, однако плоскость проекции, если смотреть на нее в положительном направлении вдоль оси  $Z$ , отображает координатные оси  $X$  и  $Y$  в привычном для нас направлении. Учитывая, что система координат экрана имеет противоположное направление оси  $Y$ , имеет смысл рассматривать такое видовое преобразование, в котором бы направление осей  $X$  и  $Y$  совпадало с экранным. При этом видовая система стала бы правосторонней, что дает некоторые преимущества (например, в левосторонней системе координат вектор внешней нормали грани имеет противоположное направление).



**Рис. 49. Сферические координаты**

Рассмотрим процесс получения видовой системы координат. Он состоит из двух поворотов, отражений и переноса. Для задания видовой системы будем использовать сферические координаты точки зрения  $E$ . Напомним, что в

сферических координатах точка задается двумя углами и расстоянием. Углы задают долготу и широту точки, а расстояние — радиус сферы (Рис. 49).

На Рис. 48 угол  $\theta$  задает положение проекции вектора точки  $E$  в плоскости  $XY$ , угол  $\varphi$  задает отклонение вектора точки  $E$  от оси  $Z$ , расстояние  $\rho$  определяет положение точки  $E$  на выбранном направлении. При этом проекция вектора точки  $E$  в плоскости  $XY$  будет иметь длину  $\rho \cdot \sin \varphi$ , а координаты точки зрения определяются соотношениями

$$e_x = \rho \cdot \sin \varphi \cdot \cos \theta,$$

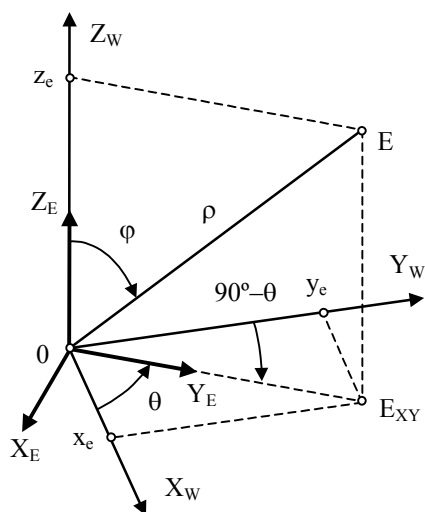
$$e_y = \rho \cdot \sin \varphi \cdot \sin \theta,$$

$$e_z = \rho \cdot \cos \varphi.$$

**Шаг 1.** Поворачиваем систему координат вокруг оси  $Z$  так, чтобы направление оси  $Y$  совпало с проекцией вектора  $E$  на плоскость  $XY$ . Угол поворота равен  $90^\circ - \theta$ , направление поворота положительное (если смотреть с положительного направления оси  $Z$ ). Матрица преобразования имеет вид:

$$R_Z = \begin{vmatrix} \sin \theta & \cos \theta & 0 \\ -\cos \theta & \sin \theta & 0 \\ 0 & 0 & 1 \end{vmatrix}.$$

Следует учитывать, что направления поворота принимается равным противоположным желаемому. Так, в данном случае ось  $Y$  поворачивается по часовой стрелке (в отрицательном направлении). Однако, если смотреть из видовой системы координат, объект поворачивается в обратную сторону. В результате преобразования получим положение видовой системы координат, показанное на Рис. 50.

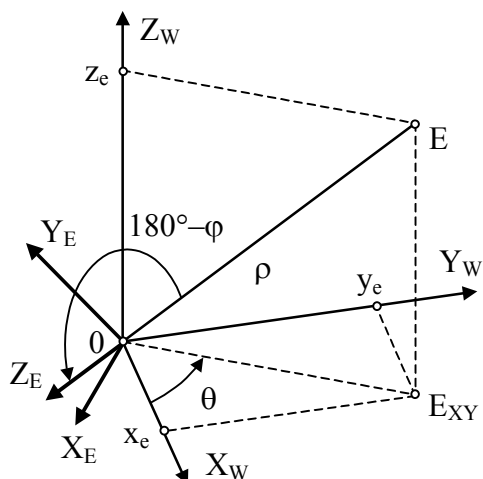


**Рис. 50. Поворот системы вокруг  $Z$**

**Шаг 2.** Поворачиваем полученную систему координат вокруг оси  $X$  так, чтобы направление оси  $Z$  совпало с направлением, противоположным направлению вектора точки зрения  $E$ . Угол поворота равен  $180^\circ - \varphi$ , направление поворота отрицательное. Матрица преобразования:

$$R_X = \begin{vmatrix} 1 & 0 & 0 \\ 0 & -\cos \varphi & -\sin \varphi \\ 0 & \sin \varphi & -\cos \varphi \end{vmatrix}.$$

Результат преобразования показан на Рис. 51.

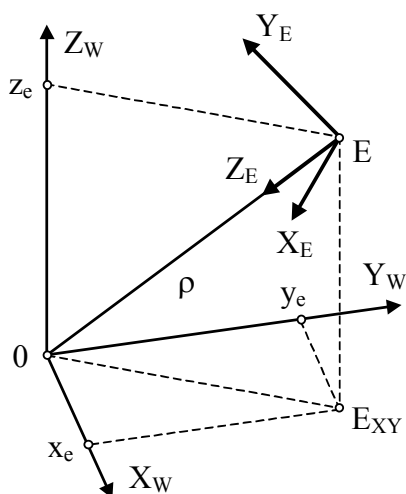


**Рис. 51. Поворот системы вокруг X**

**Шаг 3.** Перенос системы координат вдоль оси Z на расстояние ρ. Матрица преобразования имеет вид:

$$T = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & \rho & 1 \end{vmatrix}.$$

В результате преобразования система установится в точку E (Рис. 52).



**Рис. 52. Перенос системы в точку E**

**Шаг 4.** Осталось уточнить направление координатных осей X и Y. Для получения левосторонней видовой системы координат достаточно развернуть только ось X. Матрица преобразования имеет простой вид:

$$R_{YZ} = \begin{vmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix}.$$

Если нужно получить правостороннюю систему координат, следует выполнить два отражения одновременно. Матрица завершающего преобразования:

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

Суммируя все преобразования в одно, получим для левосторонней системы:

$$VP_L = \begin{vmatrix} -\sin \theta & -\cos \theta \cdot \cos \varphi & -\cos \theta \cdot \sin \varphi & 0 \\ \cos \theta & -\sin \theta \cdot \cos \varphi & -\sin \theta \cdot \sin \varphi & 0 \\ 0 & \sin \varphi & -\cos \varphi & 0 \\ 0 & 0 & \rho & 1 \end{vmatrix}.$$

Для получения правосторонней видовой системы координат матрица примет следующий вид:

$$VP_R = \begin{vmatrix} -\sin \theta & \cos \theta \cdot \cos \varphi & -\cos \theta \cdot \sin \varphi & 0 \\ \cos \theta & \sin \theta \cdot \cos \varphi & -\sin \theta \cdot \sin \varphi & 0 \\ 0 & -\sin \varphi & -\cos \varphi & 0 \\ 0 & 0 & \rho & 1 \end{vmatrix}.$$

При программировании данных преобразований следует учитывать количество операций. Если нужно получить конкретное видовое преобразование, коэффициенты матрицы могут быть рассчитаны заранее. Если же требуется непрерывное динамическое вычисление с произвольной точкой зрения, следует разложить преобразование на составляющие части и вычислять их отдельно. Так, для левостороннего преобразования получим последовательность операций:

1.  $xx = y \cdot \cos \theta - x \cdot \sin \theta$  (вместе с отражением оси X)
2.  $yy = x \cdot \cos \theta + y \cdot \sin \theta$
3.  $zz = z$
4.  $x = xx$
5.  $y = z \cdot \sin \varphi - y \cdot \cos \varphi$
6.  $z = \rho - y \cdot \sin \varphi - z \cdot \cos \varphi$

Число операций в этом случае 13 (из них умножений 8), а если использовать обобщенную матрицу, то число операций равно 17 (умножений 12). Для правосторонней системы нужно изменить знак результата в пятой операции.

Таким образом, рассмотренное видовое преобразование устанавливает соотношение между сферическими координатами точки зрения E и положением произвольной точки в видовой системе координат. Подобное преобразование используется в существующих интерактивных графических системах, таких, как, например, система *AutoCAD* фирмы *Autodesk*.

Выбор углов  $\theta$  и  $\varphi$  для задания точки зрения не представляет трудности. Интересен вопрос выбора расстояния  $\rho$  от центра мировой системы координат до точки зрения E. В следующем разделе мы попытаемся прояснить его. Пока же можно заметить, что если видовое преобразование является последним в процессе получения картинки, и после него не используется никакое другое, величина расстояния  $\rho$  не имеет никакого значения, да и координата  $z$  преобразуемых точек тоже. В этих случаях ее можно просто не вычислять.

## Простое перспективное преобразование

Для получения более реалистичной картинки после видового преобразования обычно используется перспективное. Рассмотрим простое перспективное преобразование с учетом видовой системы координат.

