

## Компьютерная графика.

1. Компьютерная графика
2. Интерактивные графические системы.
3. Анимация и видеографика.

**КГ** – это область относящаяся к информатике. В сферу интересов этой науки входят аспекты связанные с формированием изображения.

ЭВМ используется для: формирования, преобразования и вывода изображения (графических объектов)



### **Понятие машинной графики:**

В широком смысле отвечает за круг вопросов связанных с геометрической моделью, цифровой обработкой и анализом изображения, машинным зрением и т.д.

В узком смысле существует *программная* и *техническая*.

*Техническая часть* – Мониторы

*Программная часть* – драйвера, пакеты графических программ, графические протоколы и сетевые протоколы.

### **Основные понятия КГ**

- Принципы описания и представления их в памяти ЭВМ
- Алгоритмы формирования и преобразования изображения
- принципы построения тех. Средств ввода вывода изображения
- принципы организации систем машинной графики их программного обеспечения

Существует 3 направления:

- компьютерное зрение (Comp.Vision (CV))
- обработка изображения (Comp. Image Proc.(CIP=IP))
- компьютерная графика (Comp. Grath. (CG))

CV – у нас есть изображение, хотим получить описание т.е. занимаемся анализом образов



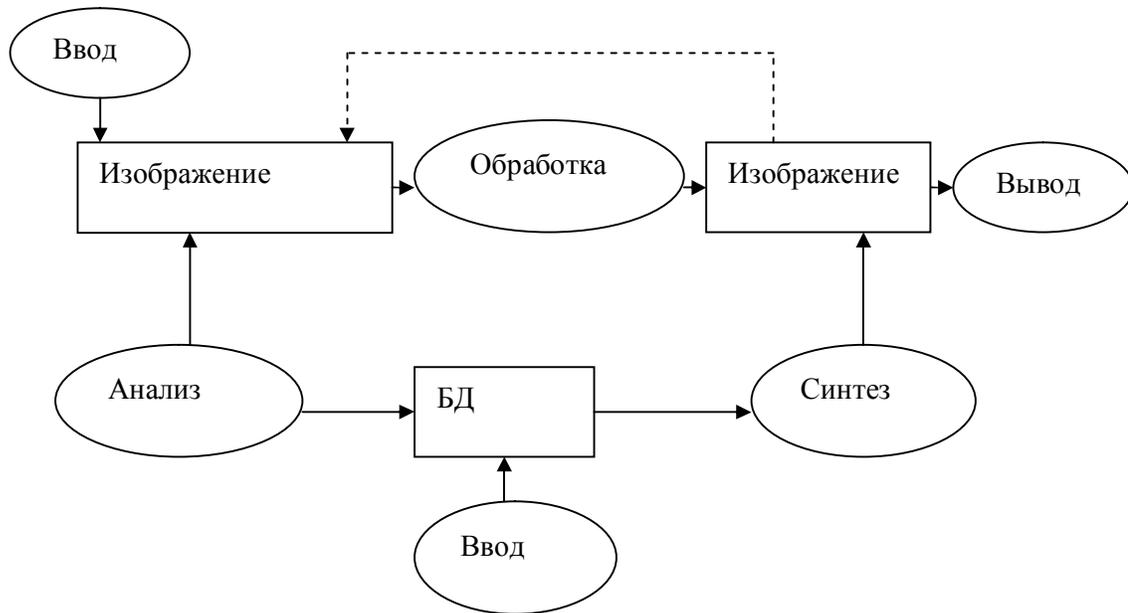
CG – имеется описание, хотим получить отображение (синтез изображения)



IP- есть изображение а мы производим его обработку



Схема изображения процессов в графической системе.



БД – база данных структурированного описания изображения

Формирование элементов, хранящихся в БД называется синтезом изображения ( м. проходить как по замкнутому так и по разомкнутому циклу)

Анализ – оценка, идентификация измерения изображения

Обработка – процесс который используется для улучшения качества изображения (приведения изображения к виду удобному для его последующего анализа и синтеза)

Виды информации:

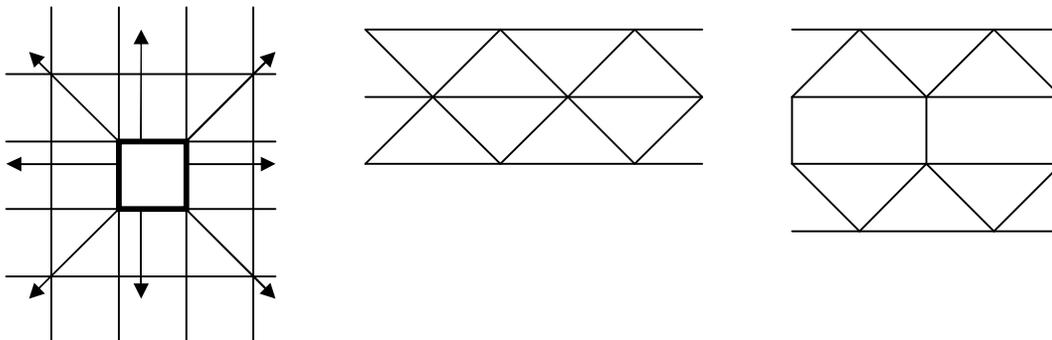
Информация м.б. :

- координатная ( место расположения, 2-мерная, 3- мерная и т.д.
- знаковая (хранится в wmf в виде некоторых знаков)
- видеоизображения

информация представляется в wmf массиве квантованных отчетов на поле (дискретное поле)

Дискретное поле – соответствует решетке и эта решетка обычно бывает равномерной ( к ее узлам привязывается наша информация ), этот элемент называется растр

### Формы растра



## Геометрическое моделирование

Целью ГМ является представление объектов

Чаще всего объекты реальные

Требования к ГМ :

- Любая модель, которую можно сконструировать не должна противоречить реальному объекту

- Допустимость конструирования модели целиком ( мощность модели)

- Возможность вычисления доп. Параметров (объема, веса и т.п.) Вычисляемость модели.

Компактность модели – количество информации необходимое чтобы эту модель использовать в различном алгоритме.

Характер открытости – возможность применения модели в других системах.

### *Математические основы компьютерной графики*

Способы представления базовых геометрических объектов(типов), преобразования

Представление геометрических объектов не зависит от конкретного времени.

1. Представления о геометрической сущности

2. Способы описания этой сущности в конкретной системе координат

3. Способы описания математической абстракции

Базируется на понятии аффинной геометрии и проекционного пространства.

Базируется на 3-х типах: скаляры, точки и вектора. (носят геометрическую сущность)

### *Геометрические определения базовых типов*

Роль фундамента – понятие точки  $R^0, R^1, R^2, R^3, \dots$

В 2-мерной геометрии точка – это положение ее (единственный атрибут)

Есть некая система координат

Есть 3 пространства интересующих нас: векторное пространство, аффинное пространство, евклидово пространство.

Скаляр – некое вещественное число, задаваемое для описания объектов, сущностей.

Вектор – некая направленная линия от 1-ой точки ко 2-ой

Линии 2-х сущностей в векторном пространстве: скаляр и вектор.

В аффинном пространстве добавляется тип точка. В евклидовом пространстве добавляется понятие расстояние.

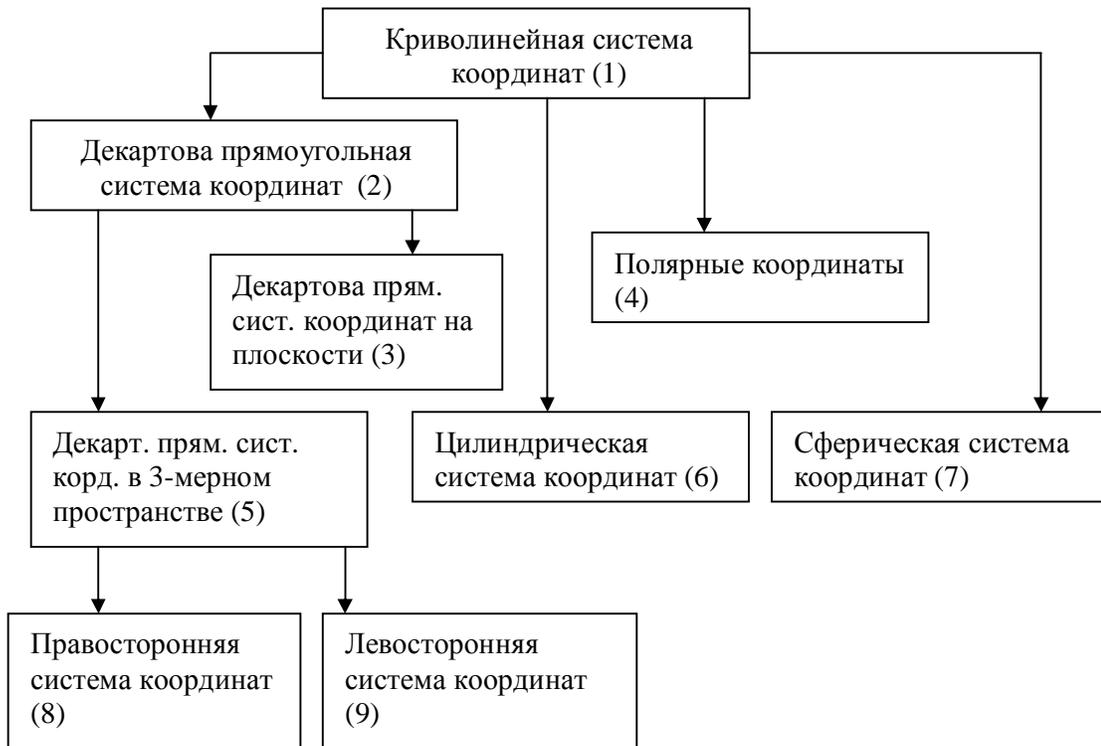
### **Система координат**

Положение точки  $t$  в пространстве может быть описано в виде некоторых линейно независимых векторов  $(v_1, v_2, v_3)$  А введя скаляры  $(a_1, a_2, a_3)$  можно описать вектор.

$$W = a_1 v_1 + a_2 v_2 + a_3 v_3$$

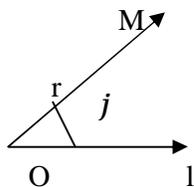
### **Декартова (Картезианская) система координат**

Структура взаимосвязей между системами координат



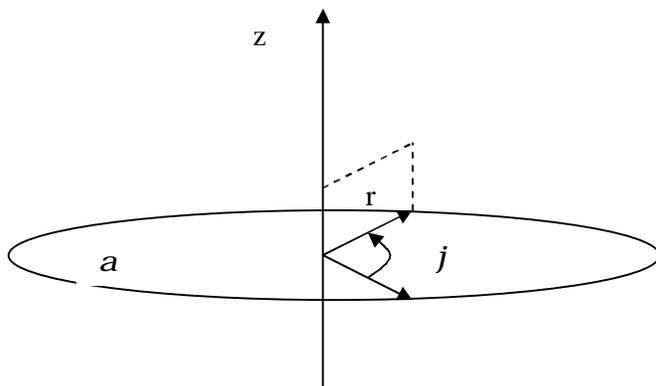
1) Базовая косоугольная система координат  
 Координаты определяются осями (  $x$  – ось абсцисс,  $y$  - ось ординат)  
 Расстояние определяется проекциями  $P(x, y)$

4) Полярная система координат



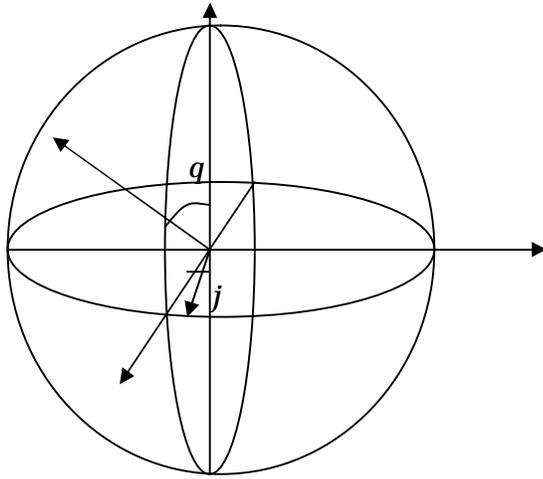
(. )0 – полюс,  $j$  - полярный угол ,  $r$  – полярное расстояние.  $M(r, j)$   
 Соответственно  $x = r \cos j$   
 $y = r \sin j$

6) Цилиндрические координаты  
 Есть некая плоскость



$Z$  проекция на точку  $M$

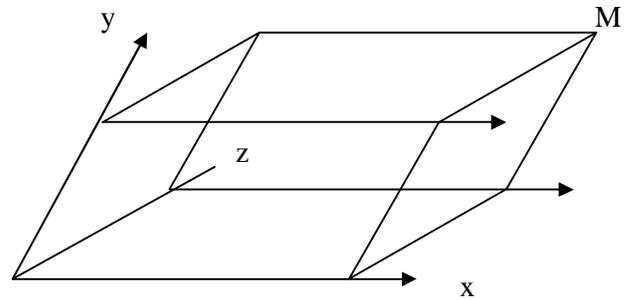
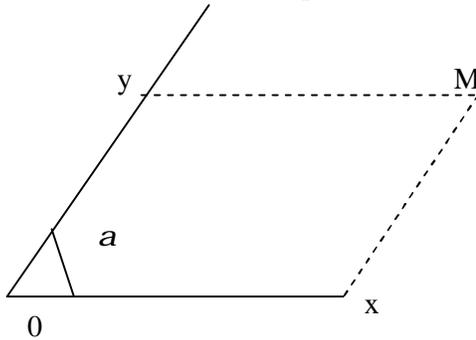
7) Сферические системы координат



угол  $q$  - полярное расстояние  
 угол  $j$  - долгота  
 $x = r \sin j \cos q$   
 Соответственно  $y = r \sin j \sin q$   
 $z = r \cos j$

1.  $r = |OM|$
2.  $j$
3.  $q$

Косоугольная система координат



Декартова система координат	$M(x,y,z)$	
Полярная	$M(r, j, q)$	$x = r \cos j$ $y = r \sin j$
Цилиндрическая	$M(r, j, z)$	$x = r \cos j$ $y = r \sin j = z$
Сферическая с коор.	$M(r, j, q)$	$x = r \sin j \cos q$ $y = r \sin j \sin q$ $z = z \cos j$

0 в середине экрана у Кватернианст.

(.)  $P(x, y)$  Все наши представления в векторах, в виде матрицы  $\begin{vmatrix} x \\ y \end{vmatrix}$

Преобразование в компьютерной графике

- проекционное преобразование
- геометрические (аффинные) преобразования

& Геометрические преобразования в компьютерной графике.

Аффинные преобразования.

Преобразования связанные с некоторыми изменениями объекта.

- Перенос

- Поворот

- Масштабирование

! Св - во аффинных преобразований

Св-во 1  $T * R^2 \rightarrow R^z$  всегда переводят прямую Q в T(a) так что множество точек прямой a отображается на множество (.) T(a)

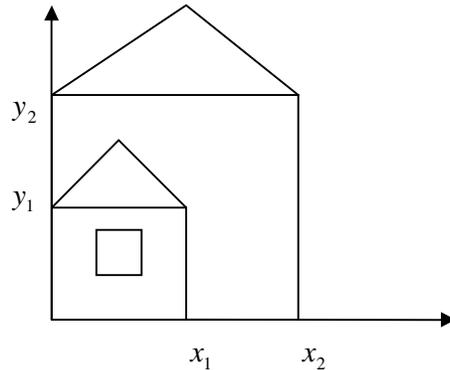
Св-во 2 Если множество прямых a и b параллельны и задано некое аффинное преобразование T(a) и T(b) будут также параллельны

Дополнительные свойства

- Произведение 2-х аффинных преобразований также аффинное преобразование
- Для каждого преобразования T существует такое  $T^{-1}$ , которое тоже является аффинным преобразованием.

Рассмотрим последовательно аффинные преобразование.

