Компьютерная графика

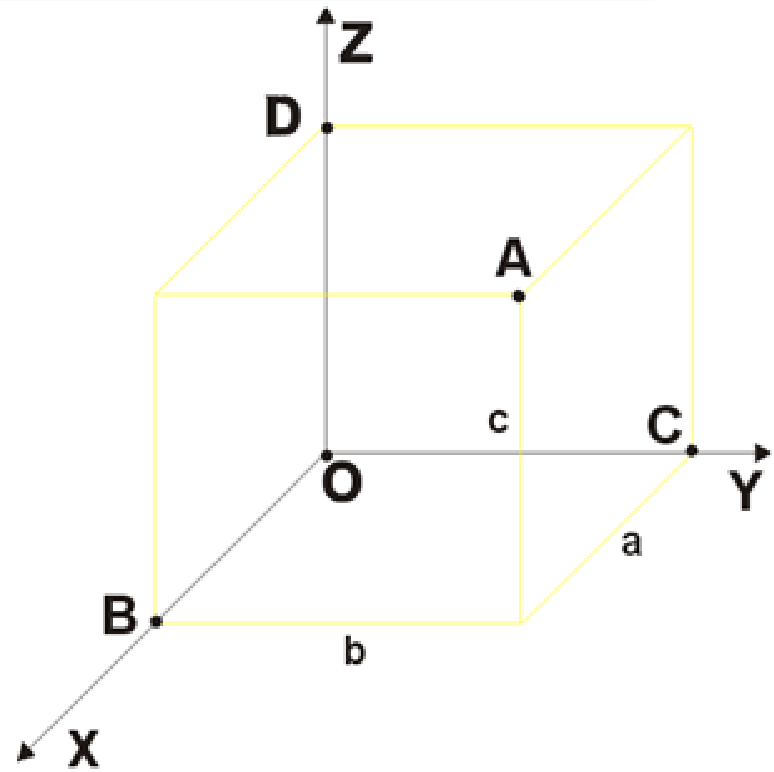
Курс лекций

Тема №3. Математические основы компьютерной графики.

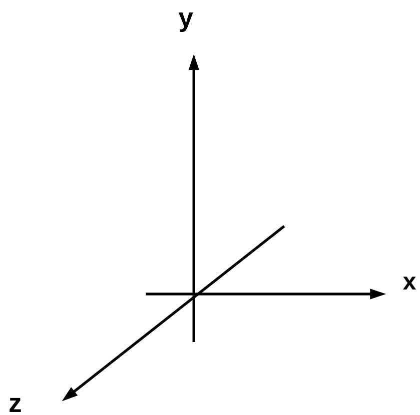
Прямоугольная декартова система координат

Прямоугольная система координат в пространстве образуется тремя взаимно перпендикулярными осями координат OX — ось абсцисс, OY — ось ординат, OZ — ось аппликат.

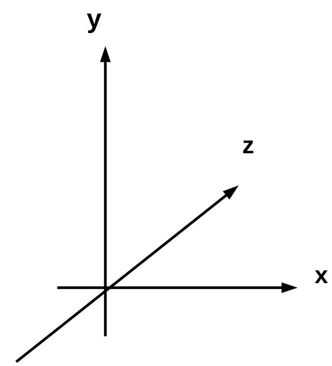
Прямоугольная система координат описывается набором ортов, сонаправленных с осями координат. В трёхмерном случае такие орты обычно обозначаются **i j k** или e_x e_y e_z .



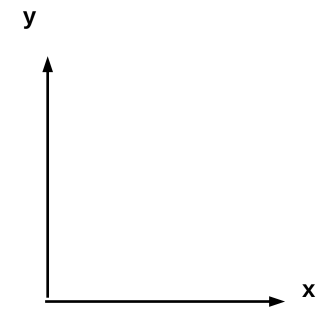
Прямоугольные системы координат



а) правосторонняя система



б) левосторонняя система



в) оконная система

Координатный фрейм в трехмерном пространстве: точка начала отсчета и тройка линейно-независимых (как правило, взаимно перпендикулярных) единичных векторов

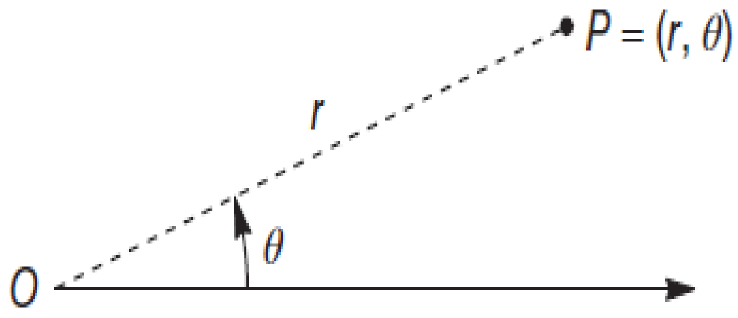
Полярная система координат

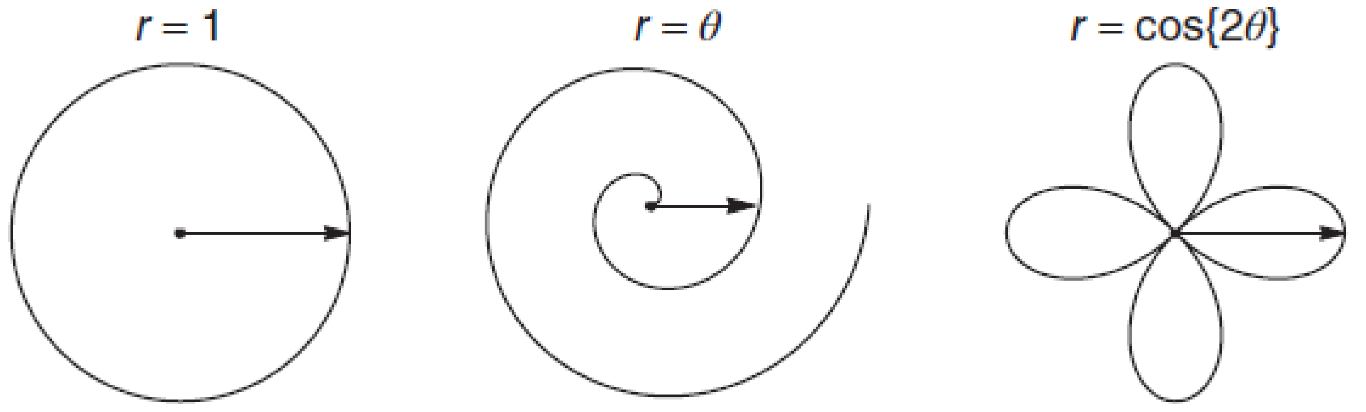
$$x = r\cos\theta$$

$$y = r\sin\theta$$

$$r = \sqrt{x^2 + y^2}$$

$$\theta = \tan^{-1} y/x$$





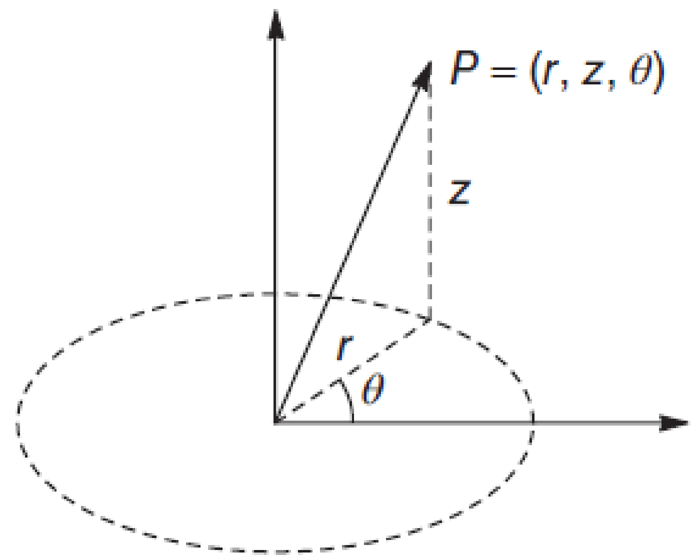
Цилиндрическая система координат

$$x = r\cos\theta$$

$$y = r\sin\theta$$

$$r = \sqrt{x^2 + y^2}$$

$$\theta = \tan^{-1} y/x$$



Сферическая система координат

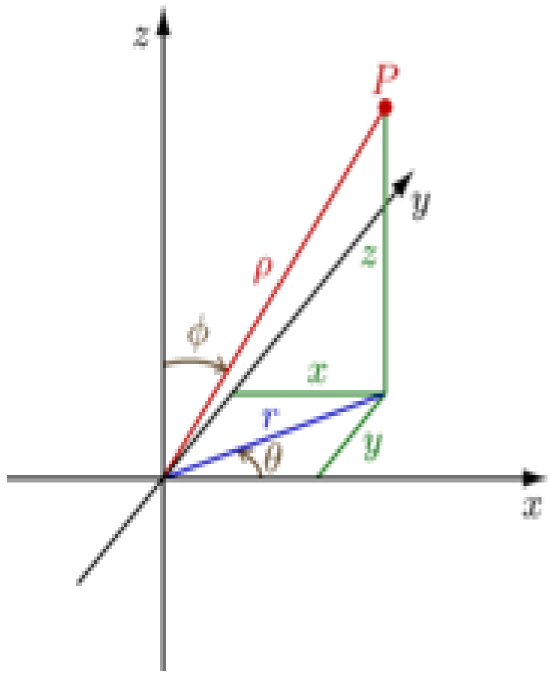
Сферические координаты - тройка $(\rho,\ \varphi,\ \theta)$, где
ρ — расстояние до зафиксированной точки,
θ и φ азимутальный и зенитный угол соответсвенно.

Закон преобразования координат от сферических к декартовым:

$$x = \rho \sin\varphi \cos\theta,$$
$$y = \rho \sin\varphi \sin\theta,$$
$$z = \rho \cos\varphi.$$

Закон преобразования декартовых координат к сферическим:

$$\rho = \sqrt{x^2 + y^2 + z^2},$$
$$\phi = \arccos\left(\frac{z}{\sqrt{x^2 + y^2 + z^2}}\right),$$
$$\theta = \operatorname{arctg}\left(\frac{y}{x}\right).$$



Барицентрические координаты

- **Барицентри́ческие координа́ты** — скалярные параметры, набор которых однозначно задаёт точку аффинного пространства
- точечный базис в n-мерном аффинном пространстве представляет собой систему из (n+1)-й точки, которые предполагаются аффинно независимыми

$$P = \sum_{i=1}^{n} \alpha_i P_i$$

$$\sum_{i=1}^{n} \alpha_i = 1$$

Векторное пространство

- определено понятие вектора;
- определены операции:
  - сложения векторов;
  - умножения вектора на скаляр;
- операции удовлетворяют аксиомам веторного (линейного) пространства:
  - коммутативность сложения; $\vec{v} + \vec{w} = \vec{w} + \vec{v}$
  - ассоциативность сложения; $(\vec{u} + \vec{v}) + \vec{w} = \vec{u} + (\vec{v} + \vec{w})$
  - существование нейтрального элемента относительно сложения; $\vec{v} + \vec{0} = \vec{v}$
  - существование противоположного элемента относительно сложения; $\vec{v} + (-\vec{v}) = \vec{0}$
  - ассоциативность умножения на скаляр; $(\alpha\beta)\vec{v} = \alpha\,(\beta\vec{v})$
  - унитарность: умножение на нейтральный (по умножению) элемент сохраняет вектор; $1 \cdot \vec{v} = \vec{v}$
  - дистрибутивность умножения на вектор относительно сложения скаляров; $\alpha(\vec{v} + \vec{w}) = \alpha\vec{v} + \alpha\vec{w}$
  - дистрибутивность умножения на скаляр относительно сложения векторов. $(\alpha + \beta)\vec{v} = \alpha\vec{v} + \beta\vec{v}$

Линейная комбинация векторов

- **Линейной комбинацией** m векторов $v_1, v_2, \dots, v_m$ называется вектор вида $\mathbf{W} = a_1\mathbf{V}_1 + a_2\mathbf{V}_2 + \dots + a_n\mathbf{V}_m$,, где $a_1, a_2, \dots, a_m$ — скаляры.
- Линейная комбинация называется **нетривиальной**, если хотя бы один из её коэффициентов отличен от нуля.
- Линейная комбинация векторов называется **<u>аффинной (барицентрической) комбинацией</u>** (affine combination), если сумма коэффициентов $a_1, a_2, \dots, a_m$ равна единице.
- Линейная комбинация называется **сбалансированной**, сумма коэффициентов $a_1, a_2, \dots, a_m$ равна нулю.
- Аффинная комбинация векторов называется **<u>выпуклой</u>**, если все коэффициенты данной комбинации неотрицательны.