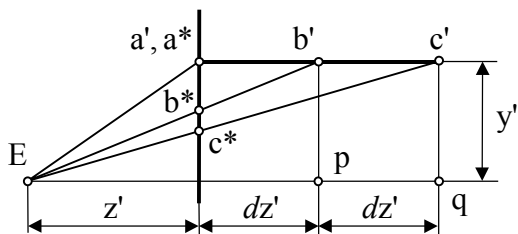
На Рис. 53 показан объект в мировой системе координат, расположенный так, что две его точки 1 и 2 имеют одну и ту же высоту  $y'$  и разные расстояния  $z'_1$  и  $z'_2$  до точки зрения E, в которой находится центр видовой системы

Экран, или иначе — плоскость проекции, в которой будут находиться экран-ные координаты, расположен на расстоянии  $d$  от точки зрения. Проекции вы-соты  $y'$  на экран составляют  $y_1^*$  и  $y_2^*$  для точек 1 и 2 соответственно. Из по-добия треугольников следует, что

Рассматривая картинку в другой плоскости, совершенно аналогично можно получить соотношения для координат  $x$ :

Полученные соотношения для  $x^*$  и  $y^*$  являются формулами простого перспективного преобразования (штрихом обозначены видовые координаты):

Как было показано в разделе «Перспективные преобразования», правильное соотношение между проекциями равных длин не зависит от расстояния  $d$  до плоскости проекции. Рассмотрим теперь, как проецируются отрезки, параллельные направлению визирования, то есть параллельные оси  $Z$  видовой системы координат (Рис. 54).



На Рис. 54 показаны два отрезка  $a'b'$  и  $b'c'$ , имеющие высоту  $y'$ , проецируемые из точки зрения  $E$ . Длины отрезков одинаковы и равны  $dz'$ . Проекции отрезков на плоскость экрана равны  $a^*b^*$  и  $b^*c^*$  соответственно. Как следует из формул перспективного преобразования, на величины проекций влияет расстояние от точки зрения до преобразуемой точки, то есть расстояние  $z'$ .

Заметим, что расстояние  $z'$  зависит от  $\rho$ . Действительно, координата  $z'$  точки определяется в видовом преобразовании по формуле

$$z' = \rho - x \cdot \cos \theta \cdot \sin \varphi - y \cdot \sin \theta \cdot \sin \varphi - z \cdot \cos \varphi.$$

Обозначая часть формулы после  $\rho$  через  $z$ , где  $z$  — координата  $z$  точки в мировой системе координат, получим:

$$z' = \rho - z.$$

Найдем соотношение между проекциями, а именно, во сколько раз проекция  $a^*b^*$  больше проекции  $b^*c^*$ . Иначе говоря, выясним, как влияет расстояние  $\rho$  до точки зрения на проектирование глубины. Соотношение  $\gamma = a^*b^*/b^*c^*$  назовем «величиной перспективности».

Проекцию  $a^*b^*$  найдем из подобия треугольников  $b'a^*b^*$  и  $E\rho b'$ :

$$a^*b^* = y' \frac{dz'}{z' + dz'}.$$

Проекцию  $a^*c^*$  найдем из подобия треугольников  $a's^*c^*$  и  $Eqs'$ :

$$b^*c^* + a^*b^* = y' \frac{2dz'}{z' + 2dz'}.$$

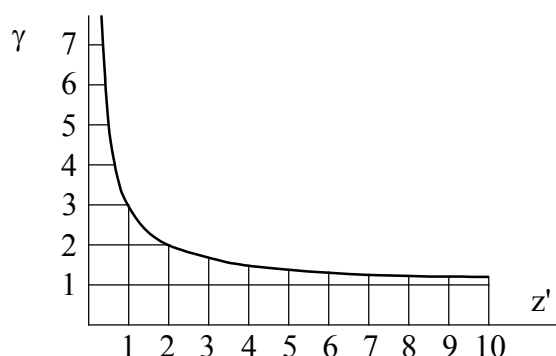
Перенеся  $a^*b^*$  в правую часть и заменив найденным для  $a^*b^*$  выражением, получим

$$b^*c^* = y' \frac{dz' \cdot z'}{(z' + 2dz')(z' + dz')}.$$

Поделив одно значение на другое, получим выражение для «величины перспективности»  $\gamma$ :

$$\gamma = 1 + \frac{2dz'}{z'}.$$

Для оценки полученного результата построим график зависимости  $\gamma = 1 + 2/z'$ :



**Рис. 55. Зависимость  $\gamma = 1 + 2/z'$**

Как следует из графика на Рис. 55, при значениях  $z'$ , меньших 2-3, проекция глубины резко возрастает, что приводит к искажению (гиперболизации) перспективности картинки.

Сказанное иллюстрирует пример перспективного проецирования домика с трех разных точек зрения. На Рис. 56 приведены три перспективные проекции, выполненные со значениями  $\rho$ , равными 400 (слева), 260 (в центре) и 190 (справа). Видно, что среднее изображение начинает приобретать гиперболизированную перспективу. Координата  $z'$  наиболее выступающей точки 7 равна 222 и 82 соответственно для левого и среднего изображений.

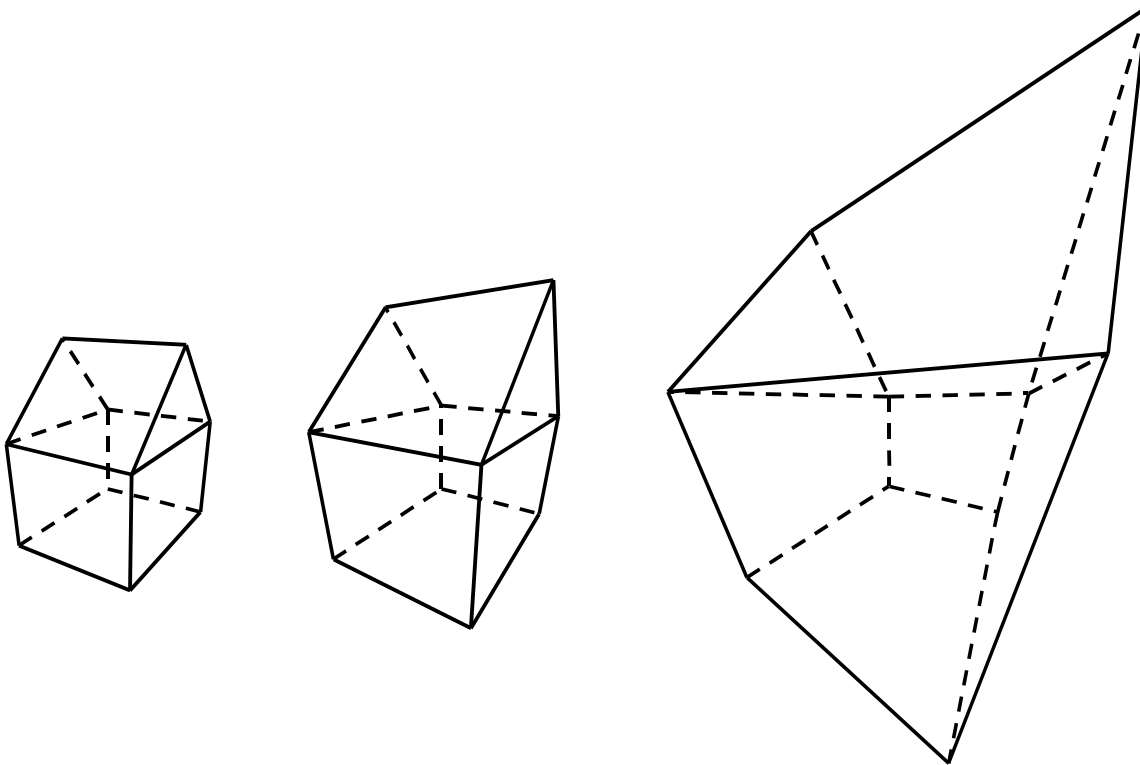


Рис. 56. Точки зрения 400 (слева), 260 (в центре) и 190 (справа)

Дальнейшее уменьшение  $\rho$  приведет к искажению изображения до неузнаваемости (Рис. 56, справа,  $z' = 12$ ).

Таким образом, при получении перспективного изображения следует учитывать, что правильная перспектива достигается, если объект достаточно удален от точки зрения. Кроме того, подставляя выражение  $z' = \rho - z$  в формулу перспективного преобразования, получим

$$y^* = y' \frac{d}{\rho - z}.$$

Это означает, что существует такое соотношение параметров видового преобразования, при котором выполнить перспективное преобразование *невозможно*, так как в знаменателе получается ноль, если  $z = \rho$ . Этого, впрочем, не случится, если придерживаться первого правила (удаления объекта от точки зрения).

### **Автостабилизация размера картинки**

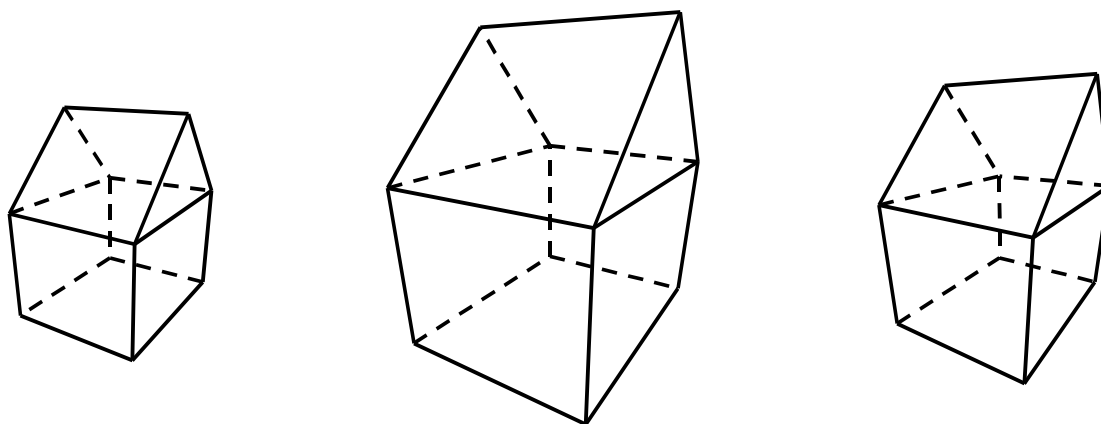
Картинка, которая получается при перспективном преобразовании, зависит от положения точки зрения  $\rho$ , и от положение экрана  $d$ :

$$y^* = y' \frac{d}{\rho - z}.$$

Эти расстояния по-разному влияют на результирующее изображение. Положение экрана  $d$  влияет только на масштаб изображения. Расстояние до точки зрения  $\rho$  влияет как на масштаб изображения, так и на «величину перспективности». В целом размер изображения определяют два фактора, что создает неудобства при динамическом выводе. Управление «величиной перспективности» при помощи положения точки зрения будет вызывать изменение масштаба изображения. Это потребует изменение положения экрана для ста



билизации размера картинке. На Рис. 57 слева и в центре приведены изображения домика при одинаковом положении экрана  $d=200$ , и разных положениях точки зрения (слева  $\rho=400$ , в центре  $\rho=300$ ). Видно, что среднее изображение значительно увеличено.