**(!) Масштабирование (scale)**



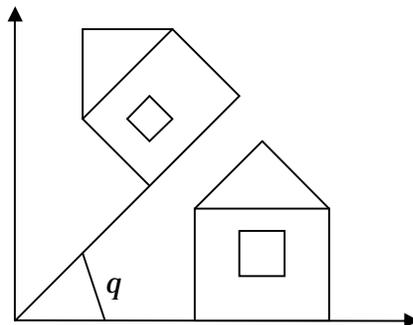
Если имеем  $P = \begin{pmatrix} x \\ y \end{pmatrix}$   
, а  $S = \begin{pmatrix} S_x & 0 \\ 0 & S_y \end{pmatrix}$   $S_x, S_y$  - коэф. Масштабиров.  
, то  $\begin{cases} x' = x * S_x \\ y' = y * S_y \end{cases}$

Свойства преобразования масштабирования

- не сохраняется длина

- не сохраняются углы

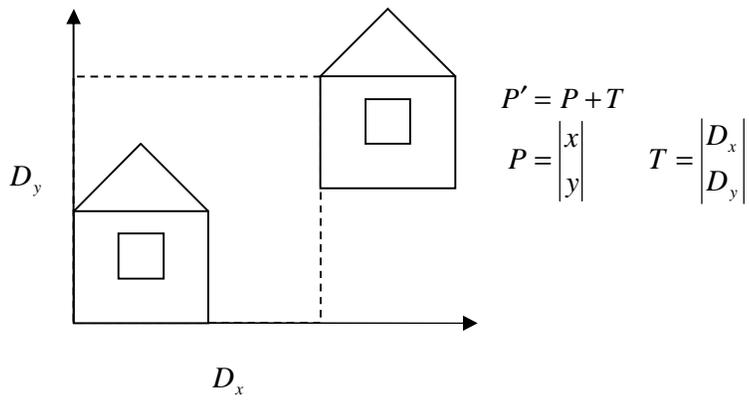
**(!) Поворот (rotate)**



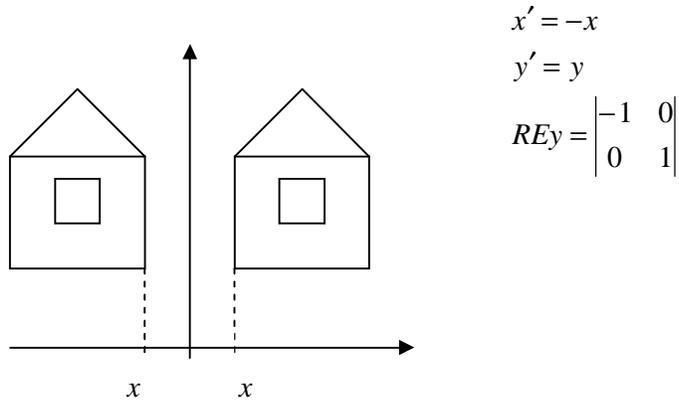
$P = \begin{pmatrix} x \\ y \end{pmatrix}$   
Если  $P' = S * P$  то  $P' = R_q * P$   
 $\begin{cases} x' = x \cos q - y \sin q \\ y' = y \sin q + x \cos q \end{cases}$

Соответственно матрица поворота  $R_0 = \begin{pmatrix} \cos q & -\sin q \\ \sin q & \cos q \end{pmatrix}$

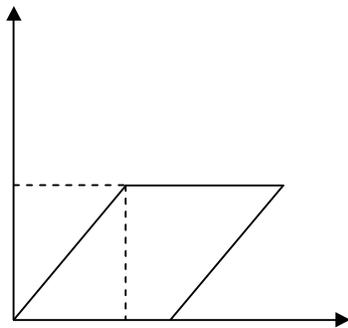
**(!) Перенос (Translate)**



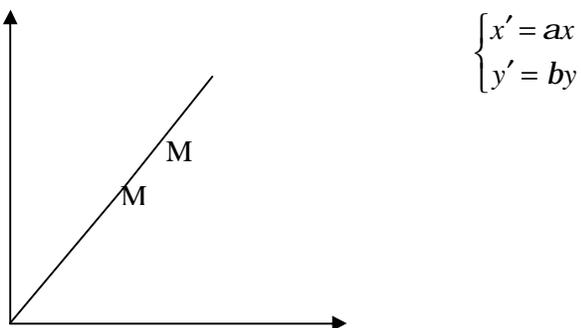
**(!) Отражение (Reflection)**



**(!) Скос (самостоятельно)**



**(!) Растяжение и сжатие**



$$[x, y, z] \begin{vmatrix} a & d & g \\ b & e & h \\ c & f & i \end{vmatrix} = \begin{vmatrix} ax+by+cz \\ dx+ey+fz \\ gx+hy+iz \end{vmatrix} = \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} \begin{vmatrix} x \\ y \\ z \end{vmatrix}$$

Матрица преобразования в однородной системе координат.

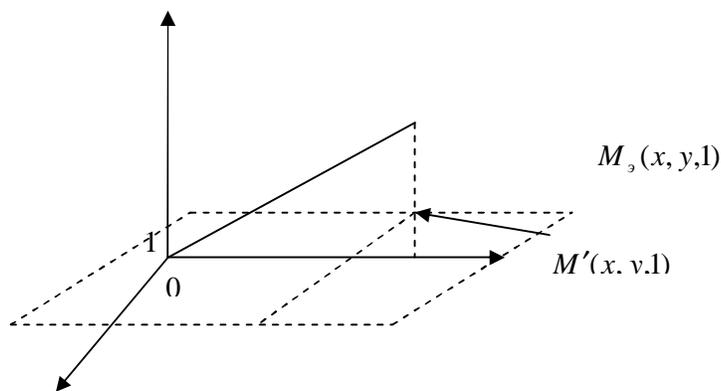
$$P' = P + T$$

$$P' = SP \quad \text{Пусть точка } M \text{ существует на плоскости с координатами } (x, y)$$

$$P' = RP$$

Однородной системой координат будет называться  $(x_1, x_2, x_3)$  связанная с точкой  $(x, y)$

$$\text{Соотношениями } x = \frac{x_1}{x_3} \quad y = \frac{x_2}{x_3}$$



Может быть задана произвольным набором координат

Матрица 3x3 для 2х мерного пространства

Матрица 4x4 для 3х мерного пространства

$(V_x, W_y, W)$

промасштабируем с коэффициентом  $W$  в 2х мерной плоскости

$$\left(\frac{x}{W}, \frac{y}{W}, 1\right)$$

$$T(dx, dy) = \begin{vmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{vmatrix}$$

$$S(S_x, S_y) = \begin{vmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

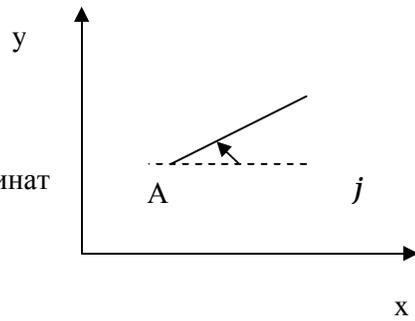
$$R(j) = \begin{vmatrix} \cos j & -\sin j & 0 \\ \sin j & \cos j & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

Построить матрицу поворота вокруг произвольной точки на плоскости  $(a, b)$

Шаги:

- 1) Перенос на вектор  $b = -a - b$

$$T_1 = \begin{vmatrix} 1 & 0 & -a \\ 0 & 1 & -b \\ 0 & 0 & 1 \end{vmatrix}$$



2) Поворот на угол  $j$  начала координат

$$R_0 = \begin{vmatrix} - & - & 0 \\ - & - & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

3) Перенос в (...) А с координатами

$$T_2 = \begin{vmatrix} 1 & 0 & a \\ 0 & 1 & b \\ 0 & 0 & 1 \end{vmatrix}$$

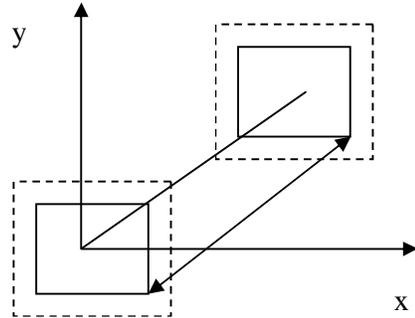
Перенос – Перенос -> коммутативные операции

Масштаб- Масштаб

Поворот- Поворот

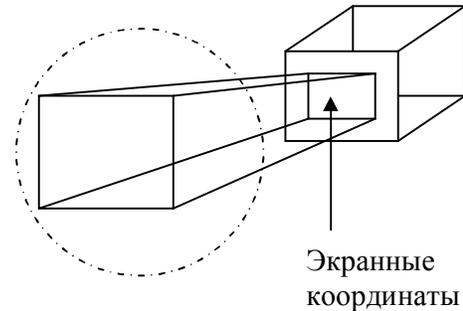
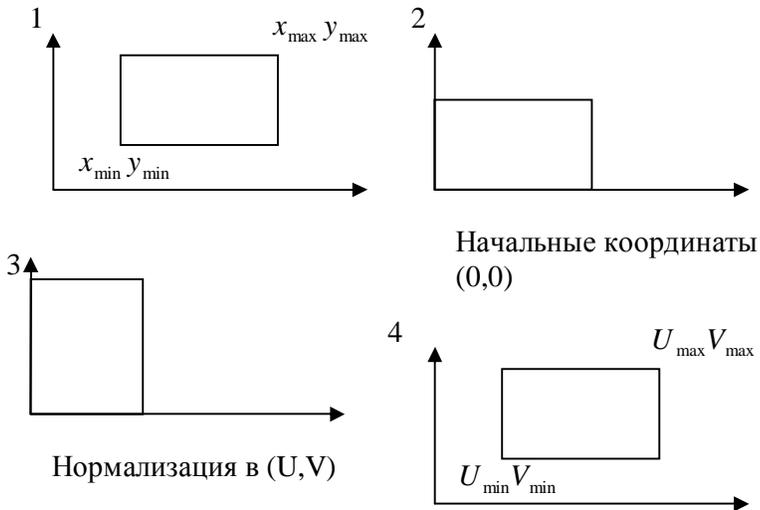
Перенос – Масштаб -> некоммутативные операции

Поворот - Перенос



### Преобразование в пространстве

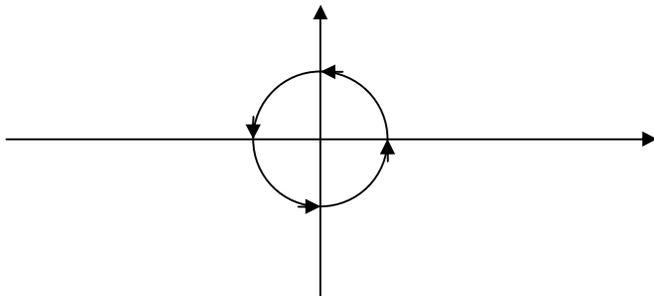
Шаги при преобразовании в VP



$$\left(\frac{x}{w}, \frac{y}{w}, \frac{z}{w}\right) \rightarrow (4x4)$$

Правосторонняя или левосторонняя по правилу буравчика

Положительными будем считать повороты (если смотреть с конца положительной полуоси в сторону начала координат) при которых поворот на 90 градусов против часовой стрелки будет проводить одну положительную ось на другую.



оси	Положительно б. направлено
x	от y к z
y	от z к x
z	от x к y

1. Перенос

$$T(dx, dy, dz) = \begin{vmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

2. Масштаб

$$S(S_x, S_y, S_z) = \begin{vmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

$$R_x(0) = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos 0 & -\sin 0 & 0 \\ 0 & \sin 0 & \cos 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

$$R_y(0) = \begin{vmatrix} \cos 0 & 0 & \sin 0 & 0 \\ 0 & 1 & 0 & 0 \\ -\sin 0 & 0 & \cos 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

$$R_z(0) = \begin{vmatrix} \cos 0 & -\sin 0 & 0 & 0 \\ \sin 0 & \cos 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

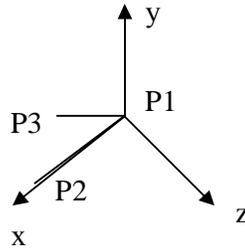
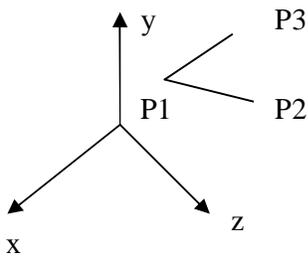
Композиция 3-х мерных преобразований

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

$$\begin{vmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} x \\ y \\ z \\ 1 \end{vmatrix} =$$

$$\begin{vmatrix} x & y & z & 1 \\ S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

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

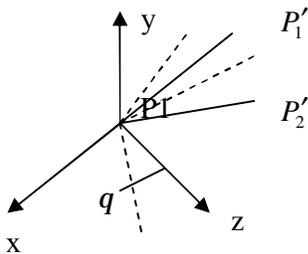


Повор. Вокруг н.к.  
Чтобы P1 и P2 совместились с  
Осью z

1. Перенос P1 в н.к.:

$$T(-x_1, -y_1, -z_1) = \begin{vmatrix} 1 & 0 & 0 & -x_1 \\ 0 & 1 & 0 & -y_1 \\ 0 & 0 & 1 & -z_1 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

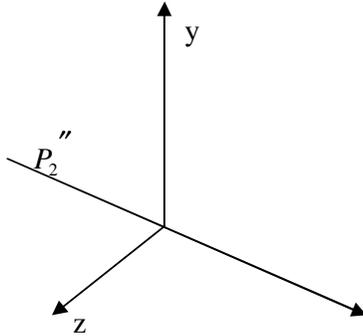
$$\begin{cases} P'_1 = TP_1 \\ P'_2 = TP_2 \\ P'_3 = TP_3 \end{cases} \text{ после 1-го шага}$$



2. Поворот вокруг оси y  $q$  - положит.,  $Aq'$  - поворот в обратную сторону  
 $q' = -(90 - q)$

$$\cos q' = z'_2 / D_1 \quad \sin q' = x'_2 / D_1$$

$$D_1 = \sqrt{(z'_2)^2 + (x'_2)^2} \quad P_2'' = R_y(q - 90)P_2'$$



3. Поверн. На угол  $j$  относит ос  $x$

$$\cos j = z_2'' / D_2 \quad \sin j = y_2'' / D_2$$

$$D_2 = |P_1''P_2''| = |P_1P_2| = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

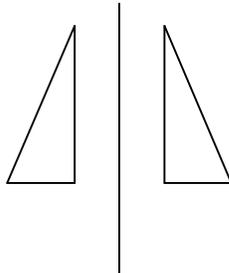
$$P_2'' = R_x(j)P_2' = P_2|j|R_y(q - 90)P_2'$$

4. Поместить  $P_3$  в плоскость  $YOZ$  – самостоятельно.

Многомерные пространства и проекции

Самая часто встреч.

1. Симметрия отображения (зеркальная симметрия)



2. Поворот – Отношение симметр. Своб. К повороту на определенный угол.

Ось поворота – поворотная ось симметрии.

Поворотная ось – ось вокруг которой может осуществить поворот до совмещения с самим собой.

Наименьший угол на который не повернуть – это Элементарный угол. Он определяет порядок, ось симметрии.

f.c.  $90^0 \rightarrow 360 = 90 * 4 \Rightarrow 4$ -ый порядок оси симметрии

120 – 3-й порядок.

3. Симметрия переноса – трансляция

Повторяемость фигуры в пространстве осуществляется ч/з некоторое расстояние.

Также называется симметрией 2-го переноса

5. Винтовая симметрия

Происходит перемещение и поворот по определенной траектории

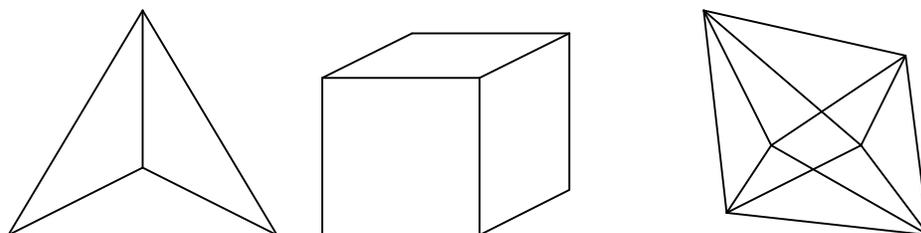
Многогранники – тело, ограниченное определен. количеством поверхностей которые представляет собственные многоугольники.

прим. Призма, пирамида.

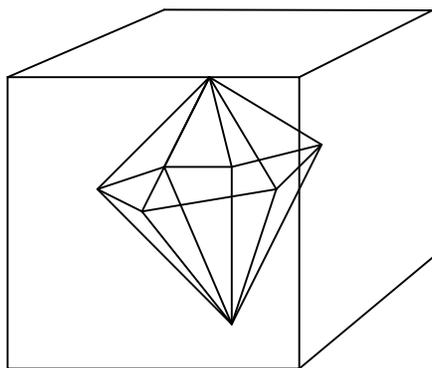
Правильный – выпуклый многоугольник обладающий максимальной 3-хмерной плотностью упаковки.

Известно 5 таких многогранников:

Платоново тело	Число сторон у граней	Число ребер у вершин	Число ребер	Число вершин	Общее число ребер
Тетраэдр	3	3	4	4	6
Куб (иксаэдр)	4	3	6	8	12
Октаэдр	3	4	8	6	12
Додекаэдр	5	3	12	20	30
Икосаэдр	3	5	20	12	30

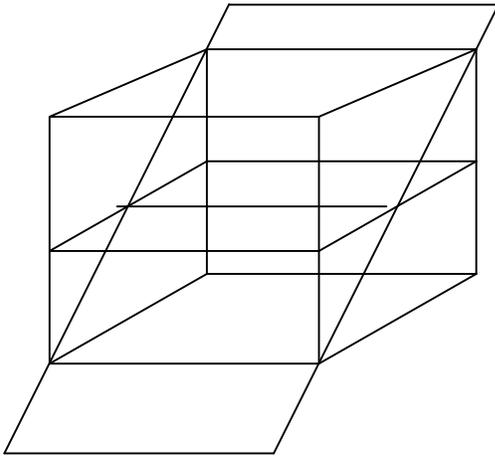


1. Они м. б. построены они из др.
2. Вокруг кажд. м. описать сферу и каждый м. вписать в сферу. Из куба – октоэдр: Вершина октаэдра – центральный граней куба.



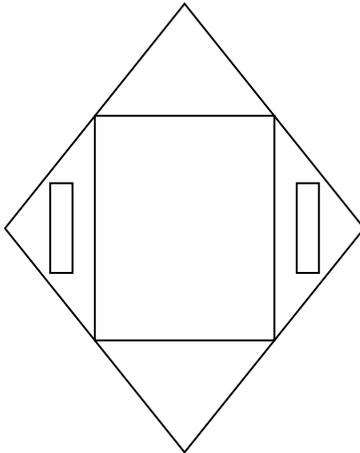
Данные тела обладают симметрией многогранников.

1. Есть центр. Симметрии
  2. Существуют плоскости симметрии
    - 3 плоскости симметрии перпендикулярны ребрам в их серединах.
    - 6 плоскостей симметрии проходящих через противоположные ребра.
  3. Оси симметрии
    - Существуют оси симметрии 4-го порядка
    - они проходят через центры граней
    - 6 осей симметрии 2-го порядка – проходят через середины противополож. Ребер
    - 4 оси зеркальной симметрии – проходят через середины противоположных ребер.
- Аналог м. для всех Платоновых тел

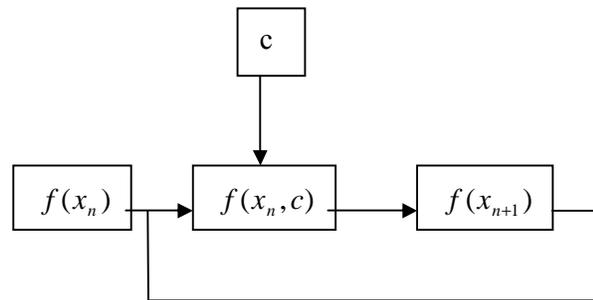


## Фрактал

1985 г. (IBM) Красота фрактала  
Кривая ПИАНО



Рекурсивна, в каждой части отрезка применяется Процедура  
Фрактал – это вызов подобного себе.