Аффинное пространство

- определено понятие точки аффинного пространства и вектора соответствующего векторного пространства
- определена операция *сложения* точки и вектора, удовлетворяющая следующим аксиомам:
    - $(P + \mathbf{v}) + \mathbf{w} = P + (\mathbf{v} + \mathbf{w})$
    - $P + \mathbf{o} = P$
    - для любых двух точек P,Q существует единственный вектор **v** такой, что $Q = P + \mathbf{v}$
- для аффинного пространства все точки являются равноправными (в частности, в нём не определено понятие *нулевой точки*, или *начала отсчёта*)

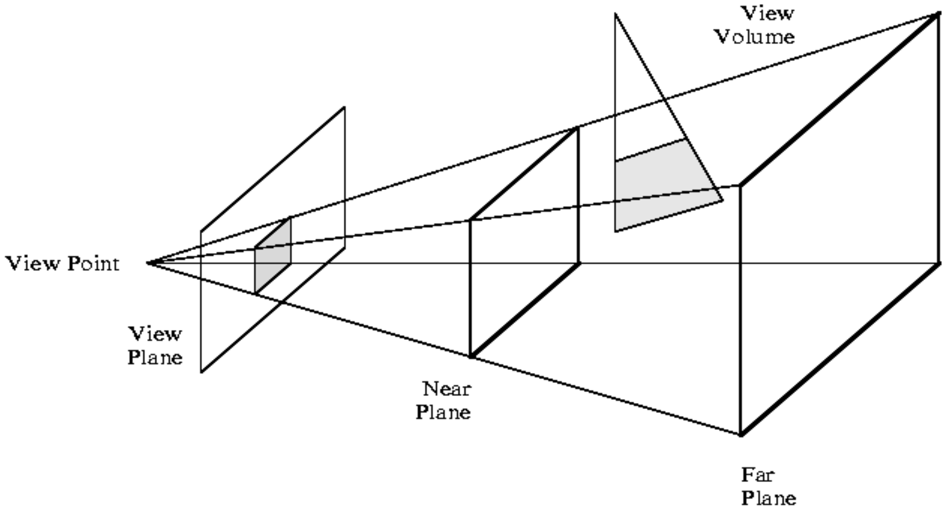
Линейная комбинация точек

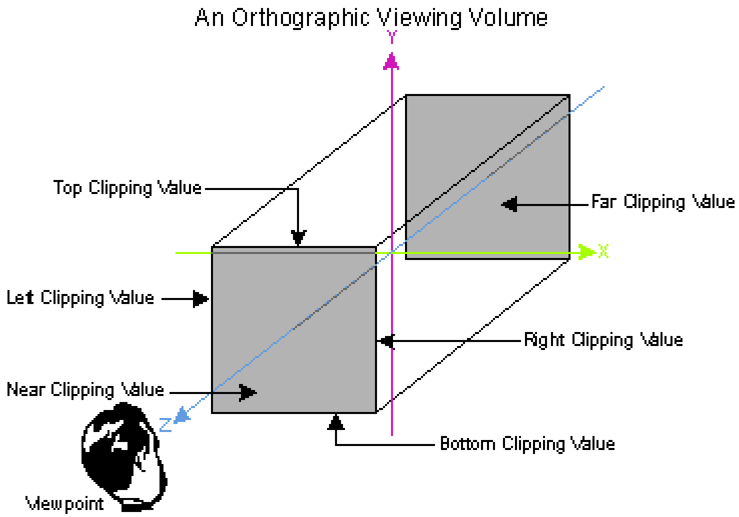
- линейная комбинация точек аффинного пространства имеет смысл только в двух случаях:
  – барицентрическая (аффинная) комбинация: сумма коэффициентов равна единице (результат – точка);
  – сбалансированная комбинация: сумма коэффициентов равна нулю (результат - вектор);
- все коэффициенты аффинной комбинации неотрицательны, тогда и только тогда, когда точка лежит внутри выпуклой оболочки, построенной на точках, образующих аффинную комбинацию (на границе выпуклой оболочки всего два коэффициента ненулевые);

$$P = \sum_{i=1}^{n} \alpha_i P_i$$

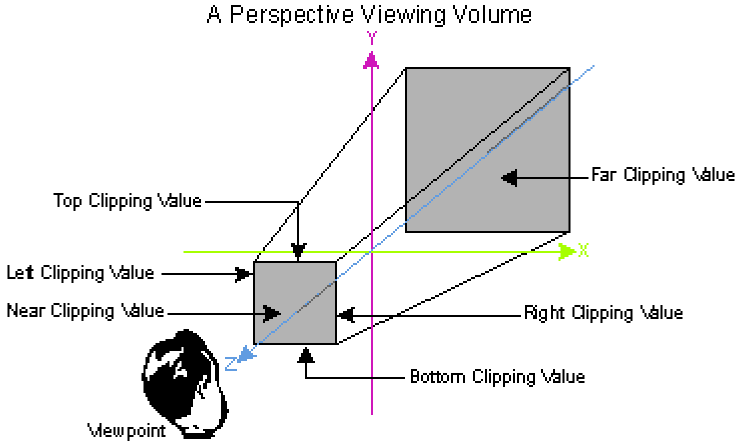
$$\sum_{i=1}^{n} \alpha_i = 1$$

Формирование сцены и изображения

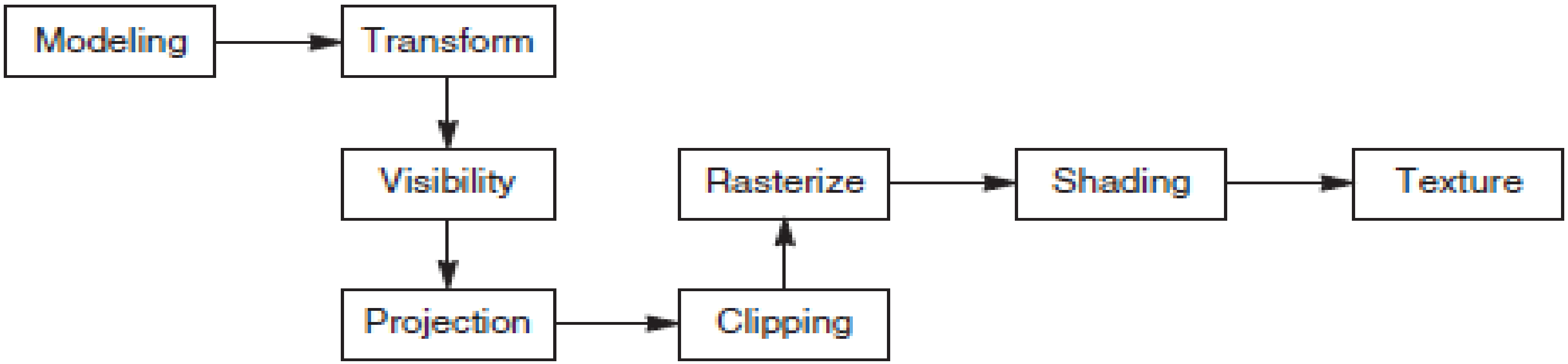




- положение и параметры объектов
- положение и атрибуты наблюдателя (камеры)
- положение и параметры источников света
- объем видимости, параметры проецирования и картинная плоскость

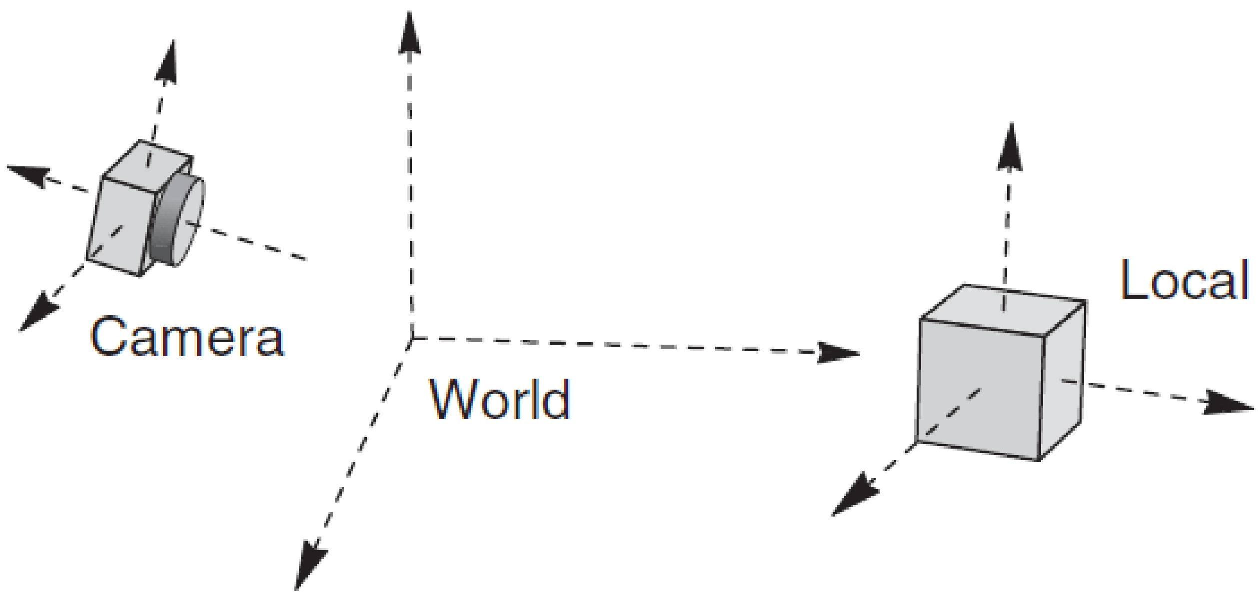


Графический конвейер



- отображение (rendering):
  - преобразования (transformation) – задание местоположения;
  - определение видимости (visibility) – область видимости (field of view) + нелицевые поверхности → отсечение (clipping);
  - проекция на картинную плоскость (projection);
  - растеризация (rasterization);
  - закраска (shading);
  - текстурирование (texturing).