**Рис. 57. Изменение «величины перспективности» разными способами**

Есть, однако, способ стабилизировать размер изображения при изменении перспективности. Достаточно совместить положение экрана с центром мировой системы координат, то есть принять  $d=\rho$ . При этом проецируемые точки всегда будут находиться «вблизи» экрана.

На Рис. 57 справа показано изображение домика с той же величиной перспективности, что и в центре, при этом  $d=\rho=300$ . Размер изображения практически не изменился.

Заметим, что положение экрана влияет на масштаб изображения. Так как использование одного значения для  $d$  и  $\rho$  не позволяет управлять масштабом, можно изменить формулу перспективного преобразования, добавив множитель масштабирования  $s$ :

$$y^* = y' \frac{d \cdot s}{z'}.$$

Для координаты  $x$  получим аналогичную формулу

$$x^* = x' \frac{d \cdot s}{z'}.$$

## Контрольные вопросы и задания

1. Чем отличаются проективные преобразования от геометрических?
2. Что называется картинной плоскостью?
3. Что называется проекцией?
4. Через какие точки сцены проходят проектирующие лучи?
5. Как, по какому признаку (признакам), классифицируются параллельные проекции?
6. В чем разница между изометрической и свободной проекцией?
7. В чем состоит цель видового преобразования?
8. Что называется точкой схода перспективного преобразования?
9. От чего зависит величина перспективности картинке?
10. Вычислите в уме матрицу видового преобразования для  $\theta=0$ ,  $\varphi=0$ .

# Растровые алгоритмы

## Алгоритм Брезенхейма

Задача рисования отрезка прямой линии на растровом устройстве является одной из наиболее важных в компьютерной графике. Практически любое графическое изображение есть сумма отрезков и строится последовательным или параллельным их вычерчиванием. От скорости, с которой система рисует отрезки, зависит общая скорость графического вывода.

В 1965 году Брезенхейм (*Bresenham*), занимавшийся программированием вывода графики при помощи планшетного графопостроителя, разработал алгоритм, предназначенный для расчета растрового представления отрезка с использованием только целых чисел.

Рассмотрим, как строится отрезок прямой линии с использованием ее математического описания. Отрезок — это часть прямой линии, заданная начальной и конечной точками. На рисунке Рис. 58 эти точки обозначены буквами P и Q и имеют координаты  $(x_1, y_1)$  и  $(x_2, y_2)$  соответственно.

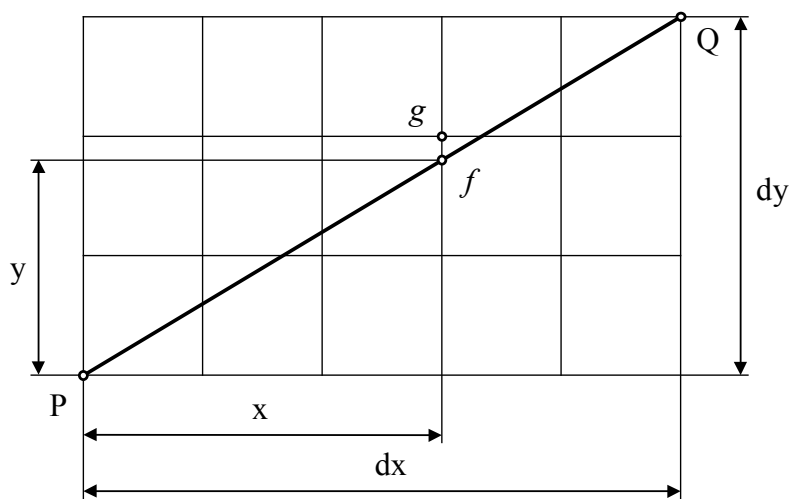


Рис. 58. Расчет точки прямой

Если принять, что начальная точка отрезка находится в начале системы координат, то координата  $y$  произвольной точки отрезка с координатой  $x$  может быть вычислена из подобия треугольников:

$$\begin{aligned} y/x &= dy/dx; \\ y &= x \cdot dy/dx. \end{aligned}$$

Обозначив отношение разностей координат через  $k$ ,

$$k = dy/dx = (y_2 - y_1)/(x_2 - x_1),$$

получим простейшее уравнение прямой:

$$y = x \cdot k.$$

Если начальная точка отрезка расположена произвольно, все координаты получают смещение  $x_1$  и  $y_1$  и уравнение прямой принимает вид

$$y = k(x - x_1) + y_1.$$

Для непосредственного построения отрезка при помощи данного уравнения можно использовать следующий алгоритм:

Процедура ОТРЕЗОК1(Цел  $X_1$ ,  $Y_1$ ,  $X_2$ ,  $Y_2$ )

Цел  $DX$ ,  $DY$ ,  $X$ ,  $Y$

Вещ  $K$

$DX = X_2 - X_1$

$DY = Y_2 - Y_1$

$K = DY / DX$

Изменяя  $X = X_1$  До  $X_2$

$Y = K * (X - X_1) + Y_1$

Точка ( $X$ ,  $Y$ )

Следующее  $X$

Конец процедуры

Рис. 59. Алгоритм рисования отрезка

Предполагается, что процедура «Точка» рисует точку на устройстве вывода. Значение координаты  $y$ , рассчитанное при помощи формулы, дает математически точное положение точки на отрезке, обозначенное на рисунке как точка  $f$ . Это вещественное значение должно быть округлено до ближайшего целого значения (на рисунке до точки  $g$ ), так как координаты задаются целыми числами. Данный алгоритм позволяет рисовать отрезки, но с малой скоростью из-за вычислений с плавающей точкой.

Отрезок рисуется слева направо с шагом вдоль оси  $X$ , равным 1. Обратите внимание, что при переходе к следующему значению  $x$  значение  $y$  изменяется максимум на 1 единицу, причем всегда в одну и ту же сторону — либо в сторону увеличения, либо в сторону уменьшения.

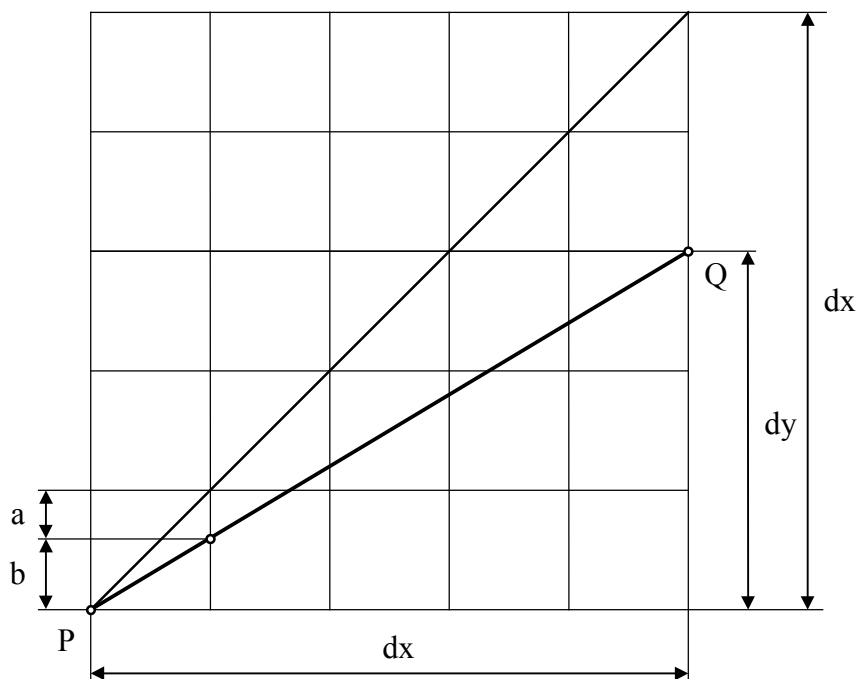


Рис. 60. Вычисление первой точки

Таким образом, процесс рисования отрезка можно представить следующим образом. Начальная (нулевая) точка отрезка выводится в позицию, определяемую ее координатами ( $x_1$ ,  $y_1$ ). Текущее значение координаты  $y$  принимается равным  $y_1$ . Далее последовательно переходим от одного целого значения  $x$  к другому и принимаем решение о том, остается текущее значение  $y$  преж

ним, или изменяется на 1. Рассмотрим процесс принятия решения о том, какое значение  $y$  следует выбрать при рисовании первой точки (Рис. 60).