Пока с целое – ничего особенного  
с вещественное – множество Монденброта  
Свойства фракталов

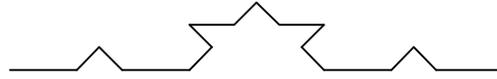
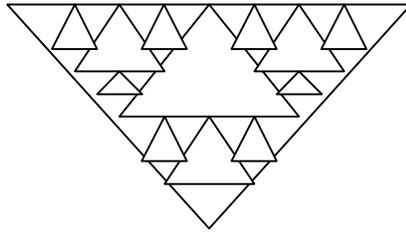
1. Компактно и ограничено.
  2. Для любой достаточно малой частицы.
  3. достаточно большая часть подобна ей
- Самоподобие – основное свойство фрактала
4. Фрактал – некая геометрическая фигура , состоящая из частей, каждая из которых представляет собой уменьшенную копию исходной фигуры.
  5. Некая граничная линия, имеющая бесконечную длину, но вводя дробную размерность можно говорить о ее длине.
- Линия не самопересекается.

Виды фракталов.

1. Геометрический
2. Алгебраический
3. Система интегрирующих функций
4. Стохастический

1) Геометрический – фрактал получаемый путем простых геометрических преобразований

Кривая Пиано, Снежинка Коха.



L- System ( одновременно с “черепаший графикой” – штрих рисование)

Если F – прочертить отрезок

+ - поворот по часовой стрелке

- - поворот против часовой стрелке

то кривая Коха  $F = F + F - F + F$

2) Алгебраический – строятся на основе некоторых алгебраических формул.

Многочастный итерационный процесс вызова функций с некоторой const c (веществ., которая плоскость)

3) Система итерирующих функций ИФС Применение таких преобразований, которые дают ту фигуру которую необходимо.

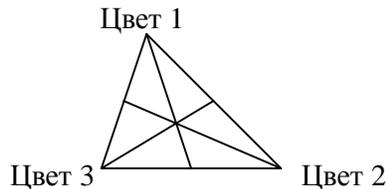
Папоротник – (рекурсивная функция с применением аффинных операций)

Рекурсивная операция требует большой вычислительной мощности.

4) Стохастический

Прим. Плазма.

Если не цвет, а координата  $\Rightarrow$  плоскость

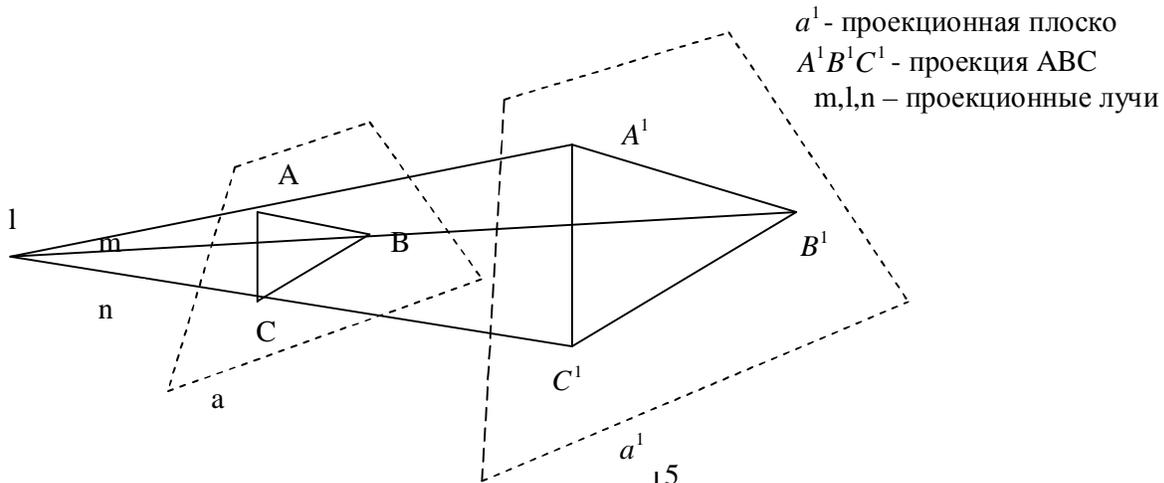


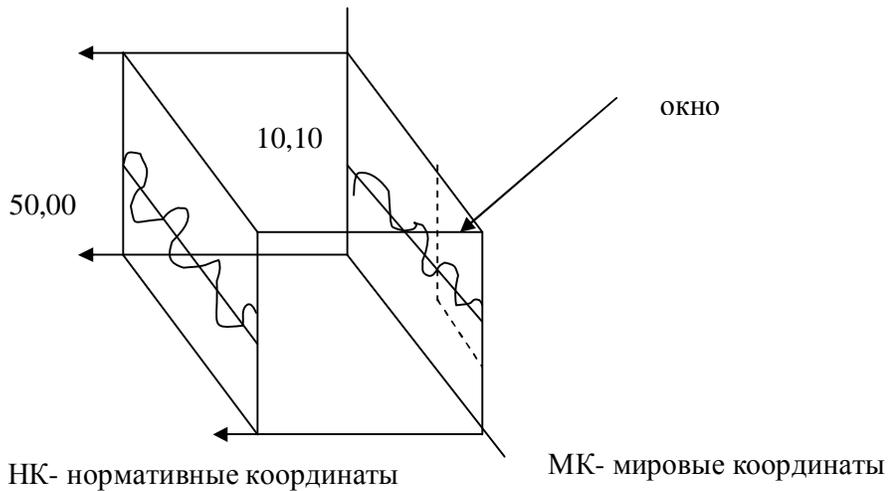
Некоторые возмущения

### Проективная геометрия

Проективная плоскость – в отличие от эвклидовой не имеет бесконечной протяженности.

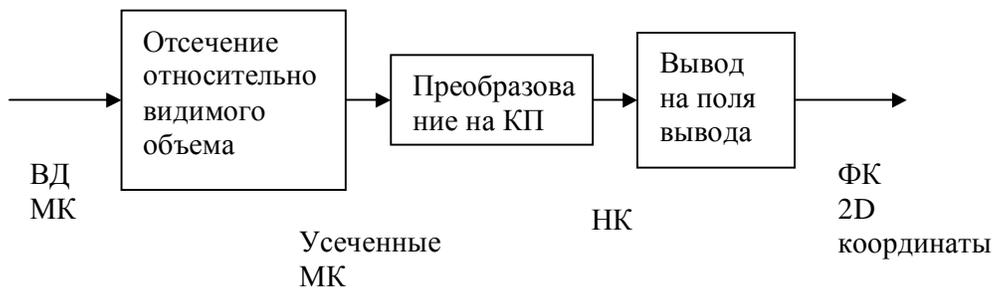
В этой геометрии не бывает параллельных линий.





отображаем на нормализованную плоскость.

**Концептуальная модель процесса 3D вывода.**



$\left. \begin{matrix} ПП \\ КП \end{matrix} \right\}$  точка VP viewplane с 3мя параметрами

- 1) Опорная точка
- 2) Нормаль к картинной плоскости
- 3) Расстояние до картинной плоскости.(КП)

Опорная (.)

Берется в МК .....

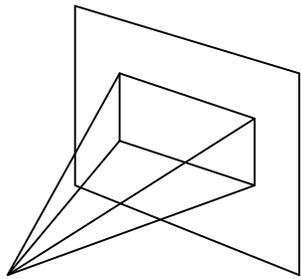
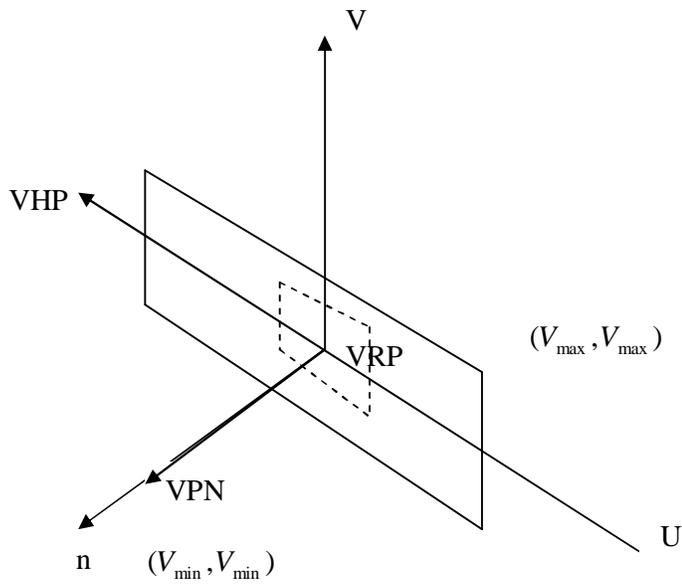
Расстояние до КП – расстояние от опорной (.) в направлении вектора нормали картинной плоскости.

Идеология << PHIGS, GKS,GORE >> стандарты

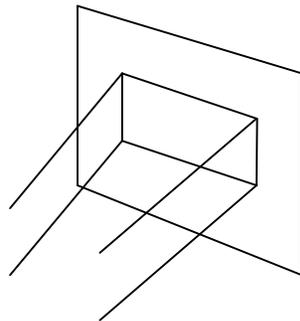
Опорная точка – VRP

Нормаль к картинной плоскости – VPN

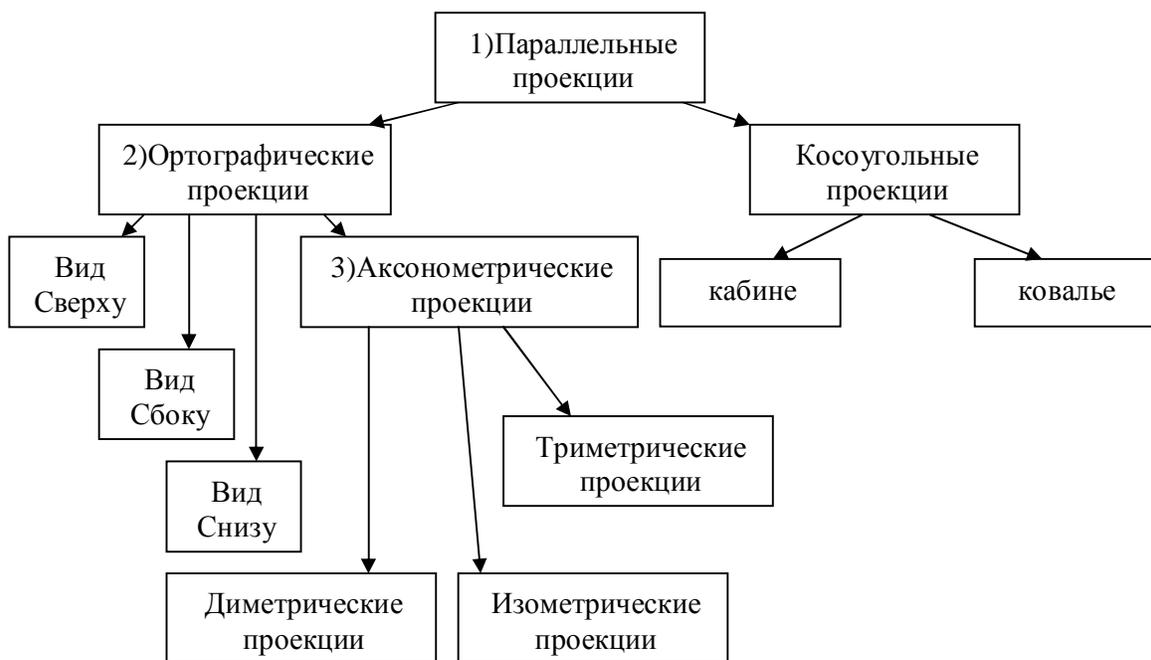
Вектор направления – VHP



Центральные проекции



Параллельные проекции





- Проецирование 3х мерного объекта строится с использованием картинной плоскости и лучей проекций для получения из n – мерной проекции n-1 – ую  
 Причем отметим что лучи проекции проходят через каждую точку объекта.

1) Определяется отношением между направлением проецирования и нормалью к картинной плоскости.

В ортографических проекциях эти эти соотношения совпадают , а в косоугольных наоборот.

2) Ортографические проекции

Используются плоскости не перпендикулярные главным осям.

3) При аксонометрическом проецировании сохраняется параллельность лучей, углы меняются, при этом может быть измерено расстояние до главных осей.

В зависимости от взаимного расположения плоскости проецирования и координатных осей различаются 3 вида аксонометрических проекций:

- Аксонометрия:

все 3 угла между координатными осями и нормалью КП равны между собой

- Димметрия

два угла между нормалью КП и координатными осями равны

- Триметрия

все углы между нормалью КП и координатными осями различны.

Все эти проекции получаются комбинацией поворотов, за которыми следует параллельное проектирование.

При повороте на угол  $\gamma$  относительно оси ординат потом  $-\gamma$   $\gamma$  абсцисс и последующее проецирование вдоль оси аппликат получим матрицу:

$$M_{акс} = \begin{vmatrix} \cos\gamma & \sin\gamma & \sin\gamma & 0 & 0 \\ 0 & \cos\gamma & 0 & 0 & 0 \\ \sin\gamma & -\sin\gamma & \cos\gamma & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{vmatrix}$$

Самостоятельно построить матричную проекцию на плоскость параллельную осям  $x = 0$   $y = 0$  ( $z \neq 0 \neq const$ )

Изометрическое проецирование обладает свойством :

Расстояние по всем 3м главным осям укорачивается с одним и тем же коэффициентом, поэтому возможно с одним и тем же масштаб коэф. по всем 3м осям.

Косоугольная

Сочетает в себе свойства ортографической проекции со свойствами аксонометрии.

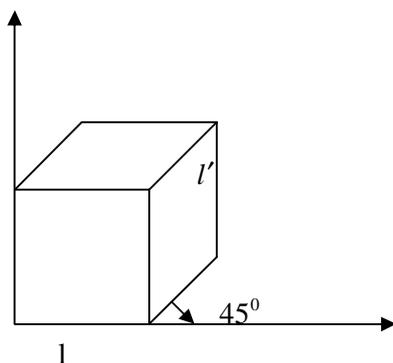
1) Проекционная плоскость перпендикулярна главной координатной оси сторона объекта , параллельная этой плоскости проецирования т. о. , то что может измерить углы и расстояния.

2) Проецирование других сторон позв. с помощью простейших геометрических преобразований определять расстояния и углы (линейное вычисление)

3) Нормаль КП и направление проецирования при косоугольной проекции не совпадают.

Свободное проецирование (ковалье)

Углы в направлении проецирования составляют с плоскостью 45 градусов. В результате проекции отрезка перпендикулярного проецирующей плоскости имеет ту же длину, что и сам отрезок.



Кабинетное проектирование (кабине)

Направление проецирования с плоскостью угол  $\arctg\left(\frac{1}{2}\right)$

При этом соотношение отрезка перпендикулярного КП 1:2 (соотношение длины отрезка и его проекции)

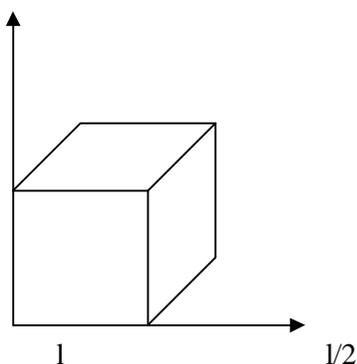
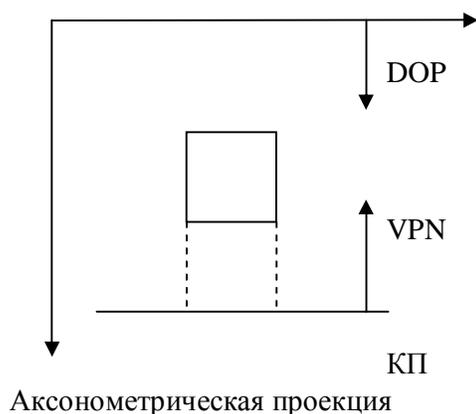


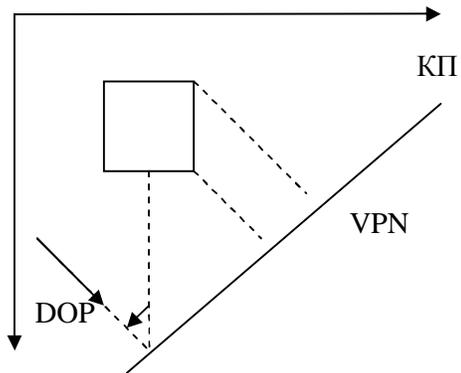
Табл. параллельное проецирование



Многовидовая ортография  
 $\angle DOP$  – direction of projection

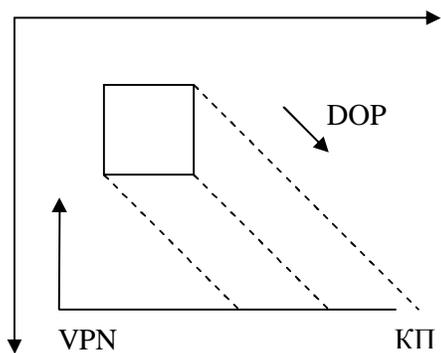
- VPN параллельна всем осям
- DOP параллельна VPN
- Видна вся грань