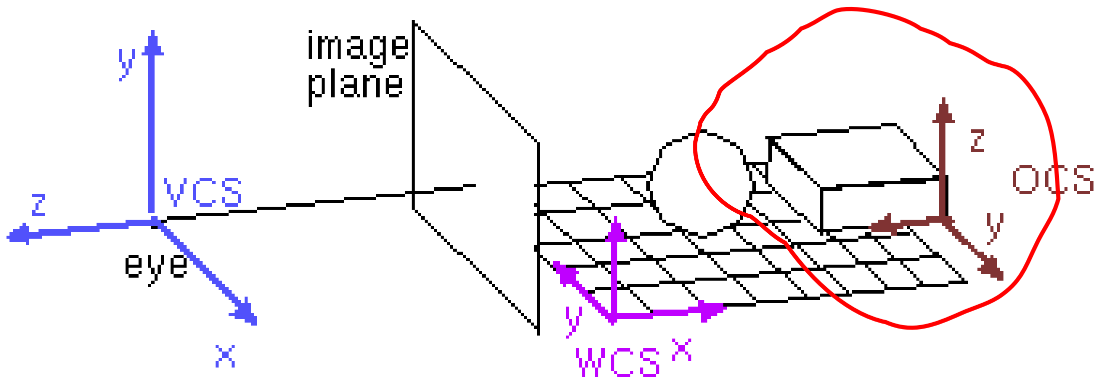
Преобразования координат: этап 1



- модельно-видовые преобразования (преобразование координат моделей отдельных объектов, заданных в локальных системах координат, к некоторой единой (мировой) системе координат и учет положения наблюдателя) → мировые координаты

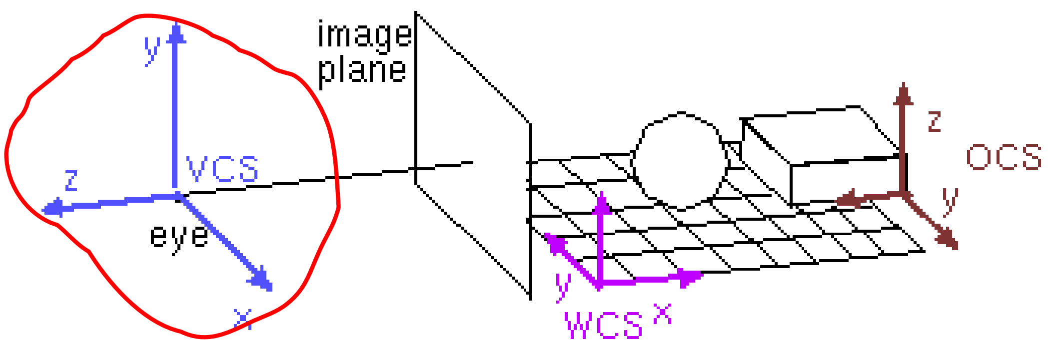
Модельное преобразование

- переводит модель, заданную в локальных (собственных) координатах, в глобальное (мировое пространство);
- сцена «собирается» из частей, с помощью модельных преобразований (обычно композиция переносов, поворотов, масштабирований);
- на выходе – модель в единых *мировых координатах.*

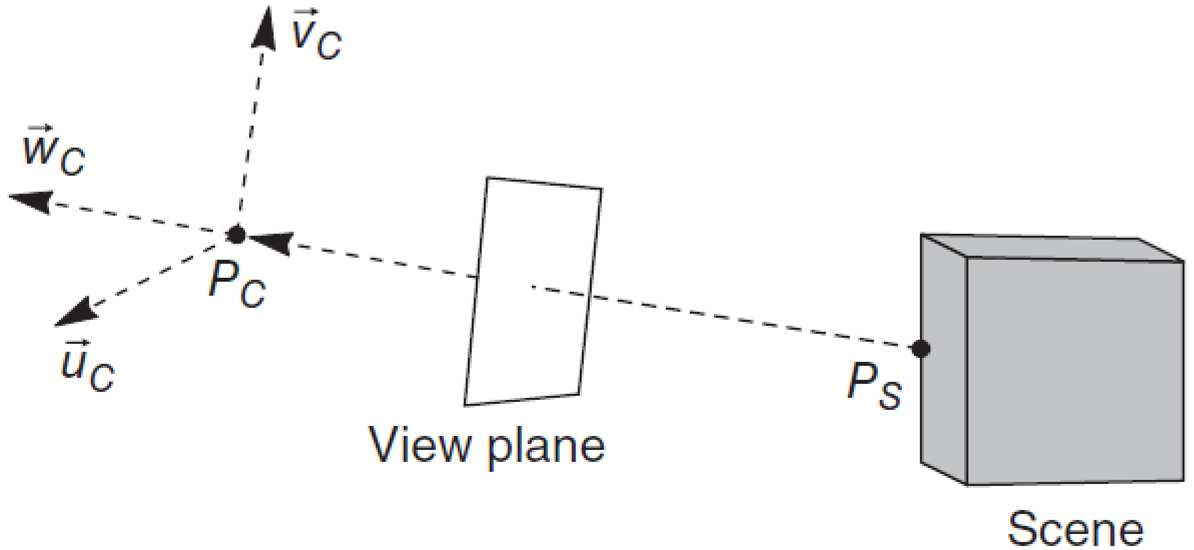


Виртуальная камера

- определяет положение наблюдателя в пространстве
- параметры:
  – положение (точка) наблюдения;
  – направление взгляда;
  – направление «вверх»;
  – параметры проекции;
- положение, направление взгляда и направление «вверх» задают видовое преобразование.



Система координат камеры



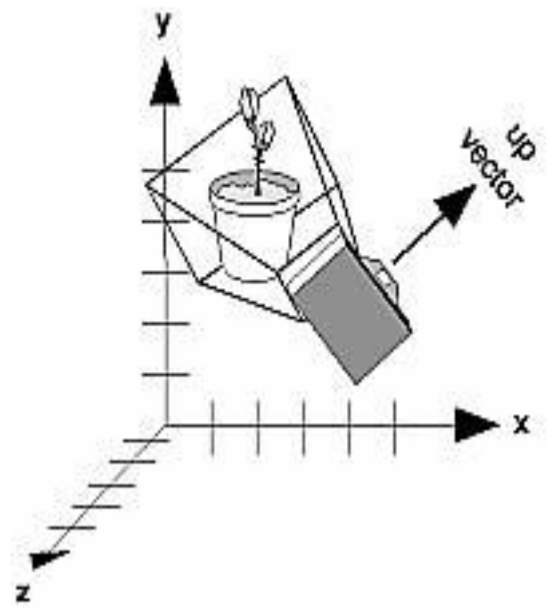
$$\vec{w}_c = \frac{\vec{n}}{|\vec{n}|}$$

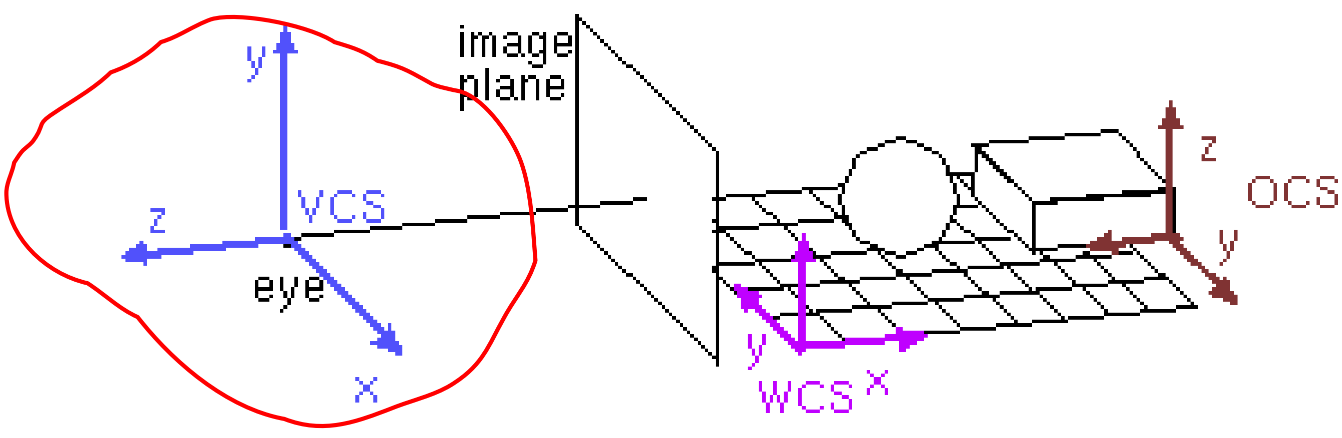
$$\vec{u}_c = \frac{\vec{v} \times \vec{w}_c}{|\vec{v} \times \vec{w}_c|}$$

$$\vec{v}_c = \vec{w}_c \times \vec{u}_c$$

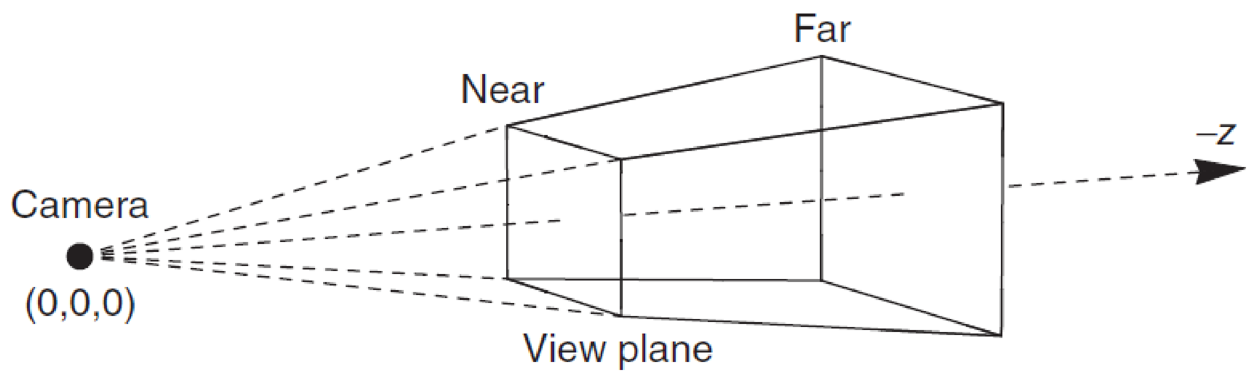
Видовое преобразование

- «подгоняет» мир под стандартную камеру, преобразует мировую систему координат в видовые координаты (которые подходят для «стандартной» камеры);
- на выходе – модель, готовая к проекции на экран.