Решение будем принимать, сравнивая значения  $b$  и  $a$ , вычитая их одно из другого. Если  $b - a > 0$ , следующая точка рисуется на 1 выше, иначе на той же высоте. Значения  $b$  и  $a$  найдем из подобия треугольников:

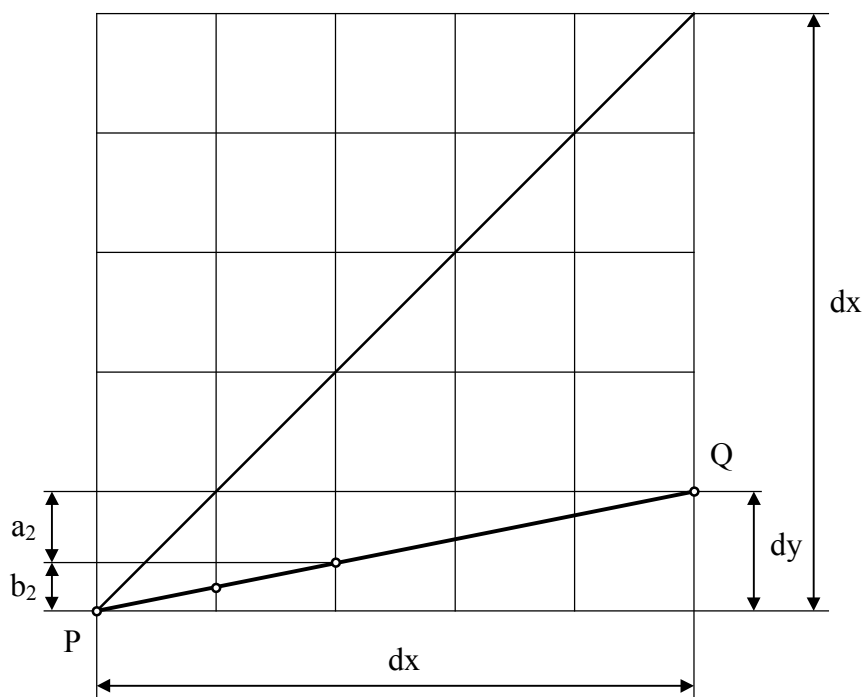
$$\begin{aligned} b/1 &= dy/dx; \\ b &= dy/dx; \\ a/1 &= (dx - dy)/dx; \\ a &= (dx - dy)/dx; \end{aligned}$$

Для обозначения разности  $b - a$  введем обозначение  $d$ :

$$d = b - a = dy/dx - (dx - dy)/dx = (2dy - dx)/dx.$$

Если  $d > 0$ , то значение  $y$  изменяется, иначе остается неизменным.

При переходе к следующей точке следует снова вычислить величину  $d$ . Вместо этого рассмотрим, как изменяется  $d$ , на какое значение. Сначала рассмотрим случай, когда значение  $y$  не изменяется (Рис. 61).



**Рис. 61. Вычисление второй точки при неизменном  $y$**

Из подобия треугольников следует:

$$\begin{aligned} b_2/2 &= dy/dx; \\ b_2 &= 2dy/dx; \\ 1 + a_2/2 &= (dx - dy)/dx; \\ a_2 &= 1 - 2dy/dx; \end{aligned}$$

Найдем разность  $d_2$ :

$$d_2 = b_2 - a_2 = 4dy/dx - 1.$$

В более общем случае для  $n$ -ой точки можно найти:

$$\begin{aligned} b_n &= ndy/dx; \\ a_n &= 1 - ndy/dx; \\ d_n &= 2ndy/dx - 1. \end{aligned}$$

Вычитая  $d_{n-1}$  из  $d_n$  и обозначая разность через  $d1$ , получим

$$d1 = 2ndy/dx - 1 - 2(n-1)dy/dx + 1 = 2dy/dx.$$

Следовательно, если при переходе к следующей точке положение  $y$  не изменится, то значение  $d$  изменяется на величину  $d1 = 2dy/dx$ .

Рассмотрим теперь, на какую величину изменится  $d$  в случае, если значение  $y$  изменилось (Рис. 62).

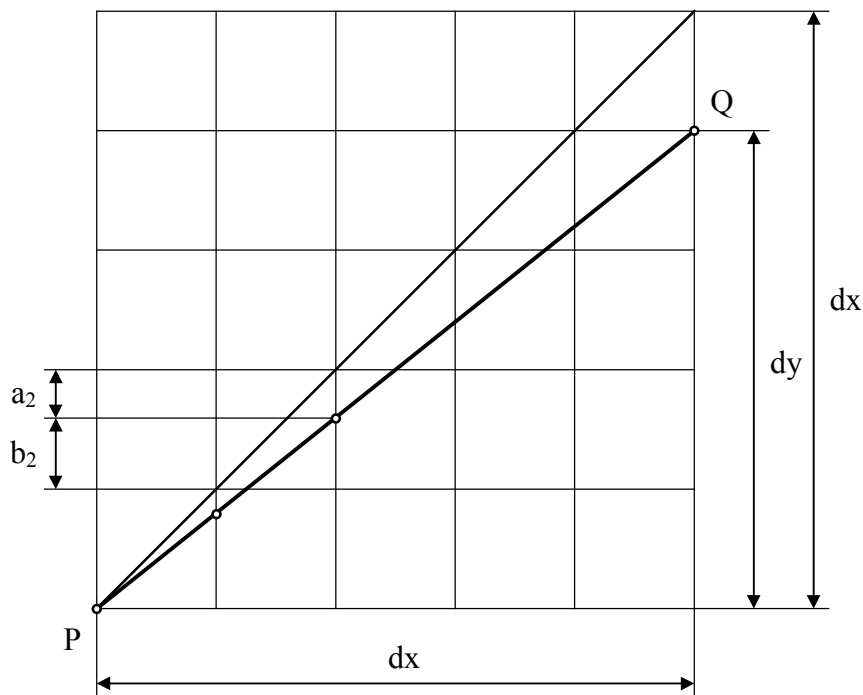


Рис. 62. Вычисление второй точки при изменении  $y$

Из подобия треугольников можно найти:

$$(b_n - n + 1)/n = dy/dx;$$

$$b_n = 2ndy/dx - 2n + 1;$$

$$a_n/n = (dx - dy)/dx;$$

$$a_n = n(dx - dy)/dx;$$

Найдем разность  $b_n - a_n$ :

$$d_n = b_n - a_n = 2ndy/dx - 2dy/dx - 1.$$

Вычитая  $d_{n-1}$  из  $d_n$  и обозначая разность через  $d2$ , получим

$$d2 = 2ndy/dx - 2dy/dx - 1 - 2(n-1)dy/dx + 2dy/dx + 1 = 2(dy - dx)/dx.$$

Значения  $d$ ,  $d1$  и  $d2$  являются вещественными числами, но все они имеют в знаменателе одно и то же значение  $dx$ . Умножив все три значения на  $dx$ , мы избавимся от знаменателя и перейдем к целым числам. Теперь принятие решения о том, какое значение  $y$  принять для данной точки, будем принимать на основе значения  $d = (b - a)/dx$ .

Полученные соотношения легли в основу алгоритма Брезенхейма, который приведен ниже на Рис. 63. Заметим, что для ускорения вычислений значения величин  $d$ ,  $d1$  и  $d2$  вычисляются сложениями и вычитаниями.

Недостаток данных алгоритмов заключается в том, что они правильно рисуют только такие отрезки, которые имеют угол наклона до  $45^\circ$  к оси  $X$ . Действительно, если отрезок имеет больший наклон, часть точек по вертикали будет

пропущена, так как число точек по горизонтали меньше числа точек по вертикали (Рис. 64). Отрезок получается несвязанным (несвязным).

**Процедура ОТРЕЗОК2 (Цел  $X_1, Y_1, X_2, Y_2$ )**

Цел  $DX, DY, X, Y, D, D1, D2$

$DX = X_2 - X_1$

$DY = Y_2 - Y_1$

$D1 = DY + DY$

$D = D1 - DX$

$D2 = D - DX$

$X = X_1$

$Y = Y_1$

Точка ( $X, Y$ )

Изменяя  $X = X_1 + 1$  До  $X_2$

Если  $D > 0$  То

$D = D + D2$

$Y = Y + 1$

Иначе

$D = D + D1$

Конец Если

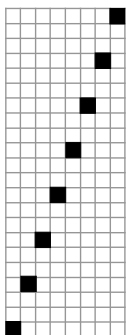
Точка ( $X, Y$ )

Следующее  $X$

Конец процедуры

**Рис. 63. Алгоритм Брезенхейма**

Кроме того, алгоритм рисует отрезки только в одну сторону, от меньших значений  $x$  и  $y$  в сторону больших. Первый недостаток устраняется делением алгоритма на две части. Первая часть рисует отрезки, угол наклона которых меньше  $45^\circ$ , вторая часть — отрезки, угол наклона которых больше. Чтобы определить, какая часть должна работать, используется сравнение разностей координат концов отрезка  $dx$  и  $dy$ . Та часть алгоритма, которая рисует отрезки с большим наклоном, работает точно так же, как и описанный выше алгоритм, только движение идет вдоль оси  $Y$ , а не вдоль оси  $X$ . Для получения второй части алгоритма нужно заменить  $X$  на  $Y$  во всех формулах.