АксонOMETрическая проекция



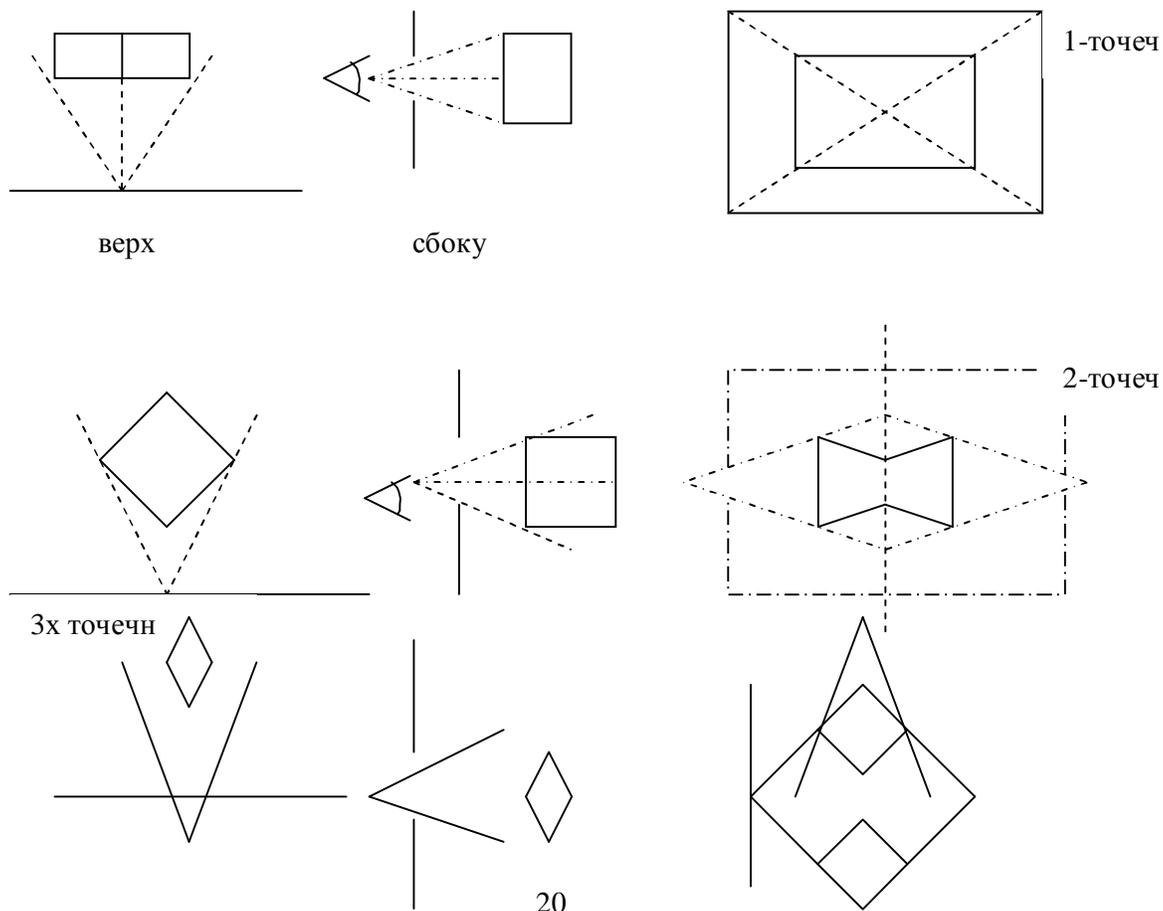
- VPN не параллельна осям координат
- DOP параллельна VPN
- Изменяется размер. Размер – функция угла между нормалью грани и DOP

Косоугольная проекция



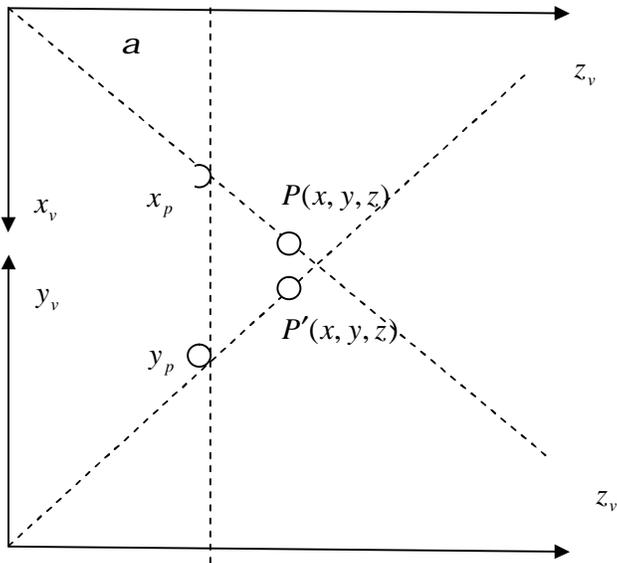
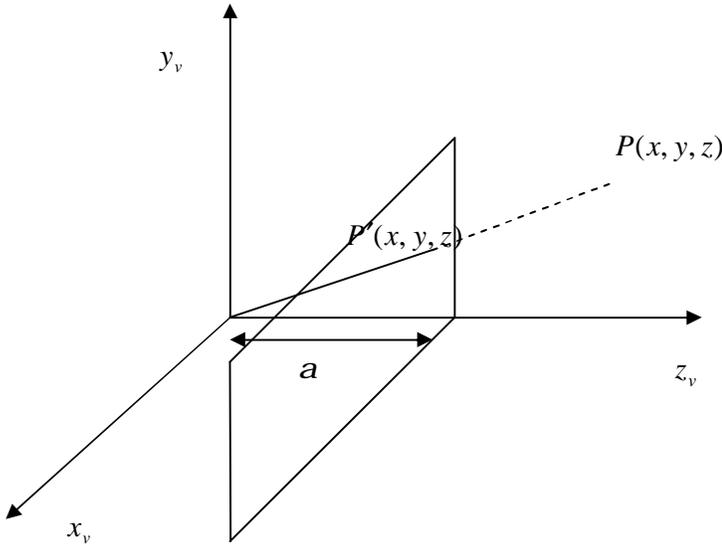
- VPN не параллельна
- DOP не параллельна VPN
- Изменение граней за исключением параллельной КП

Центральное проецирование



Объект подв. центр проецир. из любой точки.

**Математическое описание центральной проекции.**



$$\frac{x_p}{a} = \frac{x}{z}; \frac{y_p}{a} = \frac{y}{z}$$

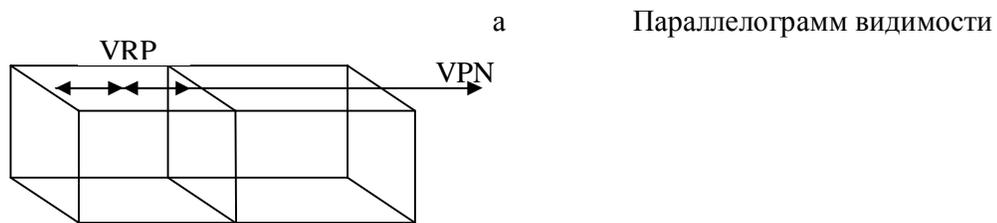
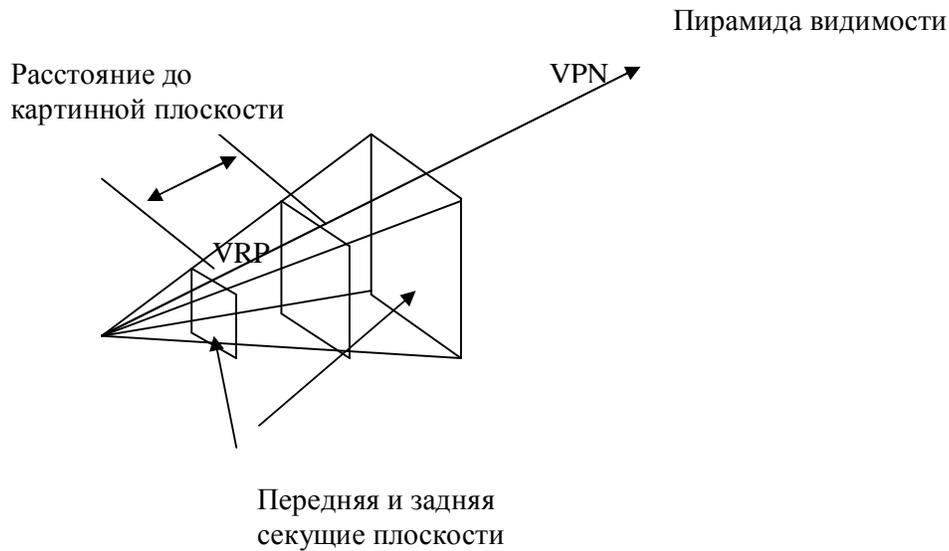
$$x_p = \frac{x}{z/a}; y_p = \frac{y}{z/a}$$

$$M_{\text{центр}} = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/a & 0 \end{vmatrix}$$

$$P \rightarrow |x \ y \ z \ 1|^T$$

$$|x \ y \ z \ w|^T = M_{\text{центр}} P = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/a & 0 \end{vmatrix} \begin{vmatrix} x \\ y \\ z \\ 1 \end{vmatrix}$$

$$|z \ y \ z \ w|^T = |x \ y \ z \ z/a|^T$$



### Интерполяция и аппроксимация кривых и поверхностей

В комплексной графике часто задача восполнения данных (несколько точек через которую и провести кривую). Это и есть интерполяция, частный случай аппроксимации. Если нужно аппроксимировать некоторую  $f(t)$ , то надо искать некоторую конечную сумму.

$$g(t) = \sum_{i=1}^n C_j(i), \text{ где } j(i) \text{ не которые простые } \phi\text{-ции, такие что может выполнить}$$

ограничения наложенные на  $g(t)$

На  $g(t)$  нужно наложить  $n$  ограничений

(Хорошее приближение)  $f(t)$

Интерполяционное ограничение

$$1) g(t_i) = f(t_i) \text{ где } i \in [1, \dots, n]$$

Эти 2 функции должны иметь одинаковые значения в выделенных точках.

2) Это смесь интерполяционных ограничений и ограничений гладкости

а)  $g(t_i) = f(t_i)$

б)  $g'(t_i) = f'(t_i) \quad (i=1, i=k)$

в)  $g(t) \in C^2$

т.е.  $g(t)$  дважды дифференцируема (непрерывна)

В машинной графике нас интересует качество объектов с которыми мы работаем (качество аппроксимации)

Свойства выражаются через внешний вид объекта -> свойства формы.

В машинной графике невозможно записать объект в форме у есть функция от  $x$

Почему?

- Форма большинства объектов не зависит от координат . Если надо восстановить кривую по нескольким точкам, которую мы изм-ем на объект то важнейший факт, это соотношение точек не зависит от системы координат.
- Форма реальных объектов часто имеет вертикальные касательные. Если такой объект выражается через  $y$  от  $x$  то невозможно выявить эту поверхность аналитическим путем.
- В машинной практике мы часто имеем дело с неплоскими и незамкнутыми поверхностями которые не могут быть представлены обычными функциями.

### Явная и неявная параметрическая формула

Явная формула это уравнение вида  $y = f(x)$

Неявная форма : 
$$\begin{cases} ax + by + c = 0 \\ x^2 + y^2 - R^2 = 0 \end{cases}$$

$f(x, y, z) = 0$  уравнение описывающее поверхность

Параметрическая форма

Каждая координата точки, принадлежащая кривой представляется функцией независимой переменной  $t$  которая называется параметром этой кривой.

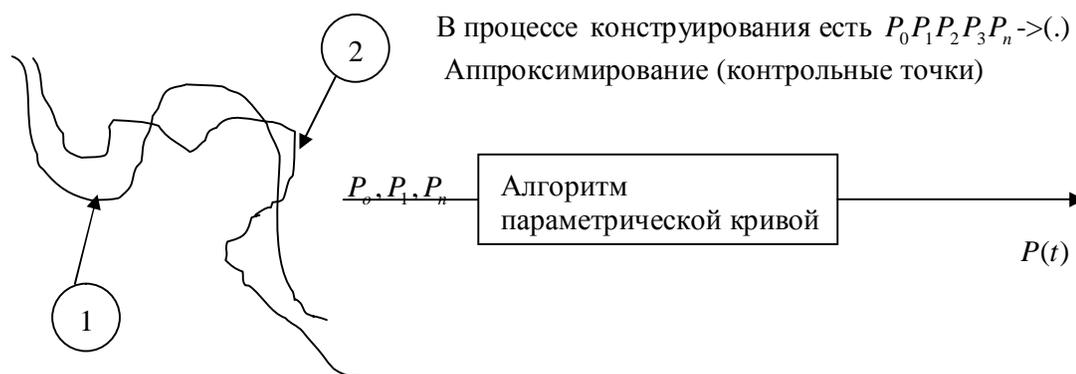
$x, y, z$  запишем как 
$$\begin{cases} x = x(t) \\ y = y(t) \\ z = z(t) \end{cases}$$

$P_c = [x(t), y(t), z(t)]$  матричное представление

Доводы в пользу используемых параметрич.

1. Возможность локального контроля формы
2. Гладкость и непрерывность в математическом смысле
3. возможность математического вычисления производных
4. Устойчивость к малым воздействиям (возмущениям)
5. Возможность использовать простые и значительно высокоскоростные методы закраски

### Сценарий работы с кривыми



- 1) Отмечаем начальные контрольные точки

- 2) Генерируем с помощью алгоритма
- 3) Коррекция кривой исправл. контрольные точки

### Интерполяция в сравнении с аппроксимацией

1) Если мы интерполируем кривую то алгоритм то алгоритм интерполяции возвращает (.) вдоль кривой  $p(t)$  которая в определенный момент времени  $t$  через эти точки проходит

Промежуток между этими точками формируется в виде гладкой кривой .

2) Кривая  $P(t)$  аппроксимируется контрольными точками и осуществляется приближение кривой к этим контрольным точкам.

В качестве полинома нам нужно: чем выше степень полинома тем больше вычислений и наоборот.

Наибольший эффект при применении кубических полиномов.

Параметрические кубические кривые – это кривые в которых  $x$   $y$   $z$  многочлены 3го порядка относительно  $t$

$$\begin{cases} x(t) = a_x t^3 + b_x t^2 + c_x t + d_x \\ y(t) = a_y t^3 + b_y t^2 + c_y t + d_y \\ z(t) = a_z t^3 + b_z t^2 + c_z t + d_z \end{cases}$$

$x'_t$

$$\begin{cases} \frac{dx}{dt} = 3a_x t^2 + 2b_x t + c_x \\ \frac{dy}{dt} = 3a_y t^2 + 2b_y t + c_y \\ \frac{dz}{dt} = 3a_z t^2 + 2b_z t + c_z \end{cases}$$

Эти уравнения задают вектор  $a$   $tg$  угла наклона

$$\frac{dy}{dx} = \frac{dy/dt}{dx/dt}, \frac{dx}{dz} = \frac{dx/dt}{dz/dt}$$

Говорим о кривой 3го порядка т.к. нет кривой более крупного порядка которая бы обеспечивала в точке сопряжения кривых друг с другом непрерывность положения и наклона .

И в тоже время обеспечивалоа прохождение кольцевых точек сегмента через заданные точки

При описании кривых можно пользоваться обозначениями  $C^{(i)}$

Значение  $i$  определяет свойства:

$C^0$  - непрерывны если не имеют разрывов

$C^1$  - кривые непрерывны если кроме того непрерывны и их касательные.

$C^i$  - эта непрерывность обозначает что непрерывность  $f$ -ции и ее первых производных.

- 1) **Метод Эрмита** – в котором задаются положения начальных, конечных точек кривой и вектора к ним
- 2) **Метод Безье** - задается положение конечных точек кривой, а для неявного задания касательных используются 2 другие точки не лежащие на кривой.
- 3) **В- сплайны** - для этого метода конечные точки не лежат на кривой в результате 1 и 2 производные являются непрерывными на кольцевых точках.

## Кривая Эрмита

Кубическая кривая при которой определяются кольцевые точки, и определены касательные вектора этой кривой  $P_1, P_4, R_1, R_4$

Эрмитовый вектор  $G_H = [P_1, P_4, R_1, R_4]$

Использовать его к  $\begin{cases} x(t) = a_x t^3 + b_x t^2 + c_x t + d_x \\ y(t) = a_y t^3 + b_y t^2 + c_y t + d_y \\ z(t) = a_z t^3 + b_z t^2 + c_z t + d_z \end{cases}$  ПИШЕМ ТОЛЬКО ДЛЯ X Для

экономии места!

$$x(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} a_x \\ b_x \\ c_x \\ d_x \end{bmatrix}$$

$$x(t) = [t^3 \quad t^2 \quad t \quad 1] C_x$$

$$x(t) = TC_x$$

$$x(0) = P_{1x} = [0 \quad 0 \quad 0 \quad 1] C_x$$

$$x(1) = P_{4x} = [1 \quad 1 \quad 1 \quad 1] C_x$$

Продифференцируем

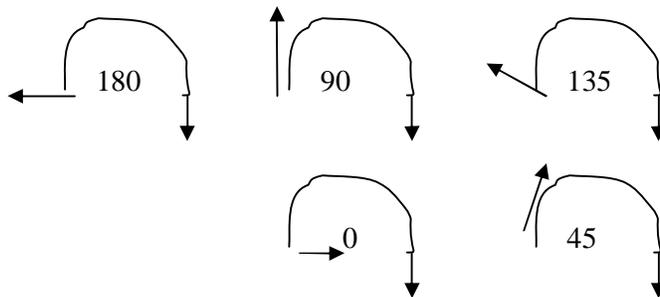
$$x'(0) = R_{1x} = [0 \quad 0 \quad 1 \quad 0] C_x$$

$$x'(1) = R_{4x} = [3 \quad 2 \quad 1 \quad 0] C_x$$

$$x'(t) = [3t^2 \quad 2t \quad 1 \quad 0] C_x$$

$$\begin{bmatrix} P_1 \\ P_4 \\ R_1 \\ R_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix} C_x \rightarrow$$

$$C_x = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_1 \\ P_4 \\ R_1 \\ R_4 \end{bmatrix} = M_{t_1} G_{t_1}$$



## Свойства

- 1)  $C^{(2)}$  - гладкая кривая
- 2) проходит через вершины  $P_0, P_m$
- 3) касательный вектор однозначно определяется через массив вершин  $P$
- 4) Не лежит в выпуклой оболочке порожденной матрицей образованной вершинами  $P$
- 5) Аффинно – инвариантна
- 6) При добавлении хотя бы 1 точки вершины возникает необходимость полного пересчета всех параметрических уравнений.
- 7) Изменение хотя бы 1-й вершины | касательной вектора в концевых точках приводит к изменению формы кривой .

### Кривая Безье.

Определяемая массивом  $P$  называется кривая определяемая следующим векторным уравнением.