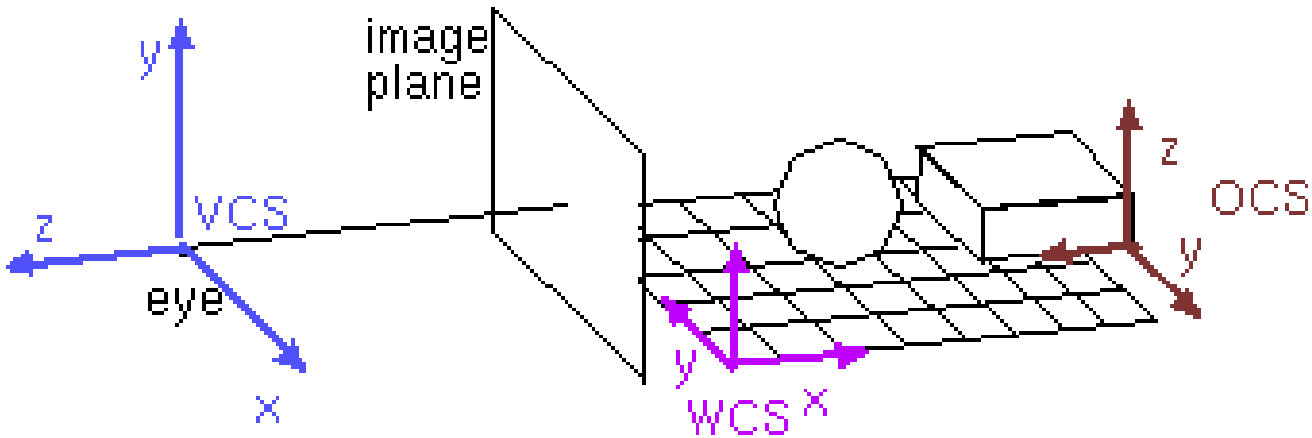


Преобразования координат: этап 2



- преобразования проецирования (в т.ч. отсечение по объему видимости) → усеченные координаты
- преобразование нормализации (приведение координат к некоторому предопределенному интервалу, например, [-1.0; 1.0]) → нормализованные координаты
- преобразование к порту просмотра (преобразование рабочей станции - из аппаратно-независимой формы в координаты устройства) → оконные координаты

Преобразования координат (в графическом конвейере)



- локальные координаты → мировые координаты → усеченные координаты → нормализованные координаты → оконные координаты
- задачи:
  – определить набор базовых преобразований, достаточный для задания всех необходимых преобразований графического конвейера;
  – определить наиболее удобный способ их представления.

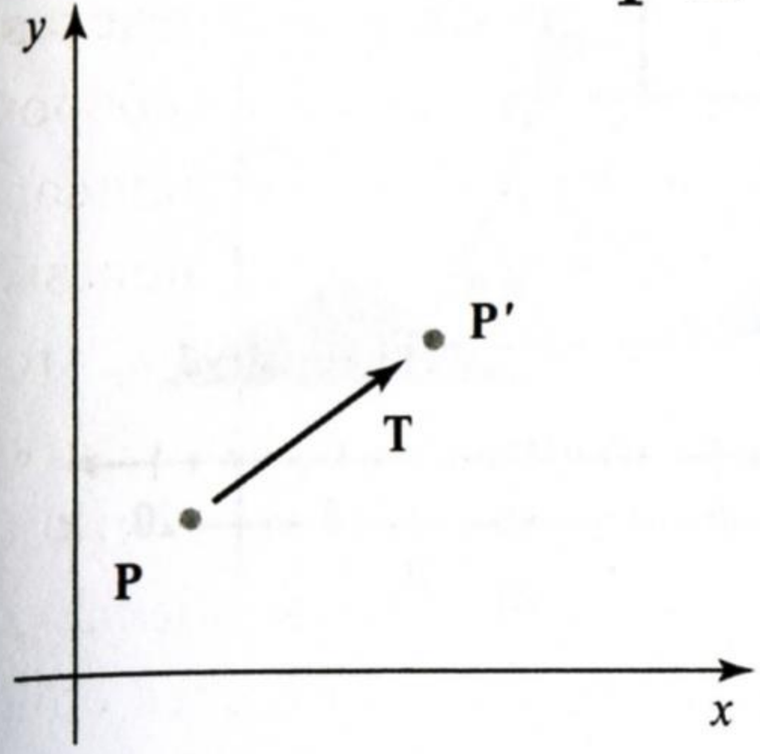
Базовые двумерные преобразования

- двумерный перенос;
- двумерный поворот;
- двумерное масштабирование;
- двумерный сдвиг.

Двумерный перенос

$$x' = x + t_x, \qquad y' = y + t_y.$$

$$\mathbf{P} = \begin{bmatrix} x \\ y \end{bmatrix}, \qquad \mathbf{P}' = \begin{bmatrix} x' \\ y' \end{bmatrix}, \qquad \mathbf{T} = \begin{bmatrix} t_x \\ t_y \end{bmatrix}.$$



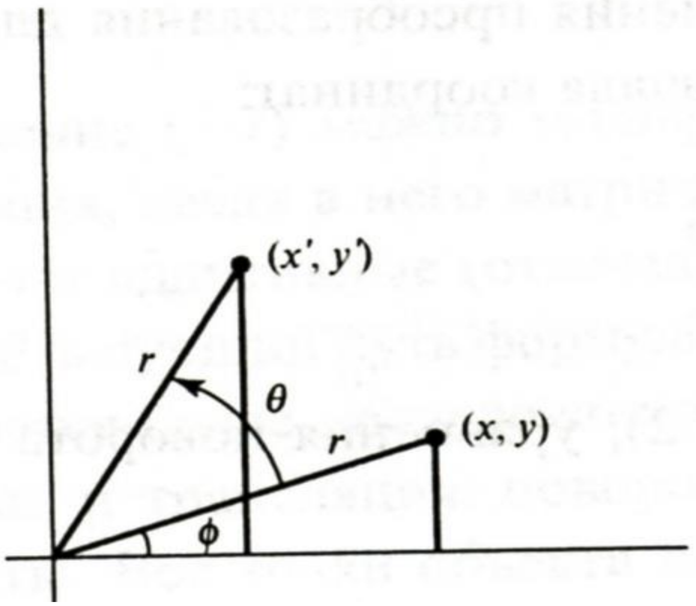
$$\mathbf{P}' = \mathbf{P} + \mathbf{T}.$$

Двумерный поворот

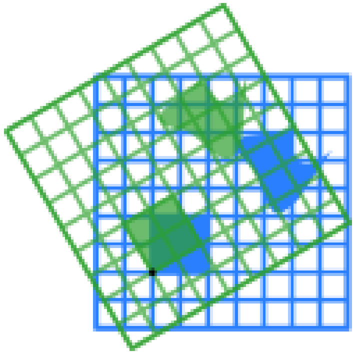
$$x' = x\cos\theta - y\sin\theta;$$
$$y' = x\sin\theta + y\cos\theta.$$

$$\mathbf{P}' = \mathbf{R}\cdot\mathbf{P},$$

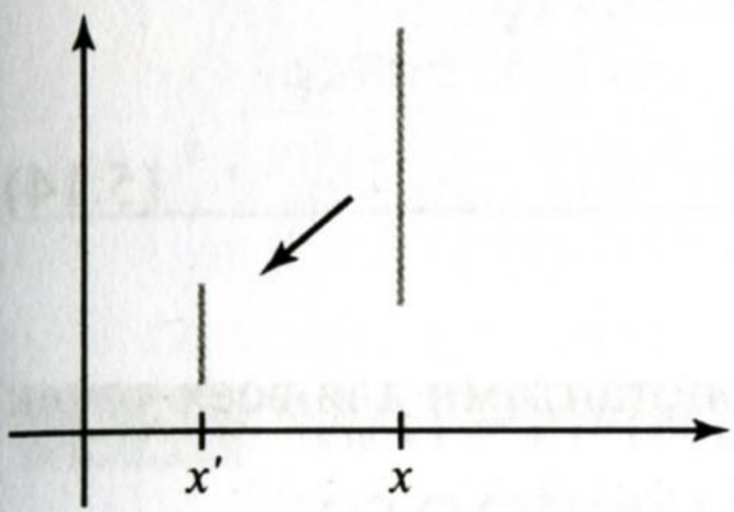
$$\mathbf{R} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}.$$



$$x' = x_r + (x - x_r)\cos\theta - (y - y_r)\sin\theta;$$
$$y' = y_r + (x - x_r)\sin\theta + (y - y_r)\cos\theta.$$



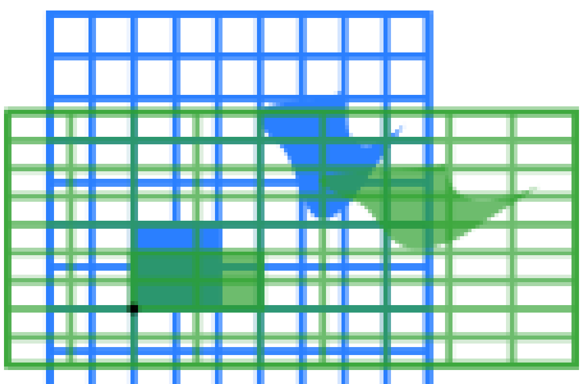
Двумерное масштабирование

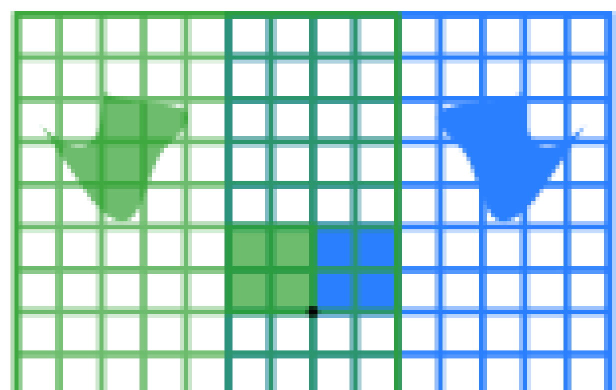


$$x' = x \cdot s_x, \qquad y' = y \cdot s_y.$$

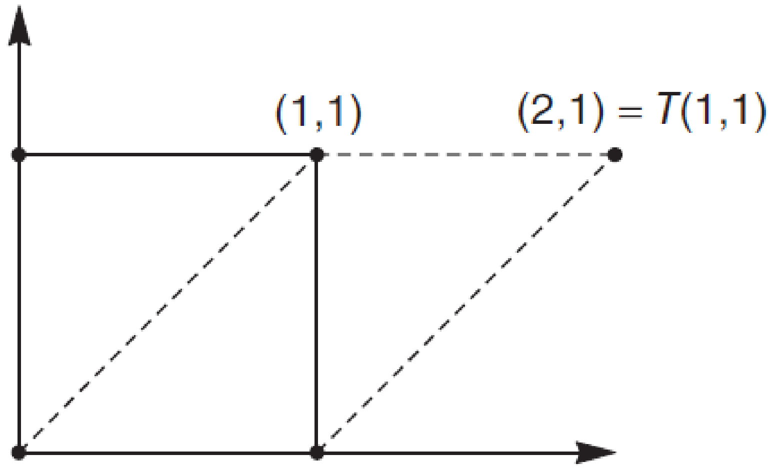
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\mathbf{P}' = \mathbf{S} \cdot \mathbf{P},$$