**Рис. 64. Пропуск точек при большом угле наклона прямой**

Второй недостаток устраняется введением специальных переменных, которые указывают направление движения (в алгоритме они обозначены как  $SX$  и  $SY$ ). Полный алгоритм Брезенхейма приведен на Рис. 65:

Главное преимущество алгоритма заключается в его скорости. Все переменные имеют целый тип, а целочисленные вычисления производятся гораздо

быстрее вычислений с плавающей точкой. Кроме того, в алгоритме нет умножений или делений.

```
Процедура ОТРЕЗОК(Цел X1, Y1, X2, Y2)
    Цел DX, DY, X, Y, D, D1, D2, SX, SY
    DX=Abs(X2-X1)
    DY=Abs(Y2-Y1)
    Если X1<X2 То SX=1 Иначе SX=-1
    Если Y1<Y2 То SY=1 Иначе SY=-1
    X=X1
    Y=Y1
    Точка(X, Y)
    Если DY>DX То -- движение по вертикали Y
        D1=DX+DX
        D=D1-DY
        D2=D-DY
        Изменяя Y=Y1+1 До Y2 Шаг SY
            Если D>0 То
                D=D+D2
                X=X+SX
            Иначе
                D=D+D1
        Конец Если
        Точка(X, Y)
        Следующее Y
    Иначе -- движение по горизонтали X
        D1=DY+DY
        D=D1-DX
        D2=D-DX
        Изменяя X=X1+1 До X2 Шаг SX
            Если D>0 То
                D=D+D2
                Y=Y+SY
            Иначе
                D=D+D1
        Конец Если
        Точка(X, Y)
        Следующее X
    Конец Если
Конец процедуры
```

**Рис. 65. Полный алгоритм Брезенхейма**

В заключение отметим, что на основе алгоритма Брезенхейма разработаны многочисленные алгоритмы для рисования дуг, окружностей, эллипсов, гипербол и т. п. Кроме этого, он идеально подходит для некоторых случаев целочисленной интерполяции.

## Алгоритм отсечения отрезка (Сазерленда-Кохена)

Часто необходимо отсечь выводимое изображение границами некоторой прямоугольной области (области вывода). Для решения этой задачи в отношении отрезков существует следующий простой алгоритм.

Плоскость построения разбивается на 9 частей линиями, проходящими через границы области вывода (Рис. 66). Каждой части плоскости вокруг области отсечения присваивается код, соответствующий весу одного из двоичных разрядов. Каждая часть представляет собой множество точек, расположенных одинаково по отношению к области вывода. Тогда вес двоичного разряда, выбранный для данной части, указывает на положение точки относительно области отсечения.

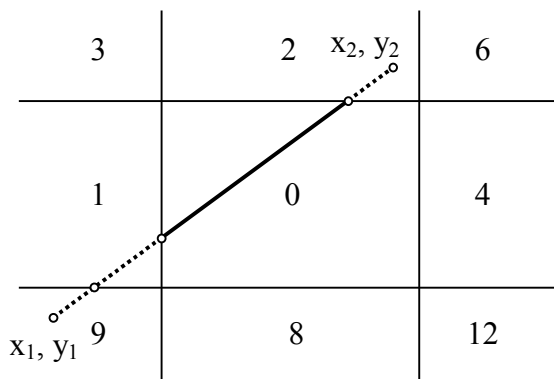


Рис. 66. 8 областей отсечения

Например, для всех точек, которые лежат левее области отсечения, присвоен код 1, означающий, что для этой области используется бит 0. Для всех точек, которые лежат выше области отсечения, присвоен код 2, соответствующий биту 1. Точкам правее области отсечения присвоен код 4 (бит 3), а точкам ниже — код 8 (бит 4).

Для концевых точек отрезка определяются коды частей, в которые они попадают. Это выполняется при помощи вспомогательного алгоритма *КодОбласти*, приведенного на Рис. 67.

Цел Функция КодОбласти(Цел X, Y, AX1, AY1, AX2, AY2)

Если  $(X < X1)$  То КодОбласти = 1

Если  $(Y < Y1)$  То КодОбласти = КодОбласти Или 2

Если  $(X > X2)$  То КодОбласти = КодОбласти Или 4

Если  $(Y > Y2)$  То КодОбласти = КодОбласти Или 8

Конец функции

Рис. 67. Алгоритм определения области отсечения

Параметрами функции являются координаты проверяемой точки  $(X, Y)$ , и две точки, заданные координатами  $(AX1, AY1)$  и  $(AX2, AY2)$ , указывающими на диагональные углы области отсечения. Эта функция устанавливает один или два бита в возвращаемом значении, сочетание которых указывает на одну из восьми частей плоскости.

Далее проверяется, лежат точки внутри или снаружи области отсечения. Если оба кода равны нулю, то концы отрезка находятся внутри области отсечения. Если коды содержат одинаковые биты, концы отрезка находятся вне области отсечения.

Далее выполняется цикл из максимум 4-х итераций. В каждой из итераций происходит отсечение части отрезка одной из граничных прямых. Сначала проверяется первая точка отрезка. После отсечения отрезка со стороны первой точки, концевые точки и их коды меняются местами и выполняется еще максимум две итерации. Алгоритм процедуры отсечения приведен на Рис. 68.



В ней параметрами являются координаты первой и второй точек отрезка и координаты двух угловых точек области отсеечения.

Процедура *Отсечение* (Цел  $X1, Y1, X2, Y2, AX1, AY1, AX2, AY2$ )

```

Цел Code1, Code2
Лог Inside, Outside
Code1 = Region(X1, Y1, AX1, AY1, AX2, AY2)
Code2 = Region(X2, Y2, AX1, AY1, AX2, AY2)
Inside = (Code1 Или Code2) = 0
Outside = (Code1 И Code2) <> 0
Цикл Пока (Не Inside) И (Не Outside)
    Если (Code1=0) То
        Обмен (X1, X2)
        Обмен (Y1, Y2)
        Обмен (Code1, Code2)
    Конец Если
    Если (Code1 И 1) То -- отсечение слева
         $Y1 = Y1 + (Y2 - Y1) / (X2 - X1) * (AX1 - X1)$ 
         $X1 = AX1$ 
    Иначе Если (Code1 И 2) То -- отсечение сверху
         $X1 = X1 + (X2 - X1) / (Y2 - Y1) * (AY1 - Y1)$ 
         $Y1 = AY1$ 
    Иначе Если (Code1 И 4) То -- отсечение справа
         $Y1 = Y1 + (Y2 - Y1) / (X2 - X1) * (AX2 - X1)$ 
         $X1 = AX2$ 
    Иначе Если (Code1 И 8) То -- отсечение снизу
         $X1 = X1 + (X2 - X1) / (Y2 - Y1) * (AY2 - Y1)$ 
         $Y1 = AY2$ 
    Конец Если
    Code1 = Region(X1, Y1, AX1, AY1, AX2, AY2)
    Code2 = Region(X2, Y2, AX1, AY1, AX2, AY2)
    Inside = (Code1 Или Code2) = 0
    Outside = (Code1 И Code2) <> 0
Конец Цикла
Отрезок (X1, Y1, X2, Y2)
Конец Процедуры

```

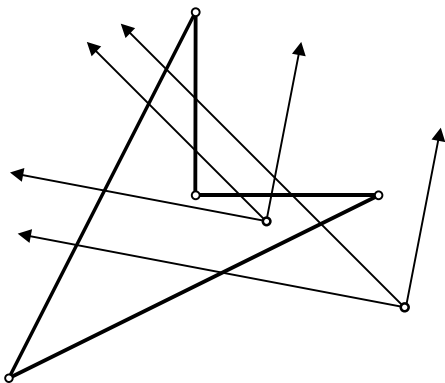
Рис. 68. Алгоритм отсеечения отрезка

Для работы данного алгоритма требуется также вспомогательный алгоритм *Обмен* для обмена двух целых значений. Процедура *Отрезок* рисует отрезок.

## Определение принадлежности точки многоугольнику

Задача определения принадлежности точки многоугольнику в компьютерной графике встречается достаточно часто, чтобы выделить ее в самостоятельную. Пусть задан произвольный многоугольник при помощи вершин, записанных в массив (матрицу)  $P$ . Требуется определить, находится ли произвольная точка  $(X, Y)$  внутри многоугольника.

Если выпустить из точки произвольный луч и подсчитать число пересечений луча с ребрами многоугольника, то точка лежит внутри, если число пересечений нечетно, и снаружи в противном случае (Рис. 69). Для простоты будем выпускать луч не в произвольном направлении, а горизонтально и вправо.



**Рис. 69. Тестирование положения точки произвольным лучом**

Особые случаи возникают, когда луч проходит через вершину многоугольника. Если луч проходит через вершину, когда оба сопряженных ребра расположены с одной стороны луча, пересечение не засчитывается (Рис. 70, слева). Когда ребра расположены по разные стороны луча, напротив, пересечение должно засчитываться (Рис. 70, справа). Пересечение засчитывается, если луч проходит через верхнюю вершину (Рис. 70, в центре и справа).



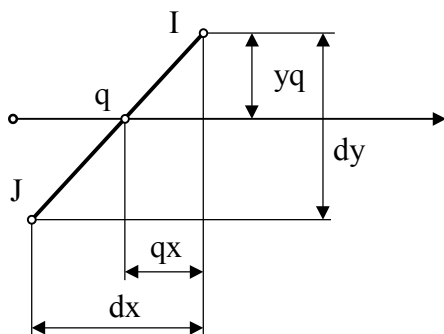
**Рис. 70. Луч проходит через вершину**

Если луч проходит через ребро (Рис. 71), пересечение не засчитывается.



**Рис. 71. Луч проходит через ребро**

Алгоритм представляет собой цикл с числом итераций, равным количеству ребер (вершин). В каждой итерации проверяется одно ребро. Сначала проверяется, что ребро не является горизонтальным. Далее проверяется, что ребро лежит не выше и не ниже линии луча. Если эти проверки пройдены успешно, линия луча пересекает ребро (Рис. 72).