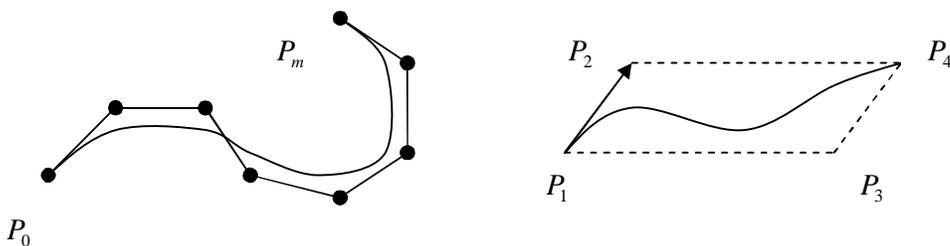
$$r(t) = \sum_{i=0}^m C_m^i t^i (1-t)^{m-i} P_i, \quad 0 < t \leq 1$$

где  $C_m^i = \frac{m!}{i!(m-i)!}$  коэффициенты в различных биномах. Суть этих коэффициентов в вершинах  $P$  – универсальные многочлены Бернштейна; они  $> 0$   $\sum = 1$

### Свойства

- 1) Гладкая кривая
- 2) Лежит в выпуклой оболочке
- 3) Ассиметрично сохраняет форму при перемене порядка вершин в массиве вершин.
- 4) Аффинно – инвариантна
- 5) Повторяет опорную ломаную
- 6) Если точки  $P_m, P_0$  лежат на одной прямой (горизонтальной| вертикальной), то кривая совпадает с ломаной.
- 7) Степень функционального многочлена напрямую связана с количеством вершин (на 1 больше) и растет при его увеличении)
- 8) При добавлении хотя бы одной вершины возникает необходимость пересчета параметрического уравнения кривой.
- 9) Изменение 1й вершины приводит к заметному изменению кривой Безье.
- 10) Т.к. поведение кривой Безье определяется не только набором вершин, но и параметрическими коэффициентами, их называют весовыми коэффициентами (параметры формы)
- 11) Формой кривой Безье можно управлять изменяя весовые коэффициенты| параметрические формы.

Опорная ломаная.



При  $m = 3$  получим универсальную кубическую кривую Безье ( универсальные вершины  $(P_0, P_1, P_2, P_3)$ ) и универсальное параметрическое уравнение

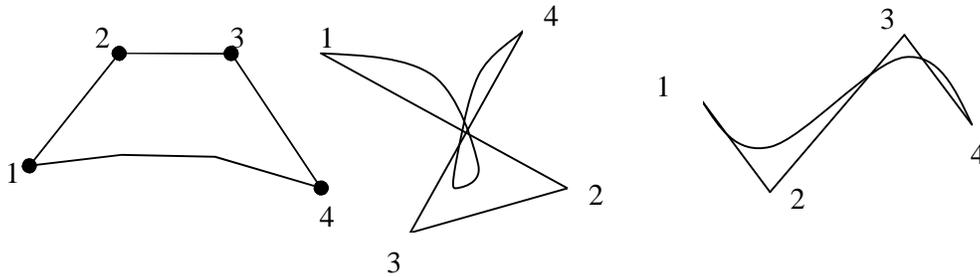
$$f(t) = (((1-t)P_0 + 3tP_1)(1-t) + 3t^2P_2)(1-t) + t^3P_3 \quad (0 < t \leq 1)$$

$$f(t) = PM_bT$$

$$f(t) = \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix}, \quad P = \begin{vmatrix} P_0 & P_1 & P_2 & P_3 \end{vmatrix} = \begin{vmatrix} x_0 & x_1 & x_2 & x_3 \\ y_0 & y_1 & y_2 & y_3 \\ z_0 & z_1 & z_2 & z_3 \end{vmatrix}$$

$$M_b = \begin{vmatrix} 1 & -3 & 3 \\ 0 & 3 & -6 \\ 0 & 0 & 3 \\ 0 & 0 & 0 \end{vmatrix} \quad T = \begin{vmatrix} 1 \\ t \\ t^2 \\ t^3 \end{vmatrix} \quad \text{где } M_b \text{ – базисная матрица кубической кривой Безье}$$

Как зависит форма кривой Безье от изменения положения точек.



### **В – сплайны**

1946 г.

Задаются коэффициенты сравнительно небольшого числа опорных точек лежащего на поверхности и через точки проводятся плавные поверхности.

Кривые представляются в виде кубических В – сплайнов. В общем сплайн может проходить через любое количество управляющих точек.

Обеспечивается направленность 1-й и 2-й производной в конечных точках.

$$x(t) = TM_sG_s$$

$$M_s = \frac{1}{6} \begin{vmatrix} 1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{vmatrix}$$

$$G_s^i = \begin{vmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ P_{i+2} \end{vmatrix}$$

- геометрический коэффициент В – сплайна ( $2 < i \leq m - 2$ )

В – сплайн



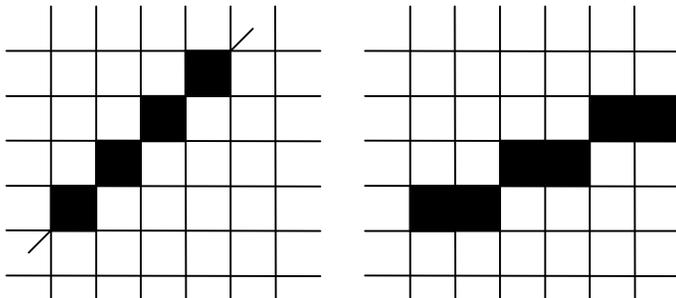
### Свойства кривой В-сплайна

- 1)  $C^{(2)}$  - гладкой
  - 2) не проходит ни через 1 точку массива
  - 3) лежит в выпуклой оболочке порождаемой этим массивом
  - 4) повторяет опорную ломаную
  - 5) если опорные точки находятся на прямой, то В – сплайн лежит на этой прямой.
  - 6) если опорные точки лежат в одной плоскости ( компланарные), то составная (В-сплайновая) кривая лежит в этой плоскости.
  - 7) изменение одной из вершин приводит только к изменению части кривой
  - 8) При добавлении 1 вершины возникает необходимость пересчета 1-х элементов кривых (параметрическое уравнение)
  - 9) аффинно инвариантна
  - 10) проективно инвариантна ( проективные преобразования не влияют на форму нашей кривой)
  - 11) существуют весовые коэффициенты
- Д\З кубические поверхности.

### Алгоритмы растровой графики

#### Генерация линий

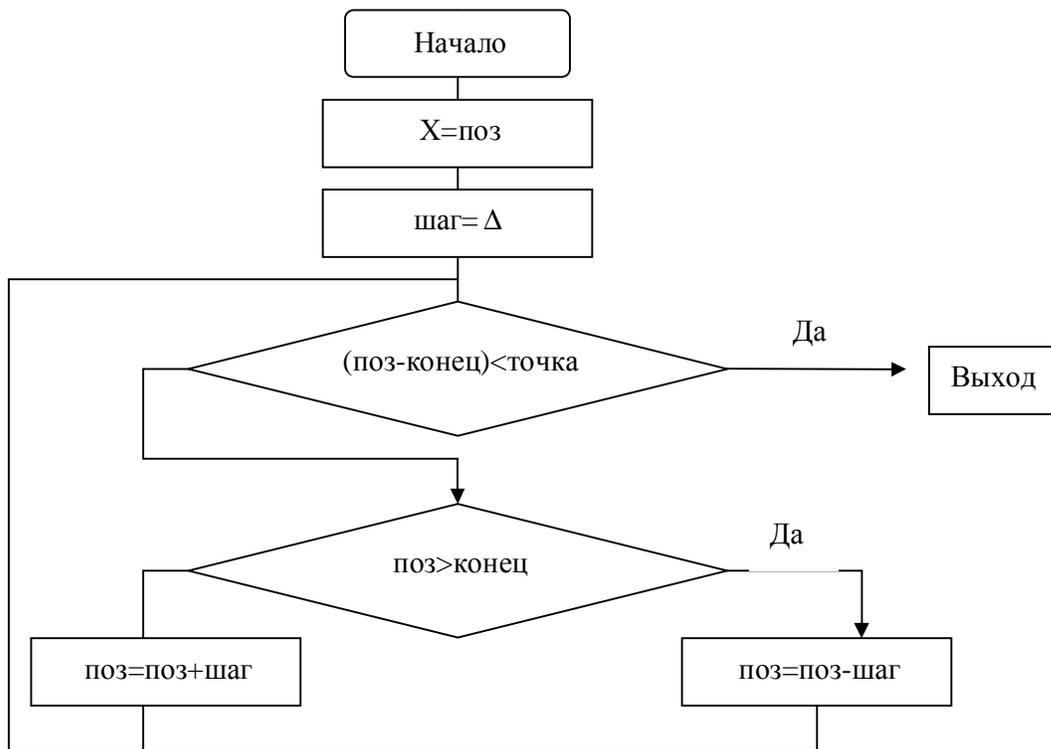
- 1) Генерируемый отрезок должен быть прямым
  - 2) Начинаться и заканчиваться в заданных точках.
  - 3) Яркость отрезка по всей длине должна быть одинаковая и не зависеть от угла и наклона
  - 4) Алгоритм рисования линий должен быть быстрым
- Все 4 условия одновременно не могут быть соблюдены



#### Алгоритмы для генерации линий.

- 1) Находят приближенную длину отрезка.
- 2) Стараются свести вычисляемые операции к минимуму. Стараются прийти к целостным данным.
- 3) Перевести реализацию этих алгоритмов на аппаратный уровень.

Простой пошаговый алгоритм.



### Цифровой дифференциальный анализатор.

Разложение отрезка в растр (генерация отрезка) состоит в решении ДУ.

$$\frac{dy}{dx} = const$$

$$\frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1}$$

$y_i$  - начальное значение для очередного шага вдоль отрезка.

$$y_{i+1} = y_i + \Delta y$$

$$y_{i+1} = y_i + \Delta x (y_2 - y_1) / (x_2 - x_1)$$

$x_1, y_1$  - концы отрезка

$x_2, y_2$

Процедура разложения в растр по методу ЦДА

1) Предполагается, что концы отрезков не совпадают

2) Аппроксимируем длину отрезка.

if  $abs(x_2 - x_1) \geq abs(y_2 - y_1)$

then  $Длинна = abs(x_2 - x_1)$

else  $Длинна = abs(y_2 - y_1)$

end

3) Полагаем большее из приращений

$x_2 - x_1$  или  $y_2 - y_1 = 1$

$\Delta x = (x_2 - x_1) / Длинна$

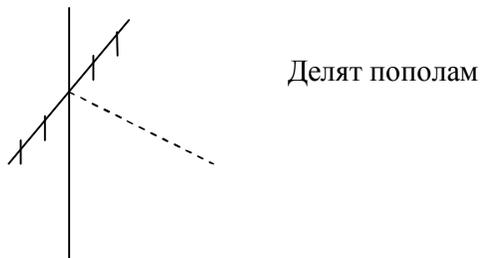
$\Delta y = (y_2 - y_1) / Длинна$

Округляем величины но не отбрасываем дробную часть.

Воспользуемся функцией sign:

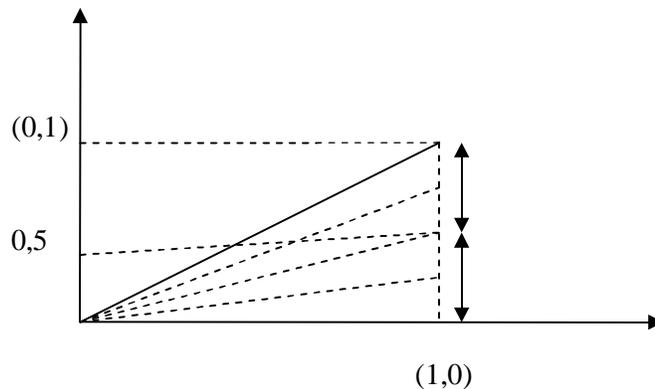
$x = x_1 + 0,5 * sign(\Delta x)$   
 $y = y_1 + 0,5 * sign(\Delta y)$   
 4) Начало основного цикла

$n = 1$   
*while* ( $i \leq \text{длина}$ )  
*Plot(Integer(x), Integer(y))*  
 $x = x + \Delta x$   
 $y = y + \Delta y$   
 $i = i + 1$   
*end*  
 Метод деления отрезка пополам.



Алгоритм Безенхейма.

$x | y$  зависят от углового коэффициента изменяется на единицу (в зависимости от коэффициента)



Координата другой зависит от расстояния между действующим положением отрезка и ближней координатой сетки.

### Растровая развертка в реальном времени

2 типа информации:

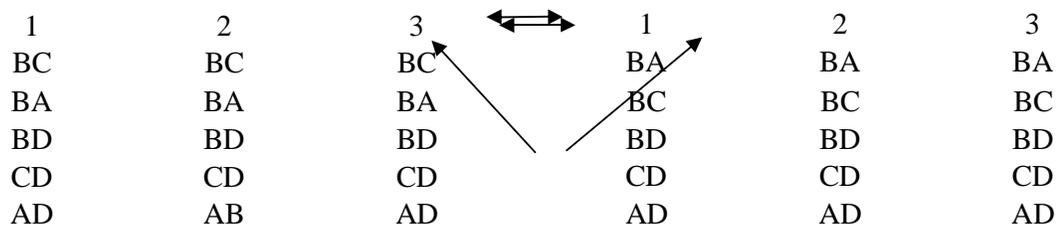
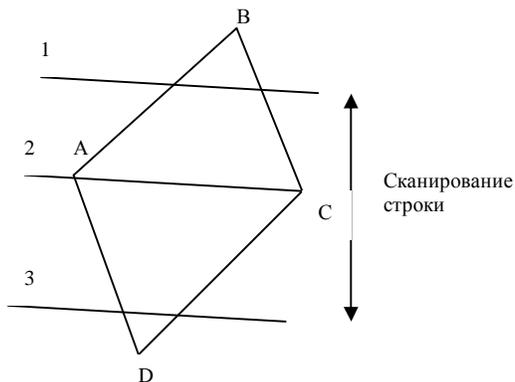
- 1) Виртуальные атрибуты (свет, цвет, оттенок)
- 2) геометрические атрибуты (координаты  $x, y$ , текстовая информация)

Информация будет упорядочена.

Вычисляется интенсивность вывода на экран.

Сложность в том : скорость вывода на экран.

Простейший алгоритм построчного сканирования.

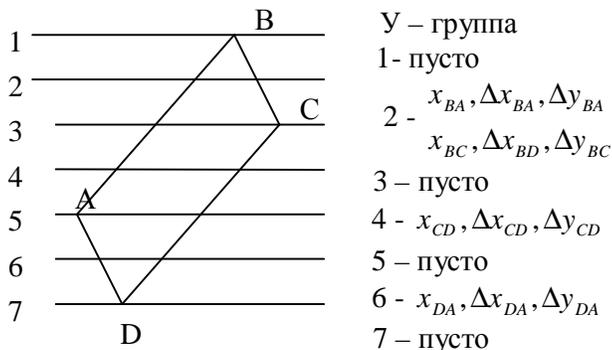


2 формы сортировки

Порядок сортировки начинаемый с  $u$  влияет на размер списка активных ребер. Либо обрабатывается больше информации чем нужно для данной строки.

Развертка с групповой сортировкой.

Проблема борьбы с избыточной информацией решается за счет введения дополнительных данных. Вначале выполняется групповая сортировка по  $u$  всех отрезков изображения.



8 – пусто

При групповой сортировке по  $u$  образуются области памяти или группы. При просмотре дисплейного списка заносится информация в группу, соответствующей наибольшей  $u$  координате.

$x$  – координата точки пересечения со сканирующей строкой.  
 $\Delta x$  - изменение координаты  $x$  при переходе одной с другой  
 $\Delta y$  - число сканирующих строк пересекаемых данным отрезком.

**Алгоритм**

- 1) Список активных ребер для каждой сканирующей строки формируется добавлением информации из каждой  $y$  группы соответствующей сканирующей строке.
- 2) Координаты  $x$  точек пересечения сортируется в порядке сканирования, а ребра из списка активных ребер преобразуются в растровую форму.
- 3) Для каждого отрезка список активных ребер ( $\Delta y$ ) число пересечения со сканирующей строкой уменьшается на единицу. Если  $\Delta y$  отрицательный то отрезок исключается из списка.

- 4) Для каждого отрезка координата  $x$  получается добавлением

$$x_{текущ} = x_{перес.предыд} + \Delta x$$

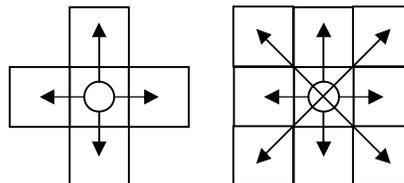
список активных ребер

сканирование строки 3 
$$\begin{aligned} &x_{BA} + x_{\Delta BA}, \Delta x_{BA}, \Delta y_{BA} - 1 \\ &x_{BC} + x_{\Delta BC}, \Delta x_{BC}, \Delta y_{BC} - 1 \end{aligned}$$

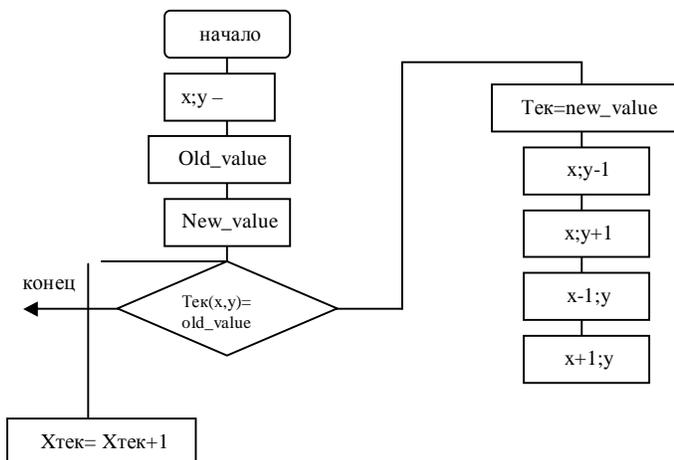
сканирование строки 5 
$$\begin{aligned} &x_{BA} + 3x_{\Delta BA}, \Delta x_{BA}, \Delta y_{BA} - 3 \\ &x_{CD} + x_{\Delta CD}, \Delta x_{CD}, \Delta y_{CD} - 1 \end{aligned}$$

сканирование строки 7 
$$\begin{aligned} &x_{CD} + 3x_{\Delta CD}, \Delta x_{CD}, \Delta y_{CD} - 3 \\ &x_{AD} + x_{\Delta AD}, \Delta x_{AD}, \Delta y_{AD} - 1 \end{aligned}$$

тип гранично определенный  
 4-рех или 8-ми связные пиксели  
 4-х связный связан в 4-х направлениях  
 8-ми связный соответственно в 8-ми



(используются реже – сложный алгоритм многопроходный)  
 Простой рекурсивный алгоритм закрашивания по затравке



О ступенчатости : (classing)

Если есть: ступенчатость

### Отсечение

В глобальном смысле в широком спектре задач применяется алгоритм отсечения.