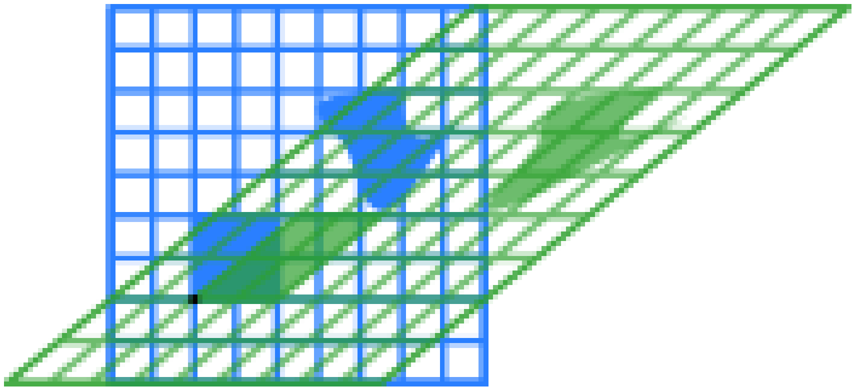


Двумерный сдвиг

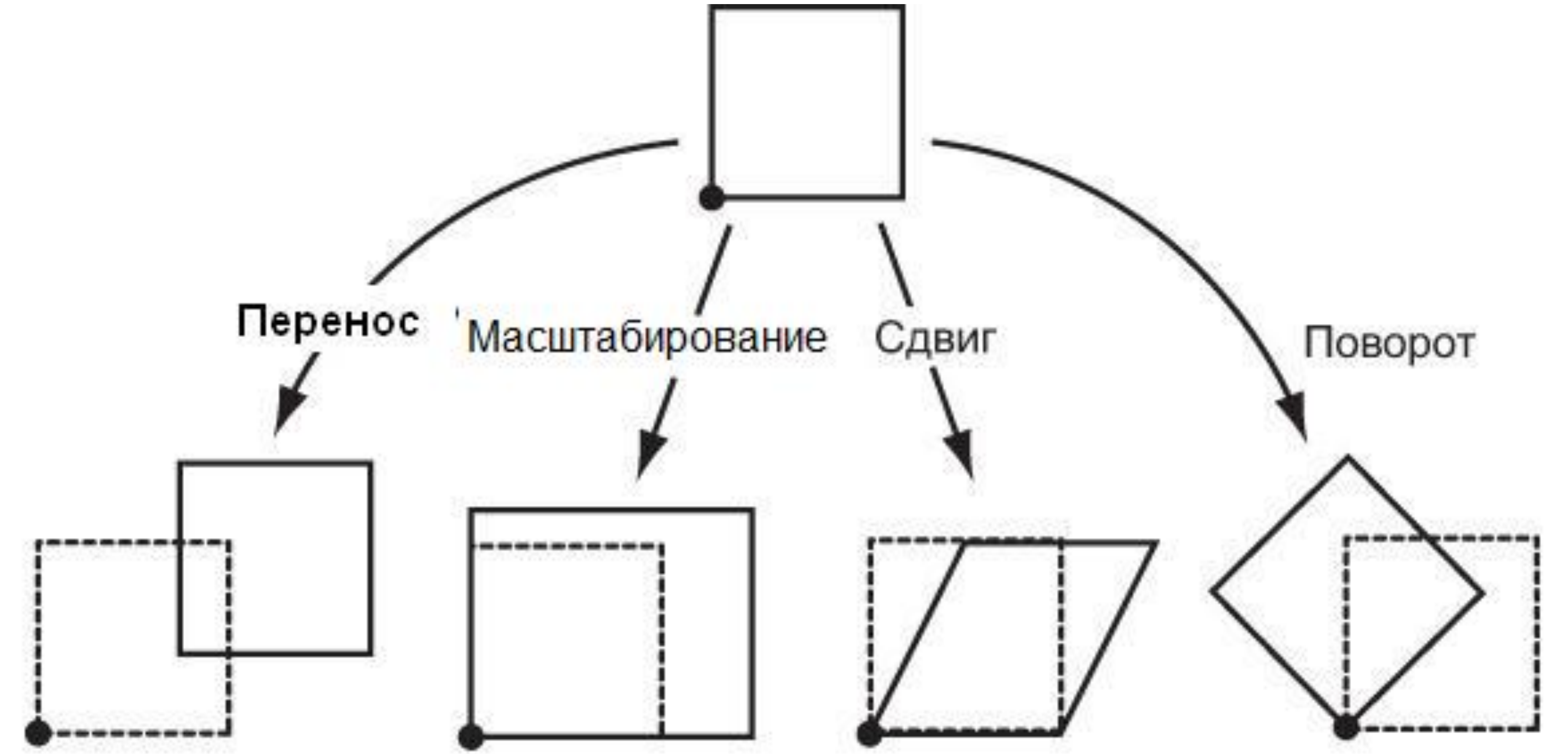


$$M = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & s_x \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$



Двумерные преобразования



$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_1 \\ t_2 \end{pmatrix}$$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

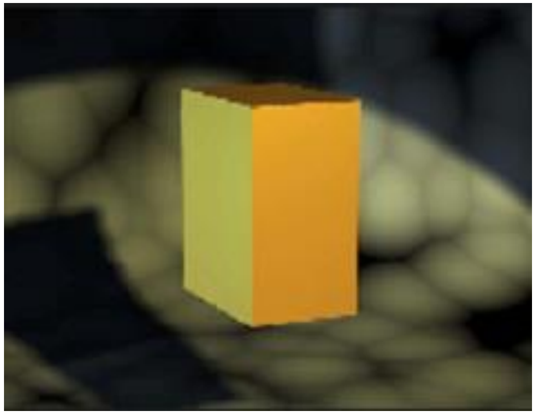
$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & s_x \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

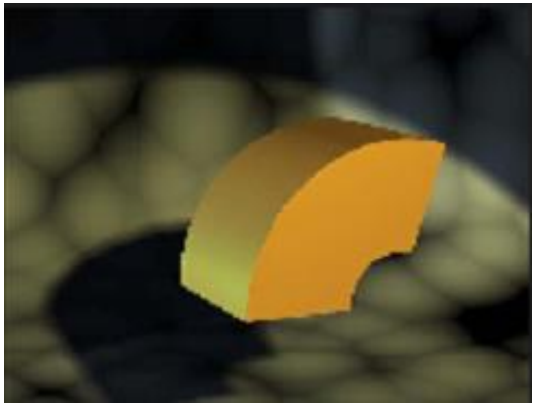
$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

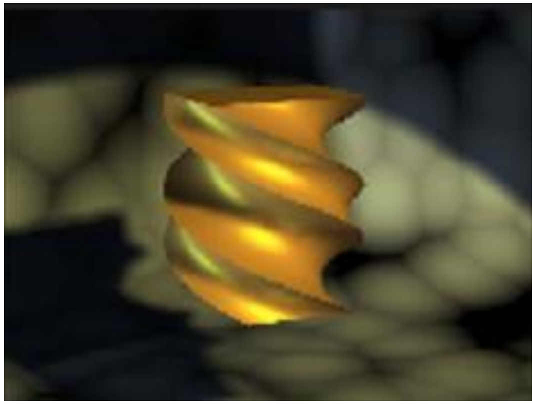
Нелинейные преобразования

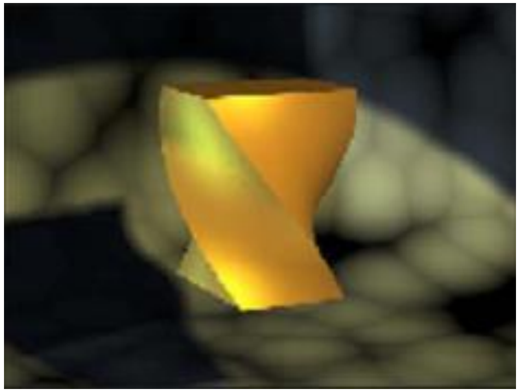
- Произвольное преобразование точек модели

  M'=T(M)









Линейные преобразования

- линейным преобразованием (линейным оператором, линейным отображением) векторного пространства является преобразование, удовлетворяющее свойству линейности:

$$T(\vec{v} + \vec{w}) = T(\vec{v}) + T(\vec{w})$$

$$T(a\vec{v}) = aT(\vec{v})$$

- линейное преобразование (линейный оператор) в некотором базисе задаётся с помощью матрицы, столбцы которой представляют собой преобразованные векторы данного базиса;
- (*) в данном случае рассматривается эндоморфизм.

Однородные координаты

Определение. *Однородными координатами точки $P=(x_1,..,x_n), P \in R^n$ называются координаты $P_{hom}=(wx_1,wx_2,..,wx_n,w), P_{hom} \in R^{n+1}$, причем хотя бы один элемент должен быть отличен от нуля.*

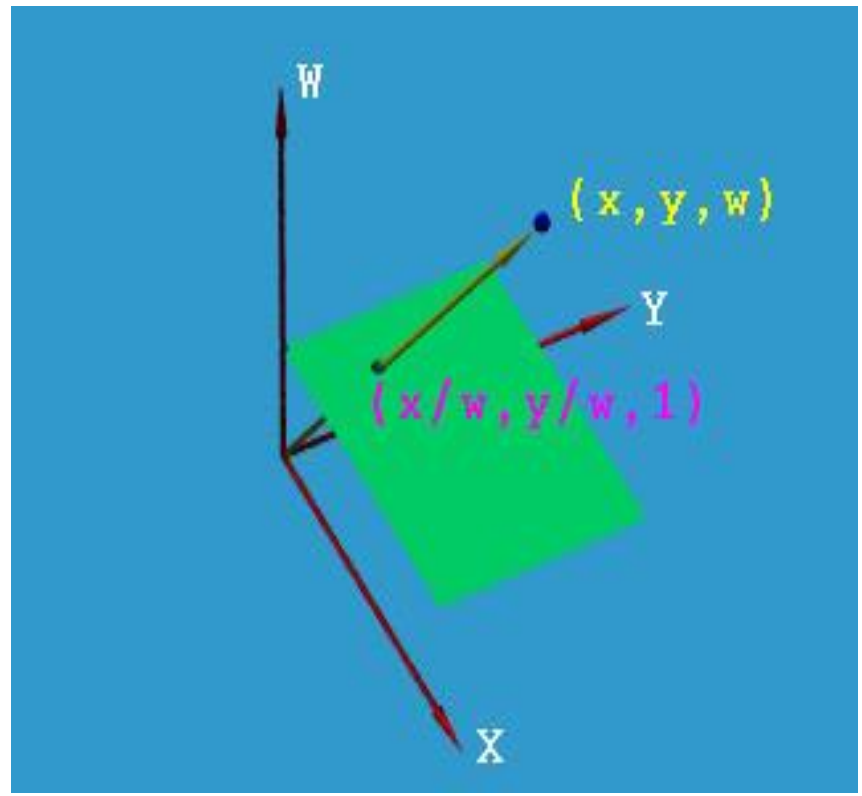
Представление вектора в однородных координатах (идеальная точка):

$$v = (v_1, v_2, v_3, 0)$$

Представление точки в однородных координатах:

$$P=(p_1,p_2,p_3,1)$$

→проективное пространство



Матричное представление геометрических преобразований

- Координаты представляются вектором-столбцом
- Геометрическое преобразование задается матрицей, умножаемой справа на вектор-столбец координат

  P' = M*P

- Матрица композиции преобразований является произведением матриц элементарных преобразований (матрицы перемножаются в обратном порядке): операция является ассоциативной, но в общем случае некоммутативной