**Рис. 72. Луч пересекает ребро**

Далее определяем отношение высот  $t=yq/dy$ . Оно показывает, на каком уровне находится линия луча по отношению к точке I. Это позволяет определить координату X точки пересечения линии луча с ребром. Если эта координата меньше, чем координата X проверяемой точки, луч не пересекает ребро. Осталось убедиться, что луч не проходит через нижнюю точку ребра.

Данный алгоритм определения принадлежности точки многоугольнику приведен на Рис. 73.

```

Лог Функция ТочкаВнутри(Цел X, Y, Матрица P)
  Цел Count, I, J
  Вещ T, XQ
  Count = 0
  Изменяя I От 1 До P.Размер
    J = (I Mod N) + 1
    Если (P(I).Y <> P(J).Y) То
      Если Не ((P(I).Y > Y) И (P(J).Y > Y)) То
        Если Не ((P(I).Y < Y) And (P(J).Y < Y)) То
          T = (Y - P(I).Y) / (P(J).Y - P(I).Y)
          XQ = P(I).X + T * (P(J).X - P(I).X)
          Если (XQ >= X) Then
            Если Не (Min(P(I).Y, P(J).Y) = Y) То
              Count = Count + 1
            Конец Если
          Конец Если
        Конец Если
      Конец Если
    Конец Если
  Следующее I
  ТочкаВнутри = Лог(Count И 1)
Конец функции

```

Рис. 73. Алгоритм определения принадлежности точки многоугольнику

## Закраска области

Для закраски области, ограниченной пикселями одного цвета (перекраски пикселей) часто используется следующий алгоритм. Пусть часть области рисования имеет один и тот же цвет. Границы этой области задаются переходами цвета, например, контуром фигуры (Рис. 74).

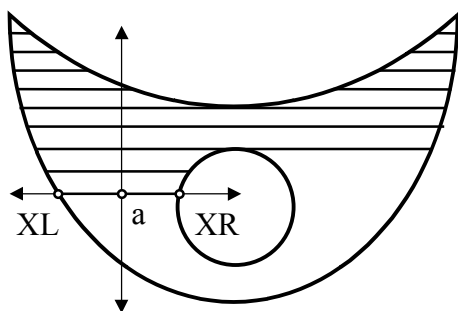


Рис. 74. Закраска области

Пусть задана точка a, лежащая внутри области. Перемещаясь от этой точки по горизонтали влево, найдем координату XL левой точки линии закраски.

Далее перемещаясь от исходной точки вправо, найдем координату XR правой точки линии закрашки. Теперь линию можно закрасить.

Далее, проходя по каждой точке закрашенной линии, ищем незакрашенные точки выше текущей линии. Если такие точки находятся, функция закрашки вызывается для них рекурсивно. После этого точно так же проверяются точки, лежащие ниже текущей линии. Этот алгоритм обнаруживает области, которые не нужно закрашивать внутри данной (дырки).

**Цел Функция Закраска(Цел X, Y, Dir, PredXL, PredXR)**

Цел XL, XR, C

XL = X

XR = X

Цикл

XL = XL - 1

C = Цвет\_точки(XL, Y)

Пока (C = Color1)

Цикл

XR = XR + 1

C = Цвет\_точки(XR, Y)

Пока (C = Color1)

XL = XL + 1

XR = XR - 1

Отрезок (XL, Y)-(XR + 1, Y), Color2

Изменяя X От XL До XR

C = Цвет\_точки(X, Y + Dir)

If (C = Color1) Then

X = Закраска(X, Y + Dir, Dir, XL, XR)

End If

Следующее X

Изменяя X От XL До PredXL - 1

C = Цвет\_точки(X, Y - Dir)

If (C = Color1) Then

X = Закраска(X, Y - Dir, -Dir, XL, XR)

End If

Следующее X

Изменяя X От PredXR До XR - 1

C = Цвет\_точки(X, Y - Dir)

If (C = Color1) Then

X = Закраска(X, Y - Dir, -Dir, XL, XR)

End If

Следующее X

Закраска = XR

End Function

**Рис. 75. Алгоритм закрашки области**

В алгоритме на Рис. 75 переменная Color1 обозначает цвет закрашиваемой (перекрашиваемой) области, переменная Color2 — цвет закрашки, процедура Отрезок рисует отрезок между двумя точками, функция Цвет\_точки возвращает цвет указанной точки.

Алгоритм вызывается с параметрами (X, Y, -1, X, X), где X и Y — координаты точки а внутри области.

## Триангуляция полигона

Задание графических объектов в виде полигональных моделей используется в компьютерной графике повсеместно. Однако при анализе и визуализации таких моделей часто намного удобнее иметь дело не с многоугольниками, а с треугольниками. Треугольник — выпуклый многоугольник, анализировать его значительно проще. Возникает задача разбиения многоугольника на составляющие его треугольники. Эта задача называется *триангуляцией*.

Разбить выпуклый многоугольник на треугольники просто. Достаточно соединить любую из его вершин со всеми другими, кроме соседних (Рис. 76).

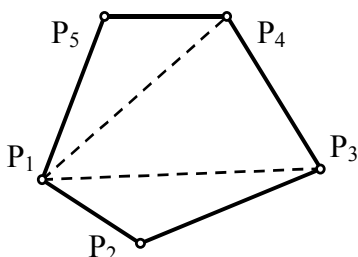


Рис. 76. Триангуляция выпуклого пятиугольника

Многоугольник задается своими вершинами в порядке обхода против часовой стрелки (и это очень важное условие). На Рис. 76 вершины обозначены символом  $P$  с индексом, соответствующим порядковому номеру вершины в описании. Алгоритм триангуляции выпуклого многоугольника добавляет в сцену ребра, образуемые вершиной  $P_1$  и вершинами  $P_3 \dots P_{N-1}$  (Рис. 77).

Процедура Трианг1 (Матрица  $P$ , Ребра  $E$ )

Цел  $I$

Изменяя  $I$  От 3 До  $P$ .Размер-1

$E$ .Добавить.Точки 1,  $I$

Следующее  $I$

Конец Процедур

Рис. 77. Алгоритм триангуляции выпуклого многоугольника

Для невыпуклого многоугольника этот алгоритм не сработает (Рис. 78).

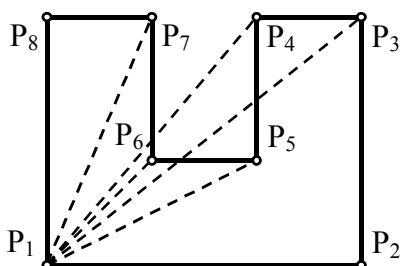


Рис. 78. Триангуляция невыпуклого многоугольника простым алгоритмом

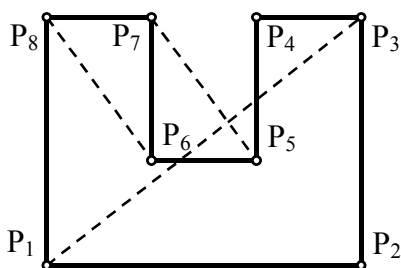
Будем рассматривать три соседние вершины  $P_H$ ,  $P_I$ ,  $P_J$   $N$ -многоугольника, как кандидата на треугольник. При этом  $H=I-1$ ,  $J=I+1$ . Для случая, когда  $I=1$ , будем принимать  $H=N$ . Для случая, когда  $I=N$ , будем принимать  $J=1$ . Таким образом мы обеспечим цикличность обхода вершин.

Вершина  $I$  будет выпуклой, если порядок обхода вершин треугольника выполняется в порядке  $H-I-J$ . Например, для вершин  $P_6-P_7-P_8$  вершина  $P_7$  будет выпуклой, а для вершин  $P_5-P_6-P_7$  — нет (Рис. 79).

Определить, образуют три вершины выпуклый треугольник, или нет, можно при помощи детерминанта:

$$D = \begin{vmatrix} X_H & Y_H & 1 \\ X_I & Y_I & 1 \\ X_J & Y_J & 1 \end{vmatrix}.$$

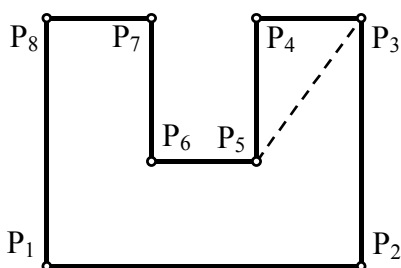
Если выполняется условие  $D > 0$ , вершины образуют треугольник.



**Рис. 79. Обход трех соседних вершин многоугольника**

Данного условия недостаточно. Три вершины  $P_1—P_2—P_3$  на Рис. 79 образуют выпуклый треугольник, однако провести диагональ  $P_1—P_3$  нельзя.

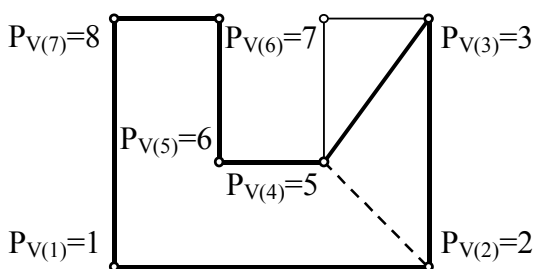
Введем дополнительное условие. Будем выбирать наикратчайшую диагональ между вершинами  $P_H$  и  $P_J$  для допустимых троек вершин. Тогда, если начинать с  $I=1$ , и обходить все тройки вершин, найдем, что диагональ  $P_2—P_3$  является первой кратчайшей (Рис. 80).