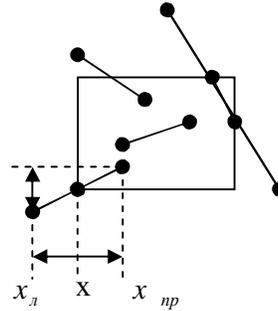
Процесс отсечения – это выделение определенной базы данных. Применяются в алгоритмах удаления невидимых поверхностей, в методах построения теней, в методах Построения текстуры ( фактуры). Выделяется 2-х мерное отсечение и 3-х мерное.

Алгоритмы могут реализовываться как программным так и аппаратным способом.

#### 2-х мерное отсечение.

Определить отрезки попадающие  
внутри прямоугольника

$$x_l < x < x_{np}$$



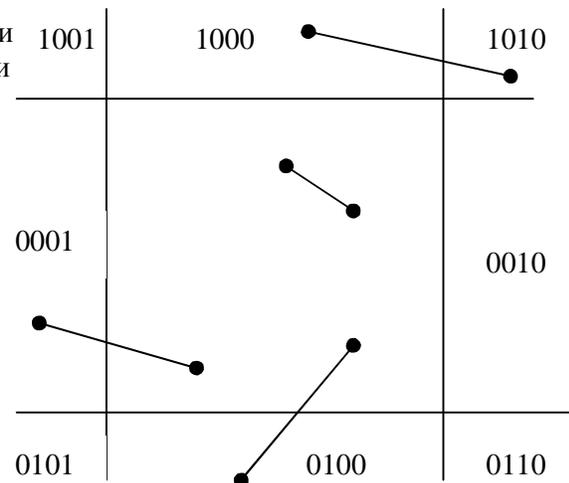
Отсечение

Способы выбора простоты решений => 2 вида

- 1) Алгоритмы использующие (группы алгоритм. Отсечения) кодирование концов отрезка или всего отрезка.
- 2) Алгоритм использует параметрическое описание отрезка и самого ока отсечения.
  1. Алгоритм деления отрезка пополам CS Койна – Сазерленда; FS –алгоритмы(fast clipping)
  2. Алгоритм Кирус-Бека.

#### *Алгоритм Козна – Сазерленда.*

Эффективен если есть отрезки внутри  
и вне и мало отрезков на пересечении



Делится на 8 областей, каждая кодируется 4-х разрядным кодом.

Внутренняя область: 0000

1-ца в разряде говорит о том к какой области принадлежит отрезок

1-ца в первом разряде => над окном

1-ца во втором разряде => под окном

1-ца в третьем разряде => над правее

Шаги определяются нахождением положительных отрезков.

- 1) Сами коды обоих концевых точек отрезка => отрезок в окне
- 2) Если логическое И кодов обоих концевых отрезков не = 0 => отрезок целиком вне окна
- 3) Если логическое И кодов обоих концов = 0 => этот подоз- ный отрезок может быть частично видим или целиком невидим т.е. на границе => определяются точки пересечения с окном.

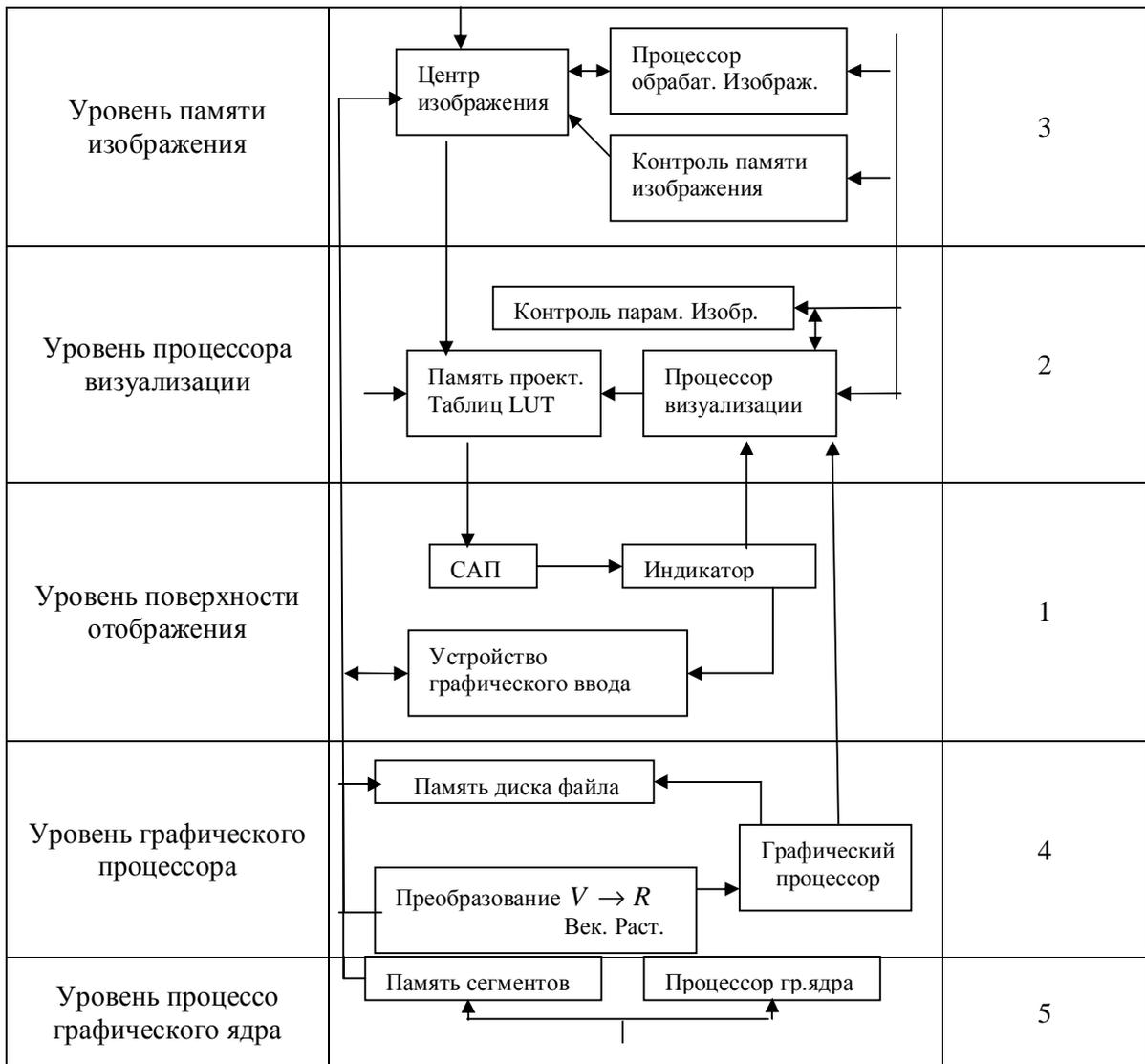
FC – алгоритм.

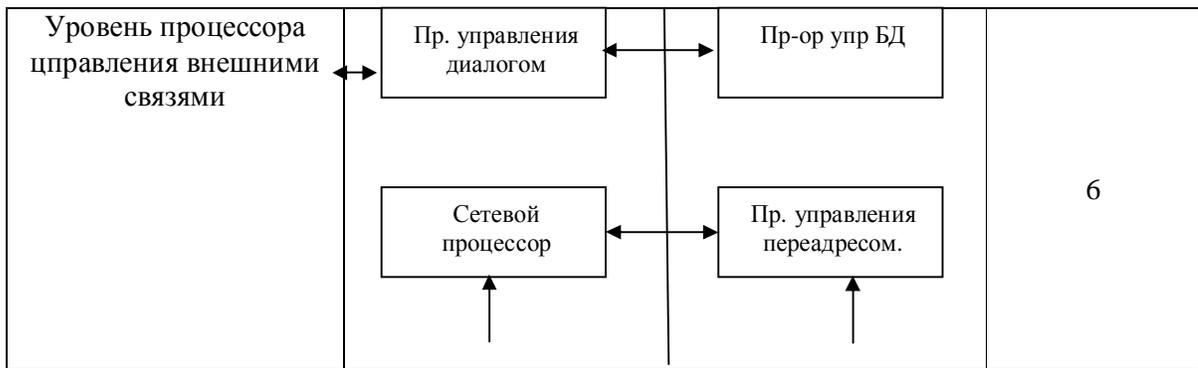
Делится на 5 областей.

Кодировка самого отрезка 16 – ричным кодом , а не как в предыдущем алгоритме концевых точек.

Алгоритм работает по определенной принадлежности отрезка к областям .

**Многоуровневая логическая структура растровой графической системы.**





Уровень ПО: На него поступает видео сигнал, чтобы преобразовать его и на индикатор надо ЦАП (цифровое аналоговое преобразование)

Частота сигнала поступающего от ЦАП определяется частотой высвечивания сигнала на цифровой трубке ⇒ индикатор – это задачник.

*LUT* - позволяет изменять данные выводимые на устройство изображения без изменения изображения находящегося в памяти.

Процесс обратного изображения осуществляет все преобразования изображения которые существуют в растровой форме.

4-ий уровень: определяет три блока.

Память дисплейного файла – содержит изображение представленное в удобном виде для преобразования  $V \rightarrow R$ .

На выходе векторная информация. Базовый уровень (4-ый).

Графический процессор осуществляет (проекционное преобразование), геометрическое преобразование на уровне векторов (аффинные)

Графический процессор осуществляет управление.

Уровень 5 информация (изображение) хранится в виде последовательности коменд.

Пам. сегм. структурирует наши графические данные в соответствии с нашими стандартами.

Индикатор (дисплей, мониторы)

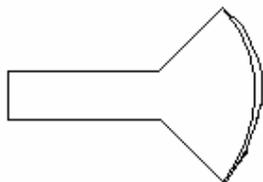
Этот блок один из основных в графической системе

2-е группы индикаторов

1) растровая

2) векторная

2. На базе ЭЛТ используют и монитор



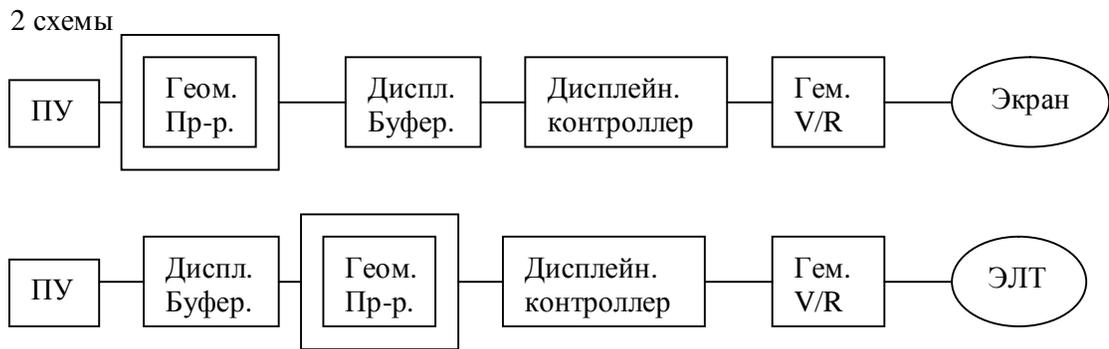
- с коротким временем свечения

Дисплей с произвольным временем сканирования (сканирует очень часто min 30-40 Нсек)

Для работы векторного дисплея.

1) Дисплейный буфер - вся информация которого должна быть выведена.

2) Дисплейный – контроллер (редактирует инф. индикатор)



Первая схема используется при буферизации.

Геометрический процессор не успевает.

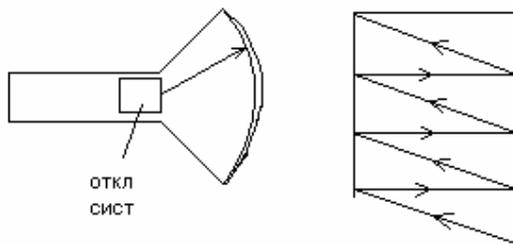
Сложные изображения за один цикл регенерации  $\Rightarrow$  дисплей файл делится на 2-е части.

Вторая схема

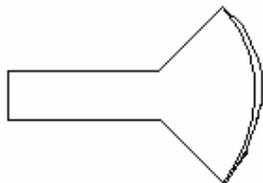
Все преобразования выполняются на лету за один цикл генерации.

### Растровый дисплей.

В виде матрицы: где находятся значения информации с которой будем работать.



30.04.04



- 1) Векторный дисплей представляет собой шторм высокого качества.
- 2) Динамическое преобразование осуществляется в реальном времени.
- 3) Высокая интерактивность.

В 80х годах 6 тысяч векторов при разрешении 2048x2048

Применяются в системах автоматического проектирования.

Обеспечение наивысшего качества при низкой скорости.

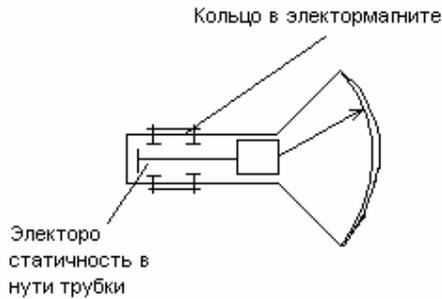
«+» у растровой: представление цветовых возможностей и быстрое представление возможность смешивания с телевиденья. Что позволяет синтезировать новый изображения.

Способность немерцающего изображения практически неограниченна.

Возможность одновременной реализации. Компьютерной графики и векторного изображения, обработки его.

Кольцо в электромагните

электро статичность в нути трубки



#### Фокусирующая система

- 1) с электростатическим отключением
- 2) с магнитным отключением

Сравним их: При 2-ом способе конструкция упрощается. Проще сфокусировать луч.

Получение более яркого изображения за счет повышения напряжения.

Но более широкое распространение электростатические

Более высокая скорость генерирования изображения.

- выше точность позиционирования луча

- Отсутствие изображения прямоугольной сети.

Уровень ПО: На него поступает видео сигнал, чтобы преобразовать его и на индикатор надо ЦАП (цифровое аналоговое преобразование)

Частота сигнала поступающего от ЦАП определяется частотой высвечивания сигнала на цифровой трубке  $\Rightarrow$  индикатор – это задачник.

*LUT* - позволяет изменять данные выводимые на устройство изображения без изменения изображения находящегося в памяти.

Процесс обратного изображения осуществляет все преобразования изображения которые существуют в растровой форме.

4-ий уровень: определяет три блока.

Память дисплейного файла – содержит изображение представленное в удобном виде для преобразования  $V \rightarrow R$ .

На выходе векторная информация. Базовый уровень (4-ый).

Графический процессор осуществляет (проеекционное преобразование), геометрическое преобразование на уровне векторов (аффинные)

Графический процессор осуществляет управление.

Уровень 5 информация (изображение) хранится в виде последовательности коменд.

Пам. сегм. структурирует наши графические данные в соответствии с нашими стандартами.

#### Индикатор (дисплей, мониторы)

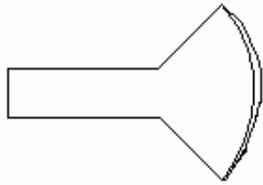
Этот блок один из основных в графической системе

2-е группы индикаторов

1) растровая

2) векторная

2. На базе ЭЛТ используют и монитор

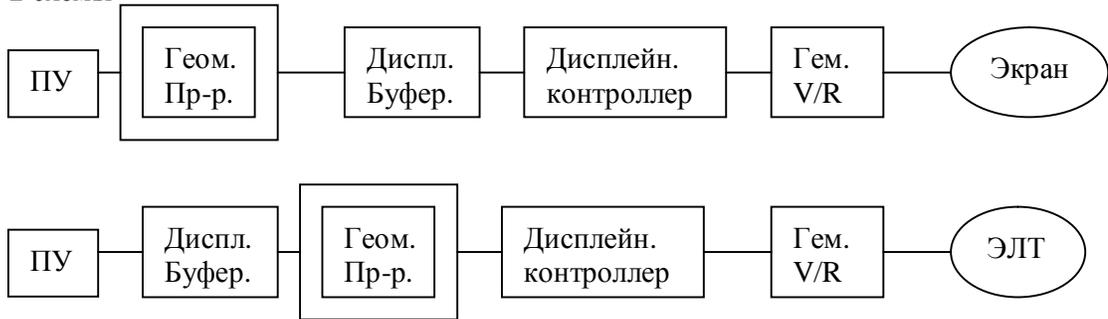


- с коротким временем свечения

Дисплей с произвольным временем сканирования (сканирует очень часто min 30-40 Нсек)  
Для работы векторного дисплея.

- 1) Дисплейный буфер - вся информация которого должна быть выведена.
- 2) Дисплейный - контроллер (редактирует инф. индикатор)

2 схемы



Первая схема используется при буферизации.

Геометрический процессор не успевает.

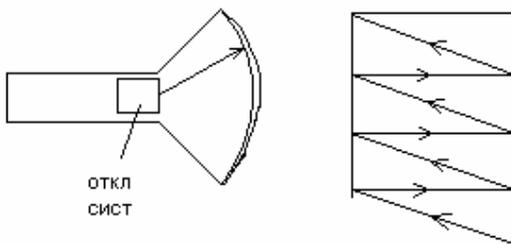
Сложные изображения за один цикл регенирации => дисплей файл делится на 2-е части.