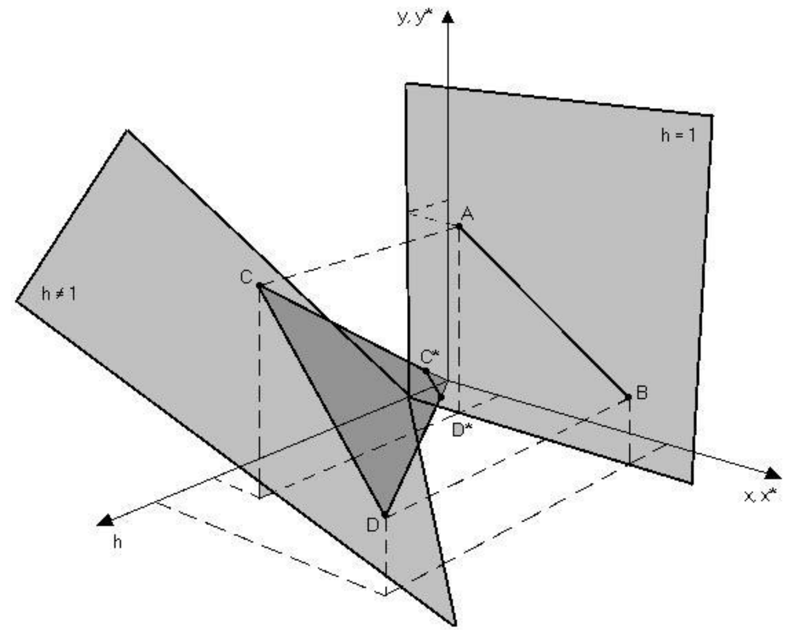
  $$M = M_2 * M_1$$

- Обратное преобразование задается обратной матрицей

$$\begin{pmatrix} x' \\ y' \\ z' \\ w' \end{pmatrix} = \begin{pmatrix} A & B & C & D \\ E & F & G & H \\ I & J & K & L \\ m & n & p & q \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Типичные линейные преобразования

- общие линейные преобразования
  - w’ != 1 (проективные, необходимо «перспективное деление»)
  - прямые переходят в прямые
- аффинные преобразования
  - w’ = 1
  - сохраняется параллельность линий
  - пример: сдвиг
- преобразование подобия
  - сохраняются углы
  - пример: равномерное масштабирование
- изометрия (движение)
  - сохраняются расстояния
  - пример: поворот, перенос



$$\begin{pmatrix} x' \\ y' \\ z' \\ w' \end{pmatrix} = \begin{pmatrix} A & B & C & D \\ E & F & G & H \\ I & J & K & L \\ m & n & p & q \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Аффинные преобразования

Отображение $f:\mathbb{R}^n \to \mathbb{R}^n$ есть **аффинное преобразование**, если найдётся обратимая матрица M и вектор $v \in \mathbb{R}^n$ такие, что

$$f(x) = M \cdot x + v.$$

- прямая переходит в прямую, плоскость – в плоскость;
- сохраняется параллельность прямых и плоскостей;
- конечные (истинные) точки отображаются в конечные;
- сохраняются относительные пропорции;
- сохраняются аффинные комбинации точек.

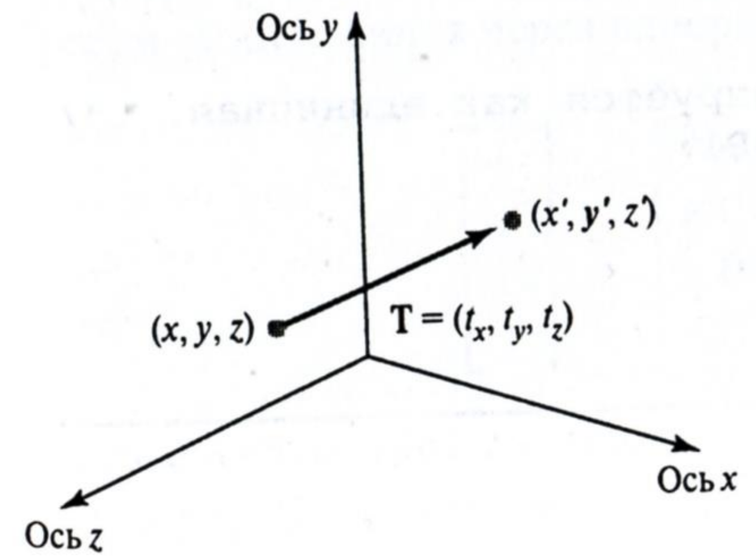
$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} A & B & C & D \\ E & F & G & H \\ I & J & K & L \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Трехмерный перенос

$$x' = x + \Delta x$$

$$y' = y + \Delta y$$

$$z' = z + \Delta z$$



$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & \Delta x \\ 0 & 1 & 0 & \Delta y \\ 0 & 0 & 1 & \Delta z \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

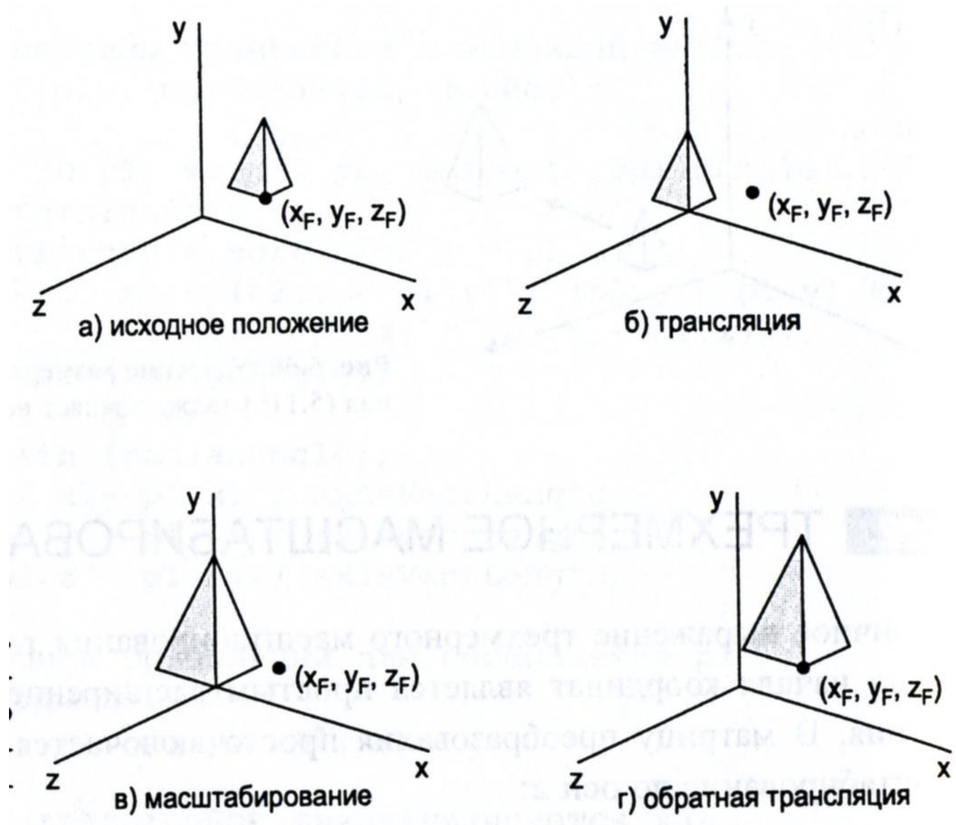
Трехмерное масштабирование

$$x' = ax$$

$$y' = by$$

$$z' = cz$$

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$



частные случаи:

- равномерное (пропорциональное) масштабирование;
- отражение от координатной плоскости

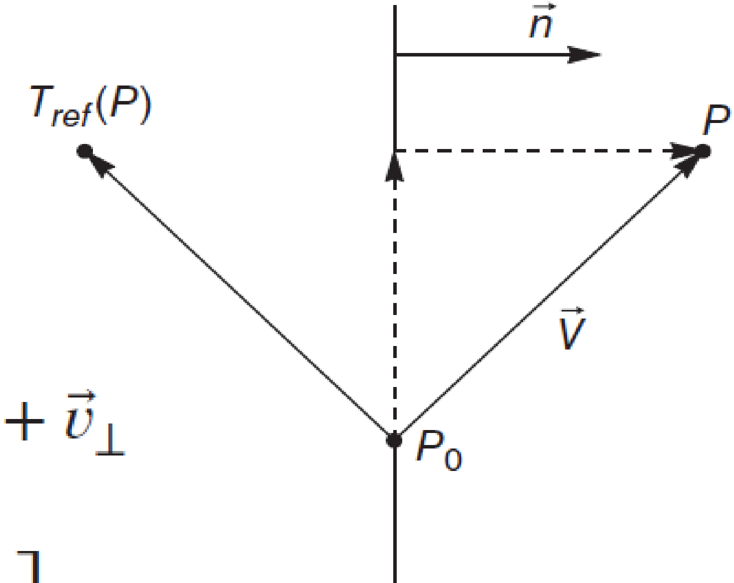
Отражение от произвольной плоскости

$$\vec{v} = \vec{v}_{\|} + \vec{v}_{\perp}$$

$$\vec{v}_{\|} = (\vec{v} \cdot \vec{n})\vec{n}$$

$$\vec{v}_{\perp} = \vec{v} - (\vec{v} \cdot \vec{n})\vec{n}$$

$$T_{ref}(\vec{v}) = T_{ref}(\vec{v}_{\|}) + T_{ref}(\vec{v}_{\perp}) = -\vec{v}_{\|} + \vec{v}_{\perp}$$



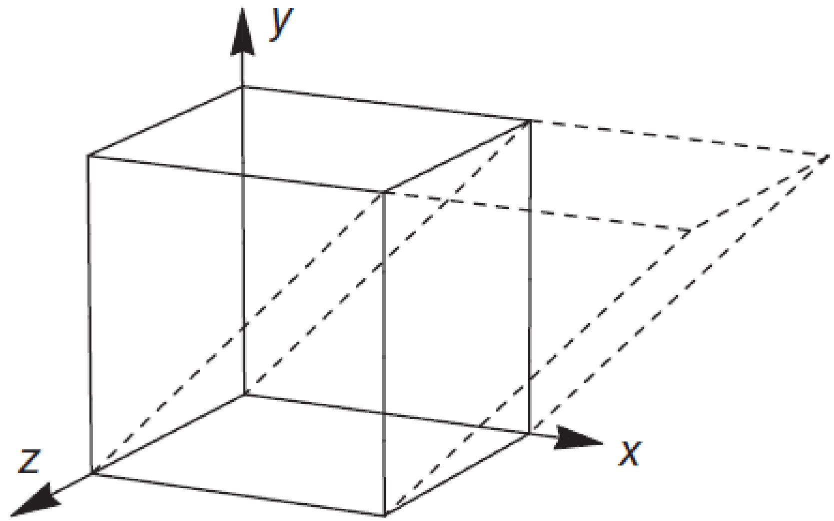
$$M = \vec{n}\vec{n}^T = \begin{bmatrix} n_x^2 & n_x n_y & n_x n_z \\ n_y n_x & n_y^2 & n_y n_z \\ n_z n_x & n_z n_y & n_z^2 \end{bmatrix}$$

$$T_{ref}(\vec{v}) = \vec{v} - 2(\vec{v} \cdot \vec{n})\vec{n} = I\vec{v} - 2M\vec{v} = (I - 2M)\vec{v}$$

Трехмерный сдвиг

$$x' = x + ay$$

$$y' = y + bx$$



$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} 1 & a & 0 & 0 \\ b & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