**Рис. 80. Первая кратчайшая диагональ**

После отсечения треугольника  $P_3—P_4—P_5$ , то есть после нахождения выпуклой вершины  $I=4$ , остался многоугольник  $P_1—P_2—P_3—P_5—P_6—P_7—P_8—P_1$ , то есть многоугольник  $P_1—P_2—\dots—P_{I-1}—P_{I+1}—\dots—P_N$ . Этот многоугольник проверяется так же, как и исходный.

Так как число точек многоугольника уменьшилось и последовательная нумерация нарушилась, нужен механизм для правильной нумерации вершин. Введем массив  $V$  для хранения последовательных номеров вершин.



**Рис. 81. Многоугольник после отсечения первого треугольника**

Изначально он имеет размерность  $N$  и заполнен номерами от 1 до  $N$ . После того, как найдена вершина  $I$ , исключаем ее из массива вершин, сдвигая его

при помощи цикла. Тогда после нахождения первой вершины  $I=4$  массив  $V$  будет содержать значения 1—2—3—5—6—7—8. Для нумерации вершин в этом случае следует использовать значения элементов массива.

**Процедура Трианг(Матрица  $P$ , Ребра  $E$ )**

```

Конст BIG = 1E30
Цел H, I, J, L, N, M, MinI
Вещ MinDiag, Diag
M = P.Размер
Размер V(M)
Изменяя I От 1 До M
    V(I) = I
Следующее I
Цикл Пока (M > 2)
    MinDiag = BIG
    Изменяя I От 1 До M
        Если (I = 1) То H = M Иначе H = I - 1
        Если (I = M) То H = 1 Иначе H = I + 1
        Если Выпуклый(P, H, I, J, Diag) То
            Если (Diag < MinDiag) То
                MinDiag = Diag
                MinI = I
            Конец Если
        Конец Если
    Следующее I
    I = MinI
    Если (I = 1) То H = M Иначе H = I - 1
    Если (I = M) То H = 1 Иначе H = I + 1
    E.Добавить.Точки V(H), V(J)
    M = M - 1
    Изменяя L От 1 До M
        V(L) = V(L + 1)
    Следующее L
Конец Цикла
Конец Процедуры

```

**Рис. 82. Алгоритм триангуляции**

Так, после исключения первого треугольника с вершиной  $I=4$ , обход троек вершин, начиная с первой, даст минимальную диагональ между вершинами  $P_{V(2)}$  и  $P_{V(3)}$  (Рис. 81). Далее массив  $V$  снова перенумеровывается и процесс повторяется с первой вершины.

**Лог Функция Выпуклый(Матрица  $P$ , Цел  $H, I, J$ , Вещ  $Dist$ )**

```

Вещ DXI, DYI, DXJ, DYJ, Det
DXI = P(V(I)).X - P(V(H)).X
DYI = P(V(I)).Y - P(V(H)).Y
DXJ = P(V(J)).X - P(V(H)).X
DYJ = P(V(J)).Y - P(V(H)).Y
Dist = DXJ * DXJ + DYJ * DYJ
Det = DXI * DYJ - DXJ * DYI
Выпуклый = Det > 0.000001

```

**Конец Функции**

**Рис. 83. Вспомогательный алгоритм триангуляции**

Алгоритм триангуляции приведен на Рис. 82. Для его реализации требуется вспомогательная функция, которая вычисляет детерминант и возвращает признак, указывающий на допустимость треугольника. Функция дополнительно вычисляет длину диагонали между вершинами  $H$  и  $J$ , точнее, квадрат длины, так как фактическое значение длины не имеет значения (Рис. 83).

Результат работы алгоритма приведен на Рис. 84. Цифрами обозначена последовательность добавления диагоналей.

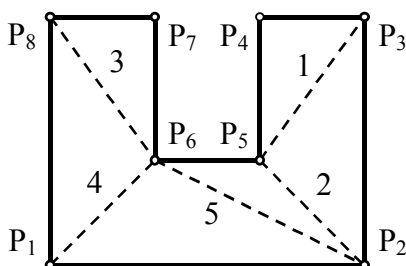


Рис. 84. Результат триангуляции

## Контрольные вопросы и задания

1. Что является основанием для выбора очередного пиксела в алгоритме Брезенхейма?
2. Почему первоначальный вариант алгоритма Брезенхейма прорисовывает не все пиксели (не обеспечивает связность)?
3. В чем основное преимущество алгоритма Брезенхейма?
4. Если первая точка отрезка находится в области с кодом 9, какая линия в алгоритме отсечения отрезка выполнит первое отсечение — нижняя горизонтальная или левая вертикальная?
5. Почему код правой нижней области в алгоритме отсечения равен 12?
6. Функция `КодОбласти` в алгоритме отсечения вернула значение 5. Это верное значение? Почему?
7. Если проверяемая точка в алгоритме определения принадлежности точки многоугольнику находится в верхней вершине (Рис. 69), то она засчитывается или нет? А если в нижней?
8. В какую сторону начинает закрашивать область алгоритм закрашки, приведенный на Рис. 75? Почему?
9. Попробуйте найти такой многоугольник, который будет неправильно триангулироваться.



## Рекомендуемая литература

1. Л. Аммерал. Принципы программирования в машинной графике. Пер. с англ. — М.: "Сол Систем", 1992. — 224 с.: ил.
2. Роджерс Д., Адамс Дж. Математические основы машинной графики: Пер. с англ./Пер. Ю. П. Кулябичев, В. Г. Иваненко; ред. Ю. И. Топчиев. — М.: Машиностроение, 1980. — 240 с., ил.
3. Порев В. Н. Компьютерная графика. — СПб.: БХВ-Петербург, 2002. — 432 с.: ил.
4. Е. В. Шикин, А. В. Боресков. Компьютерная графика. Динамика, реалистические изображения. М.: "Диалог-МИФИ", 1995. — 288 с.

## Приложения

### Таблицы матриц преобразования

#### Элементы матрицы преобразования 4×4

$a$	$b$	$c$	$p$
$d$	$e$	$f$	$q$
$g$	$i$	$j$	$r$
$l$	$m$	$n$	$s$

Элементы  $a, b, c, d, e, f, g, i, j$  выполняют частичное изменение масштаба (растяжение-сжатие вдоль координатных осей), сдвиг, отображение (отражение) и вращение. Элементы  $l, m, n$  выполняют перенос. Элемент  $s$  выполняет общее преобразование масштаба. Элементы  $p, q, r$  выполняют перспективные преобразования.

#### Матрицы отражения, масштабирования и переноса

**REFL3DXY** — отражение относительно плоскости XY

1	0	0	0
0	1	0	0
0	0	-1	0
0	0	0	1

**REFL3DYZ** — отражение относительно плоскости YZ

-1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

**REFL3DZX** — отражение относительно плоскости ZX

1	0	0	0
0	-1	0	0
0	0	1	0
0	0	0	1

**SCALE3DX** — частичное изменение масштаба по X в  $a$  раз

$a$	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

**SCALE3DY** — частичное изменение масштаба по Y в  $e$  раз

0	0	0	0
0	$e$	0	0
0	0	1	0
0	0	0	1

**SCALE3DZ** — частичное изменение масштаба по Z в  $j$  раз

$$\begin{vmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & j & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

**SCALE3D** — общее преобразование масштаба в  $1/s$  раз

$$\begin{vmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & s \end{vmatrix}$$

**TRANSL3D** — перенос на вектор  $(l, m, n)$

$$\begin{vmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ l & m & n & 1 \end{vmatrix}$$

## Матрицы вращения

**ROT3DX** — вращение вокруг оси X

$$\begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos & \sin & 0 \\ 0 & -\sin & \cos & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

**ROT3DY** — вращение вокруг оси Y

$$\begin{vmatrix} \cos & 0 & -\sin & 0 \\ 0 & 1 & 0 & 0 \\ \sin & 0 & \cos & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

**ROT3DZ** — вращение вокруг оси Z

$$\begin{vmatrix} \cos & \sin & 0 & 0 \\ -\sin & \cos & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

## Матрицы параллельных проекций

### Ортогональные проекции

**ORTHOZ** — ортография на плоскость  $Z=n$

$$\begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & n & 1 \end{vmatrix}$$

**ORTHOX** — ортография на плоскость  $x=l$

0	0	0	0
0	1	0	0
0	0	1	0
$l$	0	0	1

**ORTHOY** — ортография на плоскость  $y=t$

1	0	0	0
0	0	0	0
0	0	1	0
0	$t$	0	1

### *Прямоугольные проекции*

**FRONT** — вид спереди

1	0	0	0
0	1	0	0
0	0	0	0
0	0	0	1

**BACK** — вид сзади

1	0	0	0
0	-1	0	0
0	0	-1	0
0	0	0	1

**LEFT** — вид слева

0	0	-1	0
0	1	0	0
1	0	0	0
0	0	0	1

**RIGHT** — вид справа

0	0	1	0
0	1	0	0
-1	0	0	0
0	0	0	1

**TOP** — вид сверху

1	0	0	0
0	0	1	0
0	-1	0	0
0	0	0	1

**BOTTOM** — вид снизу

1	0	0	0
0	0	-1	0
0	1	0	0
0	0	0	1

## АксонOMETрические проекции

АксонOMETрические преобразования выполняются общей матрицей. Угол  $\alpha$  вращает объект вокруг оси Y, угол  $\beta$  — вокруг оси X.

$$\begin{vmatrix} \cos\alpha & \sin\alpha \cdot \sin\beta & -\sin\alpha \cdot \cos\beta & 0 \\ 0 & \cos\beta & \sin\beta & 0 \\ \sin\alpha & -\cos\alpha \cdot \sin\beta & \cos\alpha \cdot \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

**Для триметрического** преобразования углы произвольные.

**Для диметрического** преобразования:

$$\alpha = 22,207654298596487487055620026178^\circ$$

$$\beta = 20,704811054635429669240251093463^\circ$$

**Матрица диметрического преобразования**

$$\begin{vmatrix} 0,935414346693485 & 0,133630620956212 & -0,327326835353989 & 0 \\ 0 & 0,925820099772551 & 0,377964473009227 & 0 \\ 0,353553390593274 & -0,353553390593274 & 0,866025403784439 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

**Для изометрического** преобразования:

$$\alpha = 22,207654298596487487055620026178^\circ$$

$$\beta = 45^\circ$$

**Матрица изометрического преобразования**

$$\begin{vmatrix} 0,707106781186548 & 0,408248290463863 & -0,577350269189626 & 0 \\ 0 & 0,816496580927726 & 0,577350269189626 & 0 \\ 0,707106781186548 & -0,408248290463863 & 0,577350269189626 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

## Косоугольные проекции

Проектирующие лучи наклонены к плоскости проекции под углом  $45^\circ$ .

$$\begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -a & -a & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Для свободной проекции (проекция Кавалье)  $a = \cos 45^\circ = 0,707106781186548$ .