Вторая схема

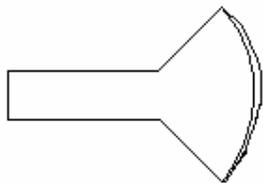
Все преобразования выполняются на лету за один цикл генирации.

Растровый дисплей.

В виде матрицы: где находятся значения информации с которой будем работать.



30.04.04



- 1) Векторный дисплей представляет собой штор высшего качества.
  - 2) Динамическое преобразование осуществляется в реальном времени.
  - 3) Высокая интерактивность.
- В 80х годах 6 тысяч векторов при разрешении 2048x2048

Применяются в системах автоматического проецирования.

Обеспечение наивысшего качества при низкой скорости.

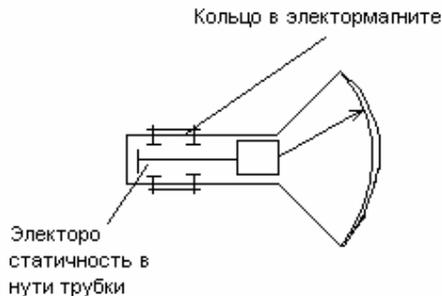
«+» у растровой: представление цветовых возможностей и быстрое представление возможность смешивания с телевиденья. Что позволяет синтезировать новый изображения.

Способность немерцающего изображения практически неограниченна.

Возможность одновременной реализации. Компьютерной графики и векторного изображения, обработки его.

Кольцо в электромагните

электро статичность в нути трубки



Фокусирующая система

1) с электростатическим отключением

2) с магнитным отключением

Сравним их: При 2-ом способе конструкция упрощается. Проще сфокусировать луч.

Получение более яркого изображения за счет повышения напряжения.

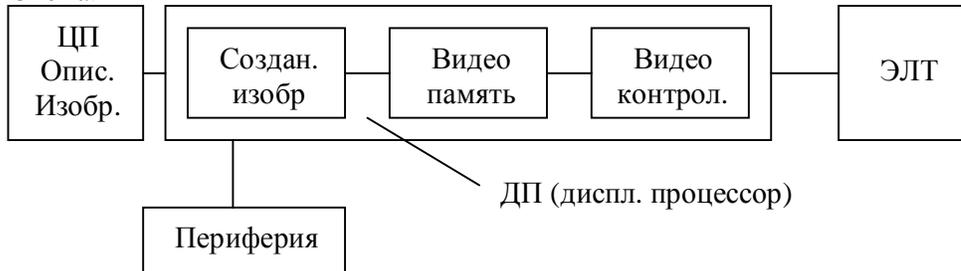
Но более широкое распространение электростатические

Более высокая скорость генерирования изображения.

- выше точность позиционирования луча

- Отсутствие изображения прямоугольной сети.

Схема.



Видеоконтроллер осуществляет формирование изображения на экране путем его строчного вывода. При этом контроллер точно соблюдает время развертки строки которая либо соответствует телевизионным кадрам либо выбирает по заданным параметрам.

{  
- Ширина полосы  
- Частота строк и кадров  
}

Настраиваемые параметры

Может быть три вида развертки изображения.

- Через сторчная

- Строчная

- Комбинация черезстрочной и строчной

Для сокращения ширины полосы пропускания строится через черезстрочная развертка.



Изображение делится на 2-е картинки имеющее половинное вертикальное разрешение.

- 1) включает все четные
- 2) включает все не четные.

Такой способ в телевизионном вещании.

Время которое необходимо для вывода на экран картинки определяется частотой кадров, и так же скоростью обратного хода луча.

Пример:

Для стандарта телевидения 625 строк.

Длительность строки 65 м сек. Где 20% это время обратного хода луча.

Важнейший параметр время обработки 1-ого пиксела, связано со временем обработки 1-ой строки.

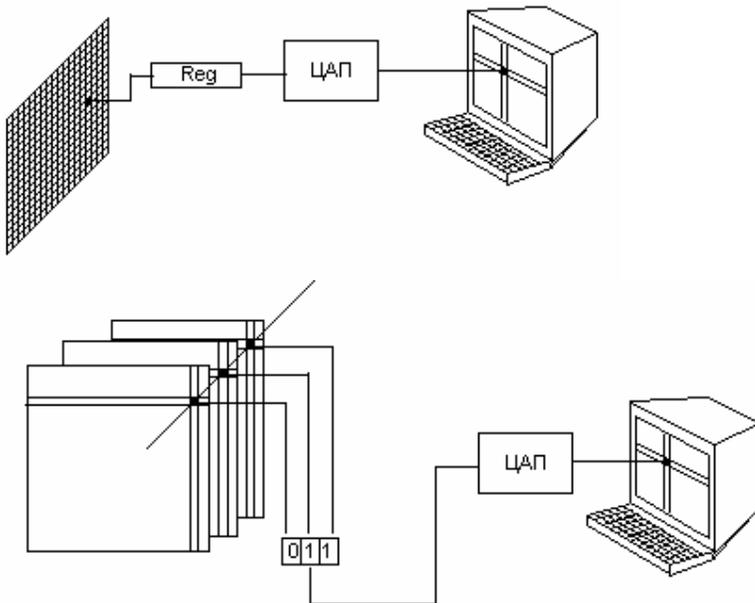
$$t_s = \frac{1 / \text{част.кадров} - t_{\text{обр.х.луча}}}{\text{число вид строк в пиксилх}}$$

$$t_h = \frac{t_s - t_{\text{обр.х.строки развертки}}}{\text{число пикс в строке}}$$

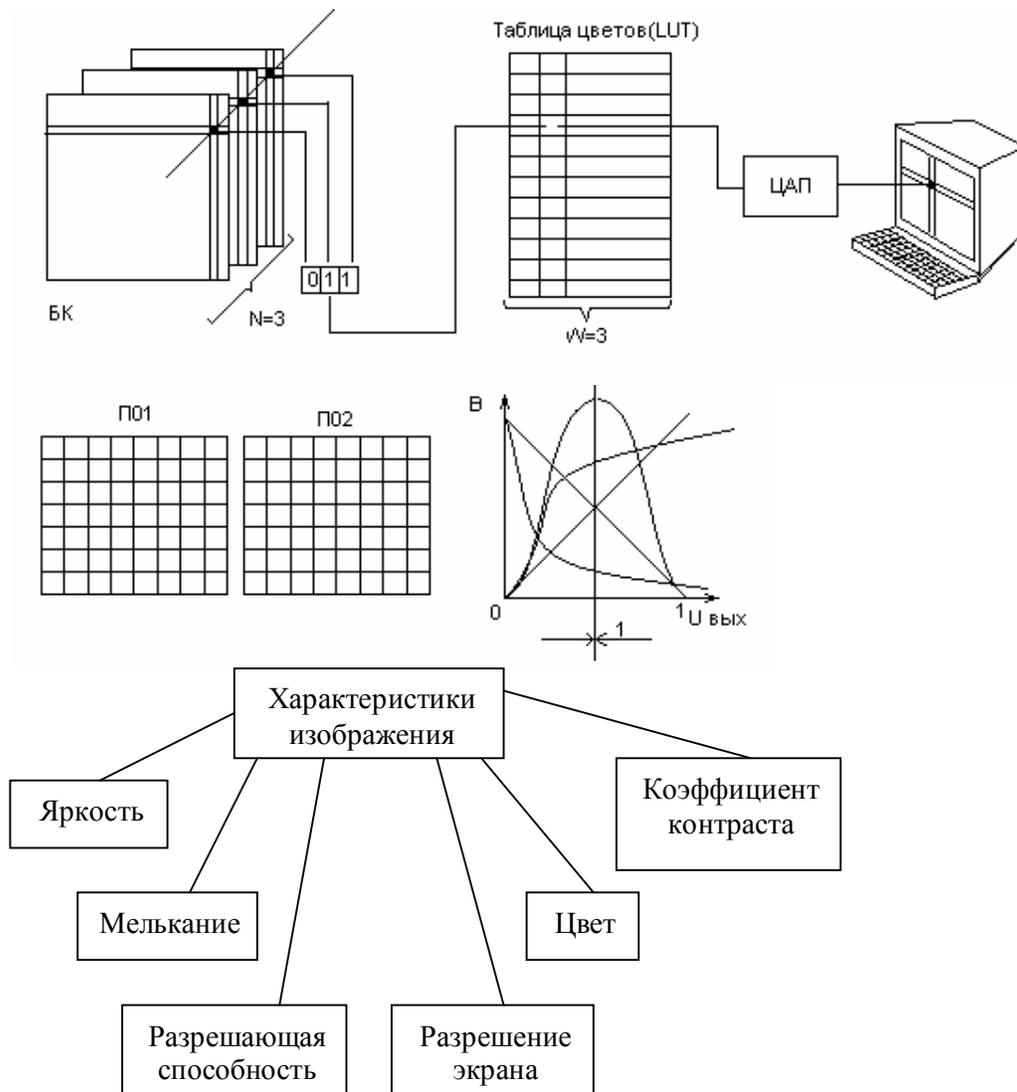
2-е группы дисплеев.

-Полноцветные (для пикселя хранится свое значение RGB пиксели имеют по одному байту на составляющую)

-Дисплей с просматриваемой таблицей (значение пиксела не сразу передается, а проходит через дополнительную таблицу в которой значение операндов различных преобразованных адресов).



Трехбитовая плоскость



Яркость – интенсивность свечения в строку наблюдателя 0.1-1 яркость изображения.  
 Воспринимаемая частота зависит от яркости и частоты мелькания.  
 Частота мелькания неразличима когда подходим к критической высокой частоте мелькания. Если частота мелькания больше чем ФКМ то яркость = усредненной яркости.

$$F_{км} = a L_{ср} + v$$

где  $a = 15,5$  для ярких и цветных изображений

Закон Ферри Поттер.

$L_{ср}$  - средняя яркость изображения.

С ростом углового коэффициента ФКМ растет.

Параметрическое зрение более сильное чем центральное.

Коэффициент контраста.

Отражает фундаментализм представления о свете.

Контраст восприятие качественного и количественного нито разню 2 частей поля видимого зрения.

Глаз различает 1 приращение яркости.

Разрешающая способность.

Характеризует в сантиметрах размеры которые мы можем выводить на экран.

Максимальная разрешающая способность определяется размером рецепторов их расположением в сетчатке (1 микрон)

Острота зрения снижается от расположения оптической оси.

В дисплее разрешающая способность 2-микрона.

Размер экрана Определяет размер выводимого изображения

Цвет.

Цвет параметр дисплея, системы управления которым позволяет формировать изображение.

1) Светность или яркость *Lighthess*

2) Цветовой ток

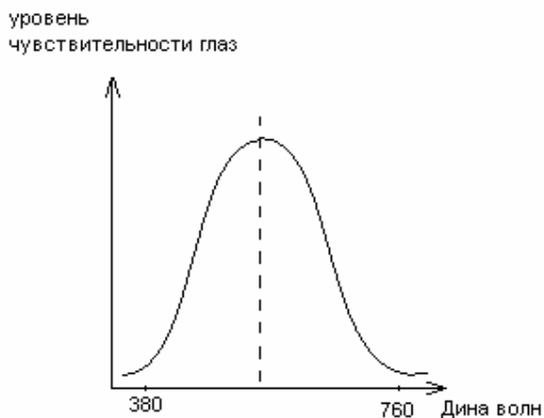
3) Насыщенность цвета .

Цвет }  
Свет }

Станд

Природа света есть электромагнитная.

Частотный спектр воспринимаемый нашим глазом.



Разрешающая способность глаза зависит от света (черно-белый и зеленый)

Зеленый наиболее приятно глазу.

Колориметрия – наука о цвете

Цвет – существуют законы Гросмона

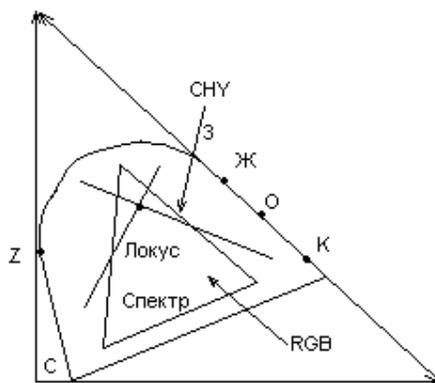
1) Цвет имеет 3х компонентную природу: насыщенность цветовой, яркость (*HSL*)

2) Все составляющие линейно зависимы

$cI \cdot eR + bB + gG$   $c, r, b, g \neq 0$

3) Цветовое пространство непрерывно.

Существует locus цветов или диаграмма цветности



(не спектральные и все вне locus спектре)

Пурпурные цвета (не входят а спектр)

(●) Б (0,33, 0,33)

(●) черного цвета – нет

МКО – международный колориметрический стандарт 1931

Есть RGB

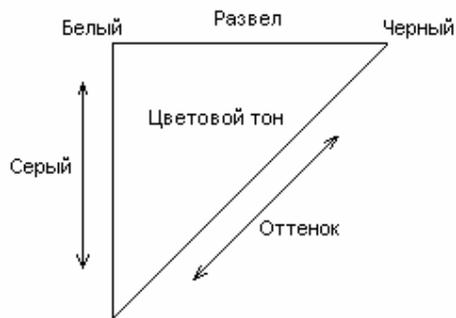
СМУ

Цвет – разделяется монохроматический и ахроматический (только полутона белого - черного)

Визуально

16  
32  
64 - количество на границе  
128  
256 ↓

Монохроматический



развел → добавл. Белой сост.

Оттенок → добавл. Черн соств.

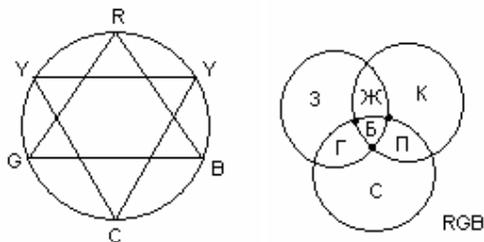
Станд

В устройствах отображения типа монитора.

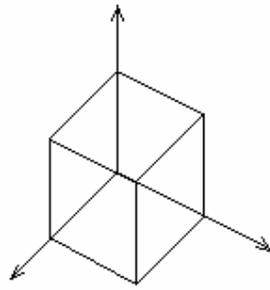
При смешивании дополнительного цвета зеленый и синий – голубой

Зеленый и красный – пурпурный

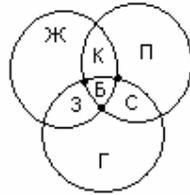
Красный и зеленый – желтый



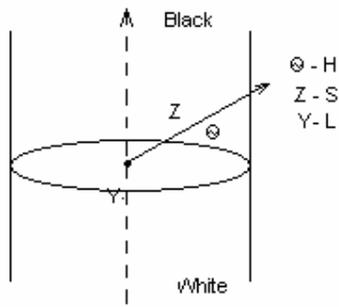
Чистый цвет получить сложно



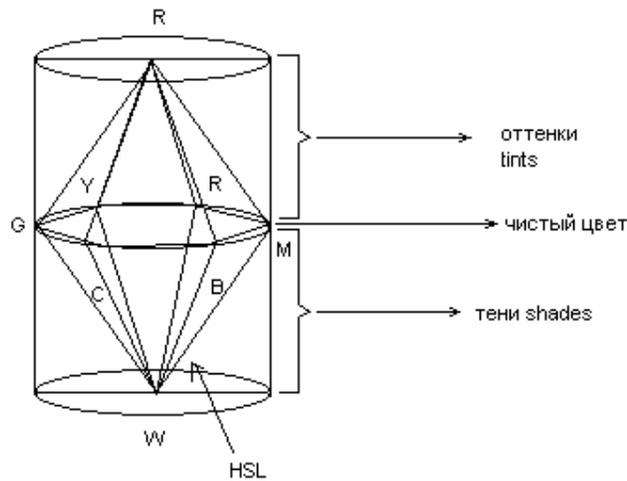
CMY

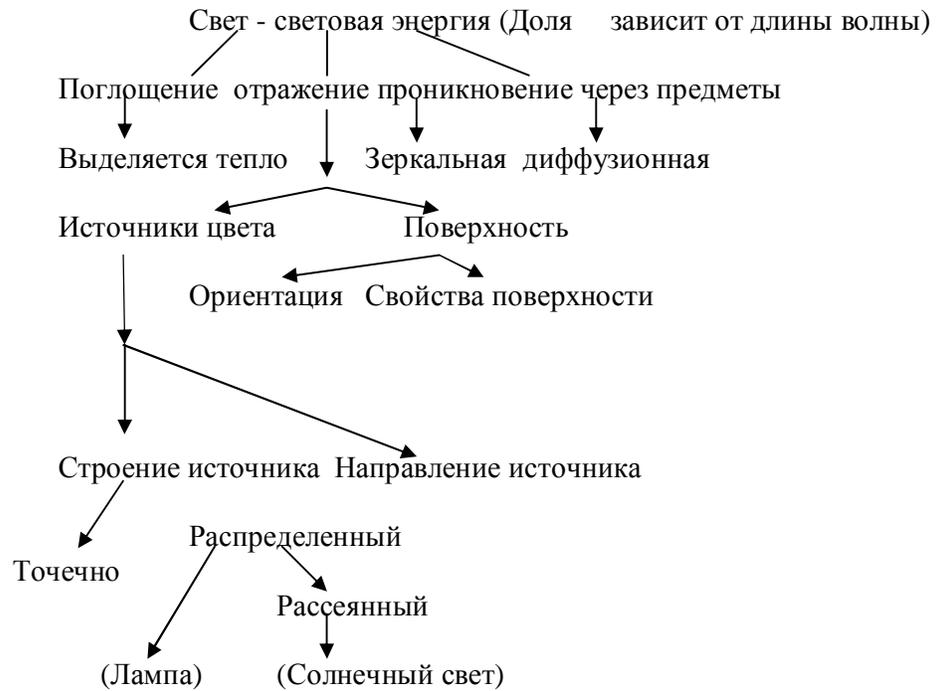


HSL - модель



Модель Максвелла





Рассеянный свет

$$I = I_a K_a + \sum_{i=1}^m \frac{I_c \cos^n B}{r + D} * K_a \cdot \cos(\Theta) + K * \cos^n(a) \text{ чем выше } n \text{ тем уже}$$

блик

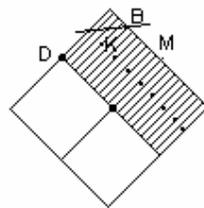
$$\cos \Theta = \frac{n * L}{|n||L|}; \cos a = \frac{RS}{|R||S|}$$

Виды закраски полигональная

- по Гуро

- по Фонго

И Гуро и Фонго живы



координаты и нормали

считается интенсивность в каждый момент времени

Берем интенсивность и интерполируем для данного полинома.