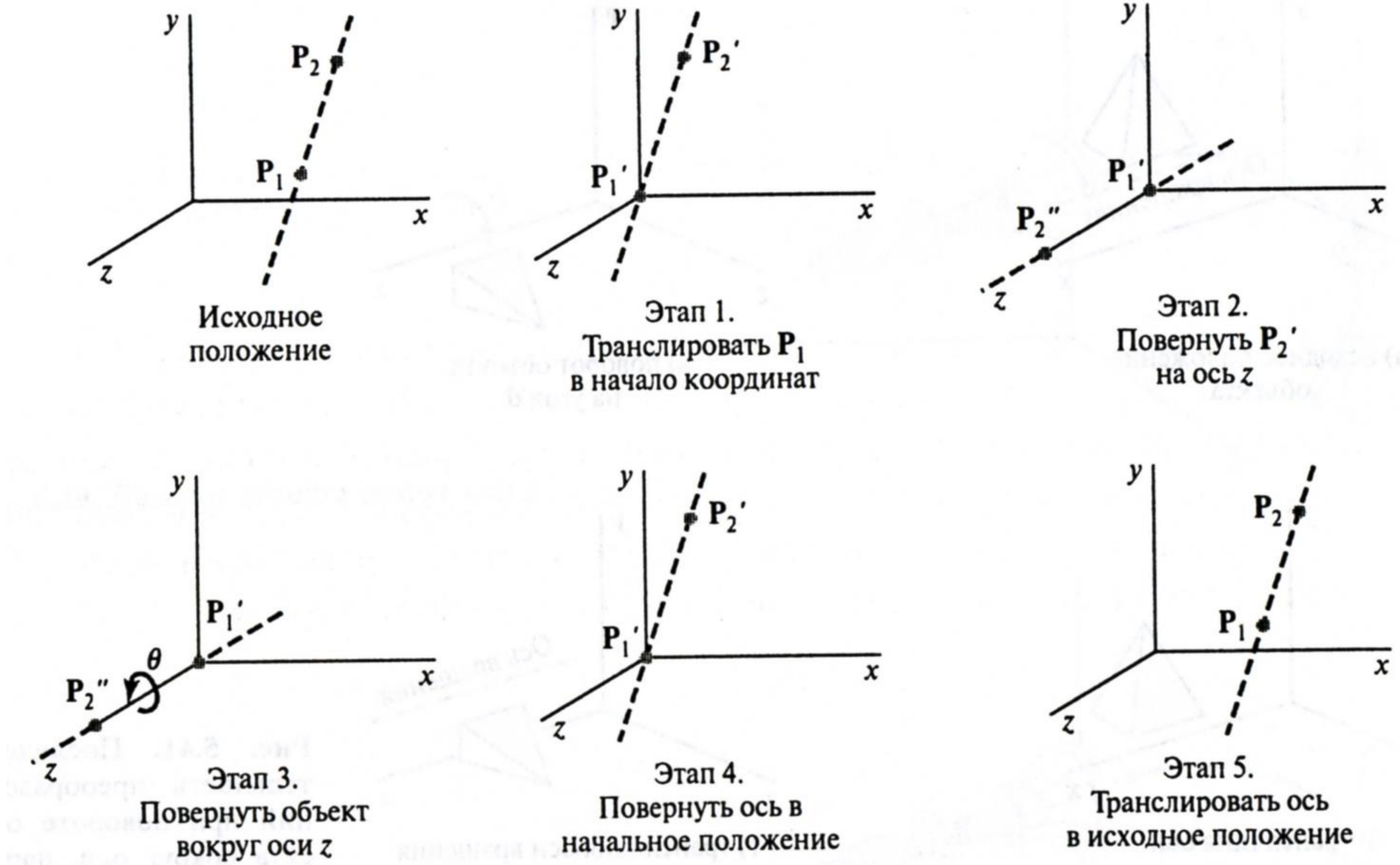
Трехмерный поворот (относительно одной из осей)

$$R_{OZ} = \begin{pmatrix} \cos\varphi & -\sin\varphi & 0 & 0 \\ \sin\varphi & \cos\varphi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_{OY} = \begin{pmatrix} \cos\varphi & 0 & \sin\varphi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\varphi & 0 & \cos\varphi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_{OX} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\varphi & -\sin\varphi & 0 \\ 0 & \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

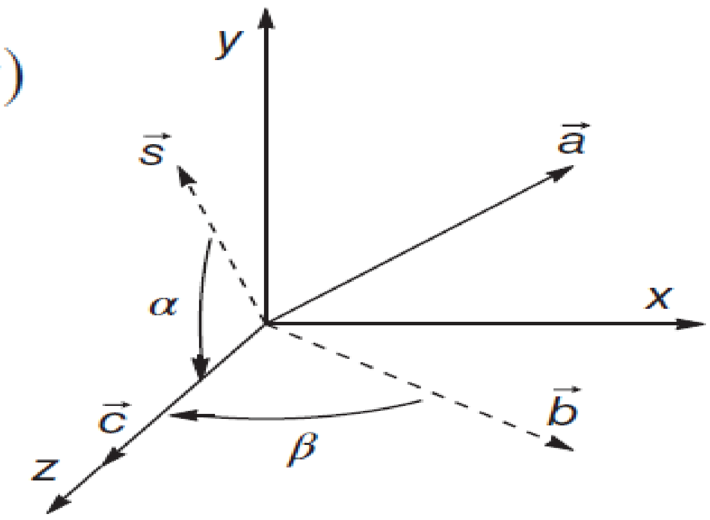
Трехмерный поворот



- Эйлер: комбинация любого числа поворотов всегда эквивалентна единственному повороту вокруг фиксированной оси

Трёхмерный поворот (относительно произвольной оси)

$$M_{arb} = M_x^{-1}(\alpha)M_y^{-1}(\beta)M_z(\theta)M_y(\beta)M_x(\alpha)$$



$$c = \cos\theta$$

$$s = \sin\theta$$

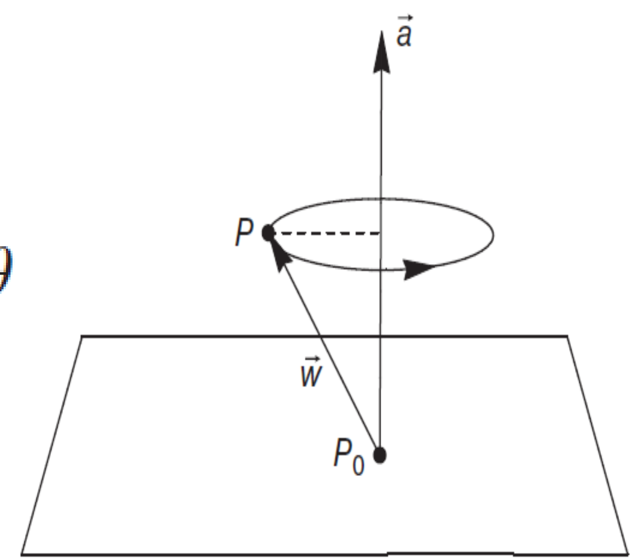
$$M_{arb} = \begin{bmatrix} c+(1-c)a_x^2 & (1-c)a_xa_y - sa_z & (1-c)a_xa_z + sa_y \\ (1-c)a_xa_y + sa_z & c+(1-c)a_y^2 & (1-c)a_ya_z - sa_x \\ (1-c)a_xa_z - sa_y & (1-c)a_ya_z + sa_x & c+(1-c)a_z^2 \end{bmatrix}$$

Трёхмерный поворот (относительно произвольной оси) (2)

$$\vec{w}_{\|} = (\vec{a}\cdot\vec{w})\vec{a}$$

$$\vec{w}_{\perp} = \vec{w} - (\vec{a}\cdot\vec{w})\vec{a}$$

$$\vec{w}_r = (\vec{w} - (\vec{a}\cdot\vec{w})\vec{a})\cos\theta + (\vec{a}\times\vec{w}_{\perp})\sin\theta$$



$$(\vec{a}\cdot\vec{w})\vec{a} = (\vec{a}\otimes\vec{a})\vec{w} = \begin{bmatrix} a_x^2 & a_x a_y & a_x a_z \\ a_y a_x & a_y^2 & a_y a_z \\ a_z a_x & a_z a_y & a_z^2 \end{bmatrix}\vec{w}$$

$$\vec{a}\times\vec{w} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}\vec{w} = C_a\vec{w}$$

$$T_{rot}(\vec{w}) = M_{rot}\vec{w} = (I\cos\theta + (\vec{a}\otimes\vec{a})(1-\cos\theta) + C_a\sin\theta)\vec{w}$$

Перемещение и ориентация

- положение объекта в пространстве задаётся с помощью шести параметров (6DoF, six degrees of freedom):
  - связанные с перемещением:
    - вперёд / назад (surging);
    - влево / вправо (swaying);
    - вверх / вниз (heaving);
  - связанные с поворотом:
    - крен (roll) : поворот вокруг продольной оси;
    - тангаж (pitch) : поворот вокруг поперечной оси;
    - рыскание (yaw) : поворот относительно вертикальной оси;
- описание поворотов:
  - углы Эйлера (определяют три угла поворота системы, которые позволяют привести исходное положение к текущему) → Gimbal lock;
  - кватернионы (Slerp: spherical linear interpolation).

Вращение с использованием кватернионов

$$\hat{q} = (\cos\theta, \vec{u}\sin\theta)$$

$$\hat{v} = (0, \vec{v})$$

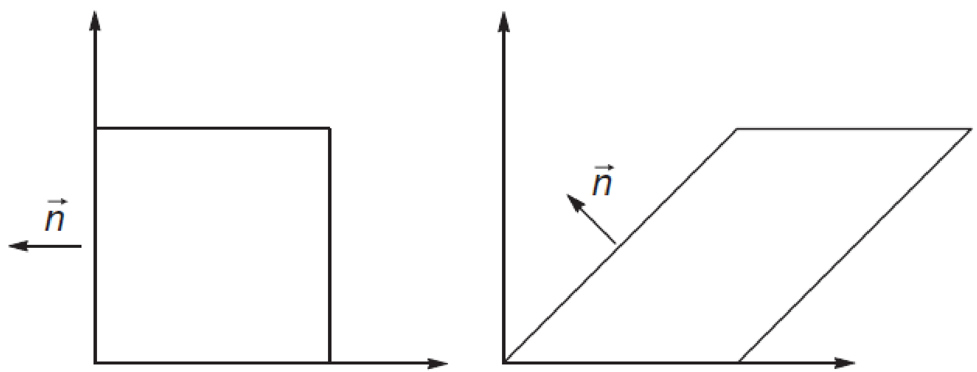
$T(\hat{v}) = \hat{q}\hat{v}\hat{q}^{-1}$ *rotates* $\vec{v}$ *around the axis* $\vec{u}$ *through an angle* 2θ

$$\hat{q} = (a, \vec{w}) = (a, b, c, d)$$

$$R(\hat{q}) = \begin{bmatrix} a^2+b^2-c^2-d^2 & 2bc-2ad & 2bd+2ac \\ 2bc+2ad & a^2-b^2+c^2-d^2 & 2cd-2ab \\ 2bd-2ac & 2cd+2ab & a^2-b^2-c^2+d^2 \end{bmatrix}$$

- Slerp: $\hat{q}(t) = \frac{\sin(1-t)\theta}{\sin\theta}\hat{q}_1 + \frac{\sin t\theta}{\sin\theta}\hat{q}_2$