Для кабинетной проекции (проекция Кабине)  $a = \frac{1}{2} \cdot \cos 45^\circ = 0,353553390593274$ .

## Перспективные проекции

В одноточечных перспективных проекциях один из элементов  $p, q, r$  не равен нулю, в двухточечных — два элемента не равны нулю, в трехточечных все три элемента не равны нулю.  $H = xp + yq + zr + 1$ .

$$\begin{vmatrix} 1 & 0 & 0 & p \\ 0 & 1 & 0 & q \\ 0 & 0 & 1 & r \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Центр проекции лежит на расстоянии  $-1/p$  ( $-1/q, -1/r$ ) от центра.

Точка схода параллельных линий противоположна центру проекции.

## Видовое преобразование

Параметрами видового преобразования являются угол  $\theta$  в плоскости XY, угол  $\varphi$  отклонения от оси Z и расстояние  $\rho$  от центра видовой системы координат (точки зрения) до центра мировой системы координат.

Для получения левосторонней видовой системы матрица имеет вид:

$$\begin{vmatrix} -\sin \theta & -\cos \theta \cdot \cos \varphi & -\cos \theta \cdot \sin \varphi & 0 \\ \cos \theta & -\sin \theta \cdot \cos \varphi & -\sin \theta \cdot \sin \varphi & 0 \\ 0 & \sin \varphi & -\cos \varphi & 0 \\ 0 & 0 & \rho & 1 \end{vmatrix}$$

Для получения правосторонней видовой системы матрица имеет вид:

$$\begin{vmatrix} -\sin \theta & \cos \theta \cdot \cos \varphi & -\cos \theta \cdot \sin \varphi & 0 \\ \cos \theta & \sin \theta \cdot \cos \varphi & -\sin \theta \cdot \sin \varphi & 0 \\ 0 & -\sin \varphi & -\cos \varphi & 0 \\ 0 & 0 & \rho & 1 \end{vmatrix}$$

## Простое перспективное преобразование

Выполняется по формулам:

$$x^* = x \cdot D \cdot \text{Aspect} / z$$

$$y^* = y \cdot D \cdot \text{Aspect} / z$$

Здесь D — расстояние до плоскости экрана, Aspect — масштабирующий множитель. Для *автостабилизации* размера изображения следует установить параметр D равным параметру  $\rho$  видового преобразования.

## Формулы приведения углов

$\beta$	$\sin \beta$	$\cos \beta$	$\beta$	$\sin \beta$	$\cos \beta$
$90 + \alpha$	$\cos \alpha$	$-\sin \alpha$	$-(90 + \alpha)$	$\cos \alpha$	$\sin \alpha$
$180 + \alpha$	$-\sin \alpha$	$-\cos \alpha$	$-(180 + \alpha)$	$\sin \alpha$	$-\cos \alpha$
$270 + \alpha$	$-\cos \alpha$	$\sin \alpha$	$-(270 + \alpha)$	$-\cos \alpha$	$-\sin \alpha$
$-\alpha$	$-\sin \alpha$	$\cos \alpha$			
$90 - \alpha$	$\cos \alpha$	$\sin \alpha$	$-(90 - \alpha)$	$\cos \alpha$	$-\sin \alpha$
$180 - \alpha$	$\sin \alpha$	$-\cos \alpha$	$-(180 - \alpha)$	$-\sin \alpha$	$-\cos \alpha$
$270 - \alpha$	$-\cos \alpha$	$-\sin \alpha$	$-(270 - \alpha)$	$-\cos \alpha$	$\sin \alpha$

## Матричные операции

**Матрица** — это прямоугольный массив чисел, называемых *элементами матрицы*, состоящий из *строк* и *столбцов* и подчиняющийся правилам матричной алгебры. Матрица из  $m$  строк и  $n$  столбцов называется матрицей  $m \times n$ :

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} = [a_{ij}]_{m \times n}$$

Матрицу, имеющую только один столбец или только одну строку, называют также **вектором**. Матрицу из одной строки называют **матрицей-строкой** или вектором-строкой, например:

$$A = [a_1 \ a_2 \ \dots \ a_n].$$

Матрица из одного столбца называется **матрицей-столбцом** или вектором-столбцом, например:

$$A = \begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_n \end{bmatrix}$$

Если число строк и столбцов равны между собой, матрица является **квадратной**. Для квадратной матрицы общее число строк или столбцов называется **порядком** матрицы. В компьютерной графике используются квадратные матрицы порядка 2..4.

**Главной диагональю** квадратной матрицы называется диагональ, проходящая через верхний левый и правый нижний углы, то есть совокупность элементов матрицы  $a_{ii}$ , где  $i = 1..m$ . Противоположная диагональ называется **побочной**.

Квадратная матрица, у которой все элементы, расположенные вне главной диагонали, равны нулю, называется **диагональной**. Диагональная матрица, все элементы главной диагонали которой равны единице, называется **единичной** и обозначается  $E_n$ :

$$E_n = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

**Суммой**  $A + B$  матриц одного типа называется матрица  $C$  того же типа, элементами которой являются суммы соответствующих элементов матриц  $A$  и  $B$ :

$$c_{ij} = a_{ij} + b_{ij}.$$

Вычисление суммы двух матриц осуществляется двукратным циклом:

```
изменяя i от 1 до m
    изменяя j от 1 до n
         $c_{ij} = a_{ij} + b_{ij}$ 
    кц
кц
```

**Вычитание** матриц производится аналогично сложению, меняется только знак операции.

**Произведением матрицы на число** называется матрица, элементы которой получены умножением каждого элемента матрицы на это число:

$$A \cdot \alpha = \begin{bmatrix} a_{11} \cdot \alpha & a_{12} \cdot \alpha & \dots & a_{1n} \cdot \alpha \\ a_{21} \cdot \alpha & a_{22} \cdot \alpha & \dots & a_{2n} \cdot \alpha \\ \dots & \dots & \dots & \dots \\ a_{m1} \cdot \alpha & a_{m2} \cdot \alpha & \dots & a_{mn} \cdot \alpha \end{bmatrix}$$

**Произведением** матрицы  $A = [a_{ij}]_{m \times n}$  на матрицу  $B = [b_{ij}]_{p \times q}$  называется матрица  $C = [c_{ij}]_{m \times q}$ , элементы которой вычисляются по формуле:

$$c_{ij} = \sum a_{ik} \cdot b_{kj}, \text{ где } i = 1..m, j = 1..q, k = 1..p,$$

то есть элемент  $ij$  матрицы  $C$  есть сумма произведений элементов  $i$ -той строки матрицы  $A$  на элементы  $j$ -того столбца матрицы  $B$ .

Произведение матриц возможно только в случае, если число строк матрицы  $B$  равно числу столбцов матрицы  $A$ , то есть когда  $n=p$ . В результирующей матрице столько строк, сколько в первой матрице-сомножителе и столько столбцов, сколько во второй матрице-сомножителе.

Для вычисления произведения матриц используется тройной цикл с накоплением суммы произведений:

```
изменяя i от 1 до m
    изменяя j от 1 до q
        cij = 0
        изменяя k от 1 до n
            cij = cij + aik · bkj
        кц
    кц
кц
```

При программировании матричных преобразований в компьютерной графике эффективным оказывается метод, при котором предварительно определяются формулы для вычисления элементов одной строки результирующей матрицы. Например, для вычисления результата перемножения матриц:

$$\begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \end{bmatrix} \cdot \begin{bmatrix} d & 0 & 0 \\ 0 & e & 0 \\ 0 & 0 & f \end{bmatrix} = \begin{bmatrix} x_1 \cdot d & y_1 \cdot e & z_1 \cdot f \\ x_2 \cdot d & y_2 \cdot e & z_2 \cdot f \end{bmatrix}$$

достаточно определить формулы вычисления элементов столбцов результирующей матрицы:

$$\begin{aligned} c_{i1} &= x_i \cdot d \\ c_{i2} &= y_i \cdot e \\ c_{i3} &= z_i \cdot f. \end{aligned}$$

Это снижает количество выполняемых вычислительных операций и ускоряет обработку изображений. Результирующая матрица получается в этом случае однократным циклом по числу строк  $m$  левой матрицы-сомножителя:

```
изменяя i от 1 до m
    ci1 = xi · d
    ci2 = yi · e
    ci3 = zi · f
кц
```