Полосы Леймана

Чтобы исключить рез. Границы дефрагментируем поверхность.

Стандартизация.

КГ

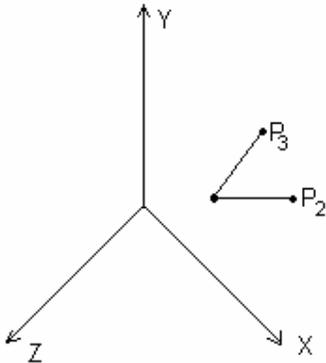
79-85	88	93	92-2000	95-2003
GKS CORE	GKS 3D PHIGS	PHIGS GL	JAVA OPENGL	JAVA 3D SVG OPEN GL

- Базисный
- Основной
- Аппаратно зависимый

Три уровня

- + Базисный графический устройства.
- + Интерф. Устройства.
- + Формат общеис. граф. данных.

Домашние задания.



- 1) Перенос точки  $P_1$  в начало координат
- 2) Поворот вокруг оси  $y$  до совмещения  $\overline{P_1P_2}$  с помощью  $yz$
- 3) Поворот вокруг оси  $x$  до совмещения  $\overline{P_1P_2}$  с отрицательной полу осью
- 4) Поворот вокруг оси  $z$  до совмещения  $\overline{P_1P_2}$  с плоскостью  $yz$

Делаем 1 шаг.

Переносим  $P_1$  в начало координат  $T(-x, -y, -z) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -x & -y & -z & 1 \end{bmatrix}$

Применение  $T$  к  $P_1P_2$  и  $P_3$  нам дает следующее

$$P_1' = P_1 \cdot T(-x_1, -y_1, -z_1) = [0001]$$

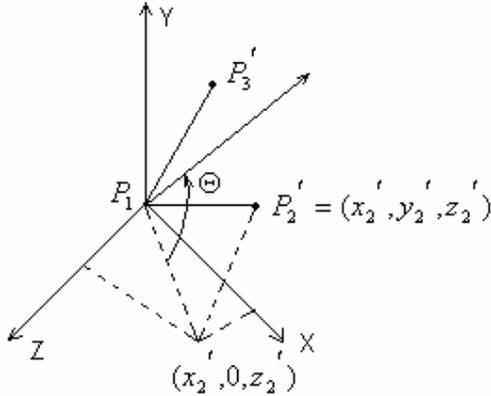
$$P_2' = P_2 \cdot T(-x_1, -y_1, -z_1) = (x_2 - x_1, y_2 - y_1, z_2 - z_1, 1)$$

$$P_3' = P_3 \cdot T(-x_1, -y_1, -z_1) = (x_3 - x_1, y_3 - y_1, z_3 - z_1, 1)$$

Делаем шаг 2.

Поворачиваем вокруг оси  $y$  на угол  $\Theta$

$$P_2' = (x_2', y_2', z_2') (x_2', 0, z_2') P_3' P_1$$



$$\text{Значит } \cos \Theta = \frac{-z_2'}{D_1} = \frac{-(z_2 - z_1)}{D_1}$$

$$\sin \Theta = \frac{x_2'}{D_1} = \frac{x_2 - x_1}{D_1}$$

$$\text{где } D_1 = \sqrt{(z_2 - z_1)^2 + (x_2 - x_1)^2}$$

$$P_2'' = P_2' \cdot R_y(\Theta) = [0, y_2, y_1, \frac{-(x_2 - x_1)^2}{D_1}, \frac{-(z_2 - z_1)^2}{D_1}, 1]$$

Делаем шаг 3

Поворачиваем вокруг оси  $x$  на отрицательный угол  $j$  :

$$\cos(-j) = \cos j = \frac{-z_2}{\|\overline{P_1 P_2}\|}$$

$$\sin(-j) = -\sin j = \frac{-y_2}{\|\overline{P_1 P_2}\|}$$

$\|\overline{P_1 P_2}\|$  - длина  $\overline{P_1 P_2}$

$$P_2''' = P_2'' \cdot R_x(j) = P_2' \cdot R_y(\Theta) \cdot R_x(j) = P_2 T R_y(\Theta) R_x(j) = [0, 0 - \|\overline{P_1 P_2}\|, 1]$$

Делаем шаг 4.

Поворачиваем вокруг оси  $z$  на угол  $a$

И так

$$P_3''' = P_3 \cdot T(x_3''', y_3''', z_3''', 1) = P_3 T(-x_1, y_1 - z_1) R_y(\Theta) \cdot R_x(j)$$

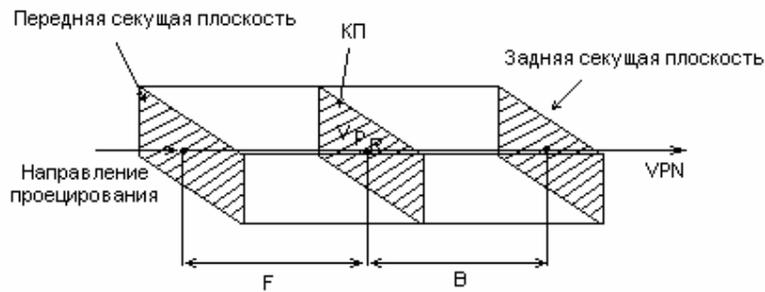
$$\cos(a) = \frac{y_3'''}{D_2}; \sin(a) = \frac{x_3'''}{D_2}$$

$$D_2 = \sqrt{(x_3''')^2 + (y_3''')^2}$$

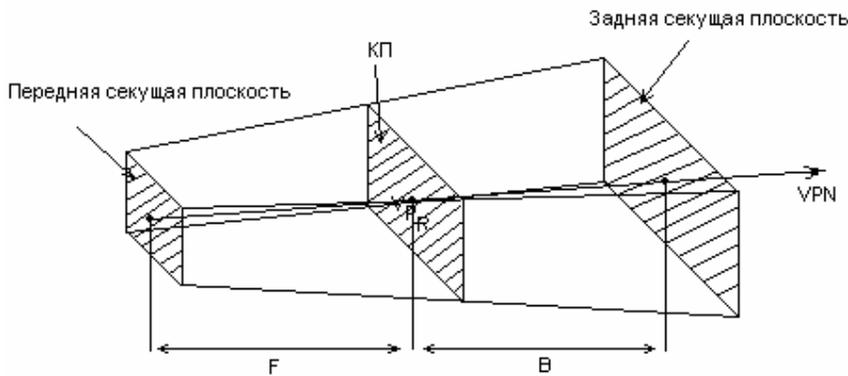
Соответственно результирующая матрица:

$$T(-x, -y, -z) \cdot R_y(\Theta) \cdot R_x(j) \cdot R_z(a) = T \cdot R$$

Усеченный видимый объем для случая ортогональной параллельной проекции.



Усеченный видимый объем



Кубическая поверхность

$$x(s,t) = a_{11}s^3t^3 + a_{12}s^3t^2 + a_{13}s^3t + a_{14}s^3 + a_{21}s^2t^2 + a_{22}s^2t + a_{23}s^2 + a_{24}s^2 + a_{31}st^3 + a_{32}st^2 + a_{33}s^2t + a_{34}s + a_{41}t^3 + a_{42}t^2 + a_{43}t + a_{44}$$

$$0 \leq s \leq 1 \quad 0 \leq t \leq 1$$

$$x(s,t) = SC_x T^t;$$

$$S = [S^3 \ S^2 \ S \ 1] \ , \ T = [t^3 \ t^2 \ t \ 1]$$

$C_x$  - коэффициенты бикубического многочлена.

$$C_y C_z \text{ -----} || \text{-----} y(s,t), z(s,t)$$

Формы Эрмита

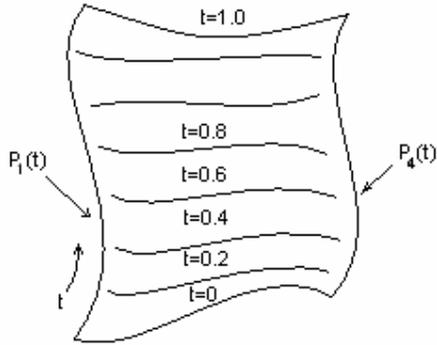
$$x(s) = SM_a G_{hx}$$

$$M_a = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$G_{hx} = \begin{bmatrix} P_1 \\ P_4 \\ R_1 \\ R_4 \end{bmatrix}$$

$$x(s,t) = SM_a G_{hx}(t) = SM_a$$

$$\begin{bmatrix} P_1(t) \\ P_4(t) \\ R_1(t) \\ R_4(t) \end{bmatrix}$$



$$P_{1x}(t) = TM_a \begin{bmatrix} q_{11} \\ q_{12} \\ q_{13} \\ q_{14} \end{bmatrix}_X \quad P_{4x}(t) = TM_a \begin{bmatrix} q_{21} \\ q_{22} \\ q_{23} \\ q_{24} \end{bmatrix}_X \quad R_{1x}(t) = TM_a \begin{bmatrix} q_{31} \\ q_{32} \\ q_{33} \\ q_{34} \end{bmatrix}_X \quad R_{4x}(t) = TM_a \begin{bmatrix} q_{41} \\ q_{42} \\ q_{43} \\ q_{44} \end{bmatrix}_X$$

$$(ABC)^T = C^T B^T A^T$$

$$\begin{bmatrix} P_{1x}(t) \\ P_4(t) \\ R_1(t) \\ R_4(t) \end{bmatrix} = \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ q_{31} & q_{32} & q_{33} & q_{34} \\ q_{41} & q_{42} & q_{43} & q_{44} \end{bmatrix} M_R^T T^T = Q_x M_R^T T^T$$

$$x(s,t) = SM_h Q_x M_h^T T^T$$

$$y(s,t) = SM_h Q_y M_h^T T^T$$

$$z(s,t) = SM_h Q_z M_h^T T^T$$

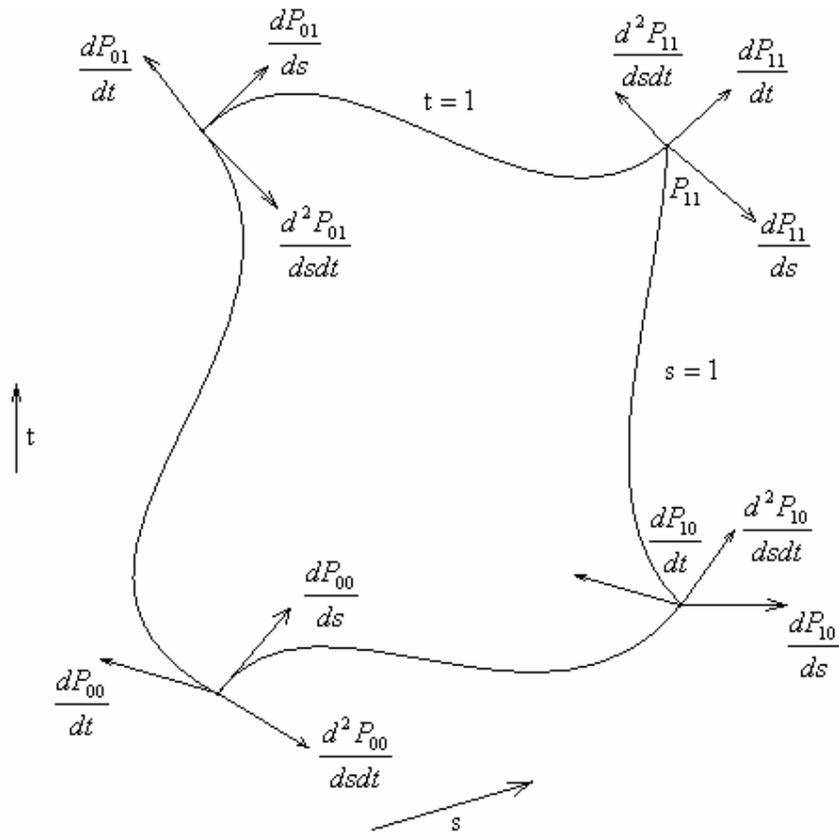
$$q_{11} = x(0,0)$$

$$q_{12} = x(0,1)$$

$$q_{13} = \frac{dx}{dt}(0,0)$$

$$q_{33} = \frac{d^2x}{dsdt}(0,0)$$

$$Q_x = \begin{bmatrix} x_{00} & x_{01} & \mathbf{M} \left( \frac{dx}{dt} \right)_{00} & \left( \frac{dx}{dt} \right)_{01} \\ \mathbf{L} & \mathbf{L} & \mathbf{M} \left( \frac{dx}{dt} \right)_{10} & \left( \frac{dx}{dt} \right)_{11} \\ \left( \frac{dx}{ds} \right)_{00} & \left( \frac{dx}{ds} \right)_{01} & \mathbf{M} \frac{d^2x}{dsdt} \Big|_{00} & \left( \frac{d^2x}{dsdt} \right)_{01} \\ \left( \frac{dx}{ds} \right)_{10} & \left( \frac{dx}{ds} \right)_{11} & \mathbf{M} \frac{d^2x}{dsdt} \Big|_{10} & \left( \frac{d^2x}{dsdt} \right)_{11} \end{bmatrix}$$



Форма Безье

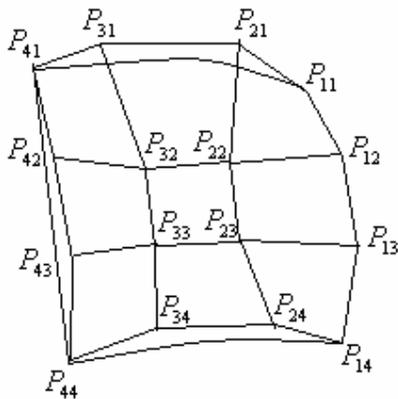
$$x(s, t) = SM_b P_x M_b^T T^T$$

$$x(s, t) = SM_b P_x M_b^T T^T$$

$$z(s, t) = SM_b P_z M_b^T T^T$$

$$M_b = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Она обладает свойством выпуклой оболочки.



Форма B - сплайнов

$$x(s,t) = SM_s P_x M_s^T T^T$$

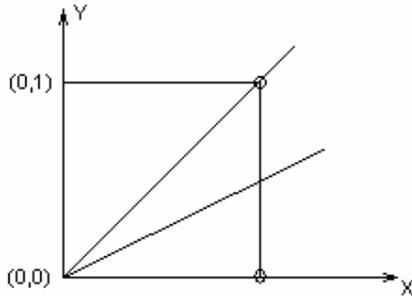
$$y(s,t) = SM_s P_y M_s^T T^T$$

$$z(s,t) = SM_s P_z M_s^T T^T$$

$C^{(2)}$  - непрерывность

$$M_s = \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & -3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}$$

Алгоритм Брезенхейна



$$\frac{1}{a} \leq \frac{\Delta y}{\Delta x} \leq 1 \text{ (ошибка } \geq 0)$$

$$0 \leq \frac{\Delta y}{\Delta x} < \frac{1}{2} \text{ (ошибка } < 0)$$

Инициализировать ошибку в  $-\frac{1}{2}$  ошибка равна ошибка плюс  $\frac{\Delta y}{\Delta x}$

Алгоритм Брезенхейна для первого октанта т.е.  $0 < \Delta y \leq \Delta x$

Дано: концы отрезка  $(x_1, y_1)$  и  $(x_2, y_2)$  - не совпадают

*Integer* - функция преобразования в целое  $x_1, y_1$

$\Delta x, \Delta y$  - целые

$l$  - вещественное.

Инициализация переменных.

$$x = x_1$$

$$y = y_1$$

$$\Delta x = x_2 - x_1$$

$$\Delta y = y_2 - y_1$$

Инициализация (с поправкой)  $\frac{1}{2}$  пикселя.

$$C = \frac{\Delta y}{\Delta x}$$

Начало основного цикла.

*for*  $i = 1$  *to*  $\Delta x$

*Plot*( $x, y$ )

*While*( $l \geq 0$ )

*end while*

$x = x + 1$

$$e = l + \frac{\Delta y}{\Delta x}$$

*next i*  
*finish*

### Обобщенный целочисленный алгоритм

Дано Концы отрезка  $(x_1, y_1)$  и  $(x_2, y_2)$  не совпадают, все переменные целые

$$\text{sign} = \begin{cases} -1 & \text{отрицательный аргумент} \\ 0 & \text{нулевой аргумент} \\ 1 & \text{положительный аргумент} \end{cases}$$

$x = x_1$   
 $y = y_1$   
 $\Delta x = \text{abs}(x_2 - x_1)$   
 $\Delta y = \text{abs}(y_2 - y_1)$   
 $s_1 = \text{sign}(x_2 - x_1)$   
 $s_2 = \text{sign}(y_2 - y_1)$

Инициализация переменных

*if*  $\Delta y > \Delta x$  *then*

Врем =  $\Delta x$

$\Delta x = \Delta y$

$\Delta y = \text{Врем}$

Обмен = 1

*else* обмен = 1

*else* обмен = 0

*end if*

Инициализация  $\bar{e}$  ;

$\bar{e} = z \cdot \Delta y - \Delta x$

Основной цикл

*for*  $i = 1$  *to*  $\Delta x$

$\text{plot}(x, y)$

*while*  $(\bar{e} \geq 0)$

*if* обмен = 1 *then*

$x = x + s_1$

*else*

$y = y + s_2$

*end if*

$\bar{e} = \bar{e} - 2 \cdot \Delta x$

*end while*

*if* обмен = 1 *then*

$y = y \cdot s_2$

*else*  $x = x + s_1$

*end if*

$\bar{e} = \bar{e} + 2 \cdot \Delta y$

*next i*

*finish*