Преобразование нормалей



$$L\vec{n} \cdot M\vec{v} = 0$$

$$(L\vec{n})^T M\vec{v} = \vec{n}^T L^T M\vec{v} = 0$$

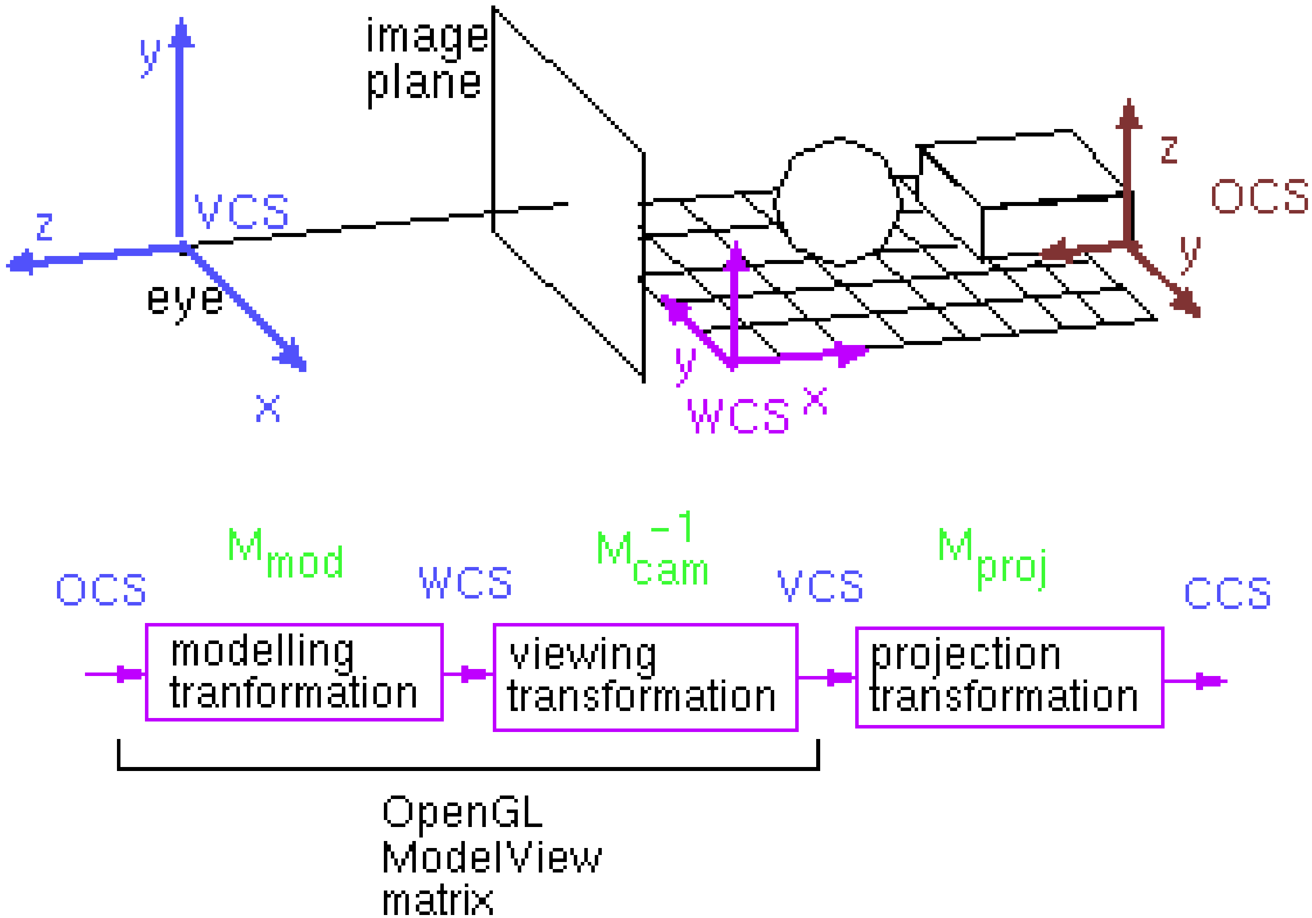
$$L = (M^{-1})^T$$

$$\vec{n} \cdot \vec{v} = \vec{n}^T \vec{v} = \begin{bmatrix} n_1 & n_2 & n_3 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = 0$$

- преобразования вершин и нормалей совпадают в том случае, когда преобразование задано ортогональной матрицей (например, повороты)

$$M = (M^{-1})^T$$

Преобразования в графическом конвейере OpenGL



Функции операций с матрицами преобразований в OpenGL

- Выбор матрицы преобразований для изменения:

```
void glMatrixMode(Glenum mode =
       {GL_MODELVIEW | GL_PROJECTION | GL_TEXTURE})
```

- Основные операции над матрицами:

```
void glLoadIdentity();
```

$$M = I$$

```
void glLoadMatrixd(GLdouble m[16]);

void glMultMatrixd(GLdouble m[16]);
```

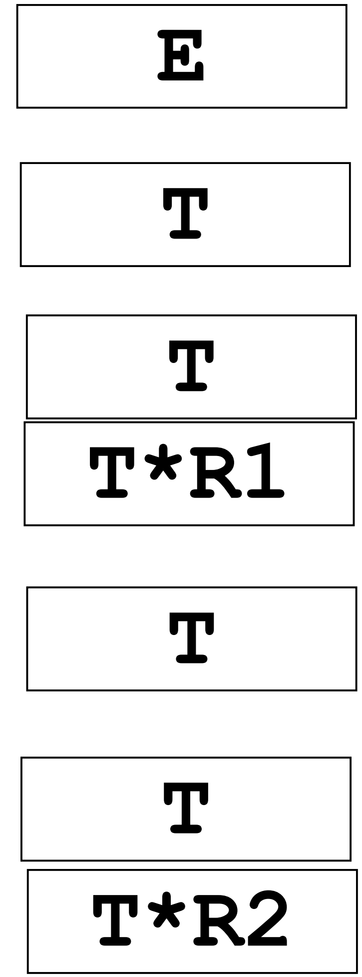
$$V = C \cdot \begin{bmatrix} m[0] & m[4] & m[8] & m[12] \\ m[1] & m[5] & m[9] & m[13] \\ m[2] & m[6] & m[10] & m[14] \\ m[3] & m[7] & m[11] & m[15] \end{bmatrix} \cdot \begin{bmatrix} v[0] \\ v[1] \\ v[2] \\ v[3] \end{bmatrix}$$

Стек матриц в OpenGL

```
glLoadIdentity();
glTranslated(…);

glPushMatrix();
glRotated(…);
glPopMatrix();

glPushMatrix();
glRotated(…);
glPopMatrix();
```



Определение глубины стека: glGetIntegerv (GL_MAX_MODELVIEW_STACK_DEPTH)

Определение текущей глубины стека: glGetIntegerv (GL_MODELVIEW_STACK_DEPTH)

Функции геометрических преобразований в OpenGL

```
void glTranslated(GLdouble x,
                  GLdouble y,
                  GLdouble z);
```

```
void glScaled(GLdouble x,
              GLdouble y,
              GLdouble z);
```

```
void glRotated(GLdouble angle,
               GLdouble ax,
               GLdouble ay,
               GLdouble az);
```

- преобразования объектов и камеры в OpenGL производятся с помощью умножения векторов координат на *текущую матрицу* в момент определения координат вершин
- при определении преобразования умножение матриц производится аналогично функции glMultMatrixd(), следовательно, преобразования необходимо задавать в обратном порядке.

Видовое преобразование в OpenGL

- стандартная камера в OpenGL:
  - наблюдатель в (0, 0, 0)
  - смотрит по направлению (0, 0, -1)
  - верх (0, 1, 0)

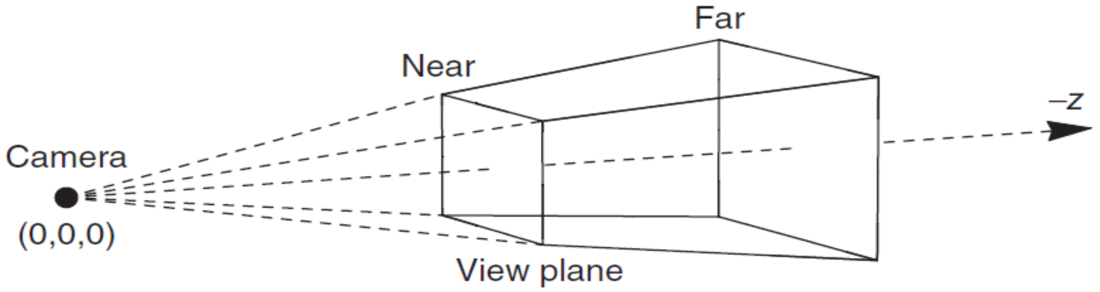
$$\text{gluLookAt}(\,eye_x,\ eye_y,\ eye_z,\ aim_x,\ aim_y,\ aim_z,\ up_x,\ up_y,\ up_z)$$

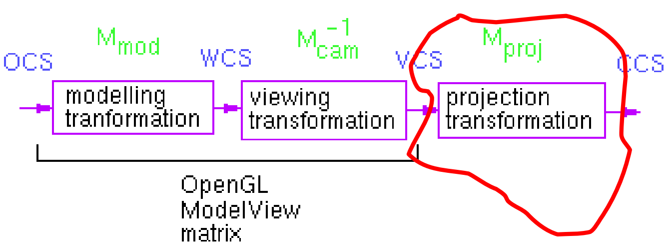
- eye – координаты наблюдателя
- aim – координаты «цели» (центра сцены)
- up – направление «вверх»

Вызывать команду gluLookAt() имеет смысл *перед* определением преобразований объектов, когда модельно-видовая матрица равна единичной

Проективное преобразование и проекция

- проективное преобразование: реализует 3D преобразование, подготавливая модель к переходу к 2D (координата Z сохраняется);
- проекция: переход к двумерной системе координат;
- после проективного преобразования необходимо отбросить координату z и получить значения в оконных координатах;
- объём видимости: ограничивает видимую область пространства;





Проекции

основные понятия:

- проецирующие лучи (проекторы)
- поверхность проекции, плоскость проекции
- центр проекции
- главная ось: ось координат
- перспективное искажение, точки схода
- коэффициент укорачивания

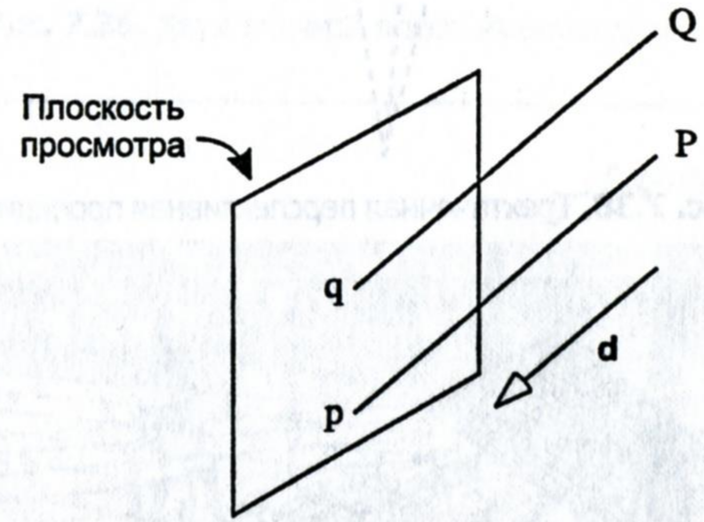
Классификация проекций

- параллельная:
  - ортогональная, ортографическая*;
  - аксонометрическая:
    - прямоугольная * :
      - изометрия;
      - диметрия;
      - триметрия;
    - косоугольная **:
      - кавалье / военная перспектива (горизонтальная изометрия);
      - кабинетная (фронтальная диметрия);
- перспективная (центральная):
  - одноточечная;
  - двухточечная;
  - трехточечная;

Параллельные проекции

Уравнение плоскости (1): $\mathbf{n}\cdot(Q-B)=0$

Проецируемая точка (2): $P+\mathbf{d}t$:



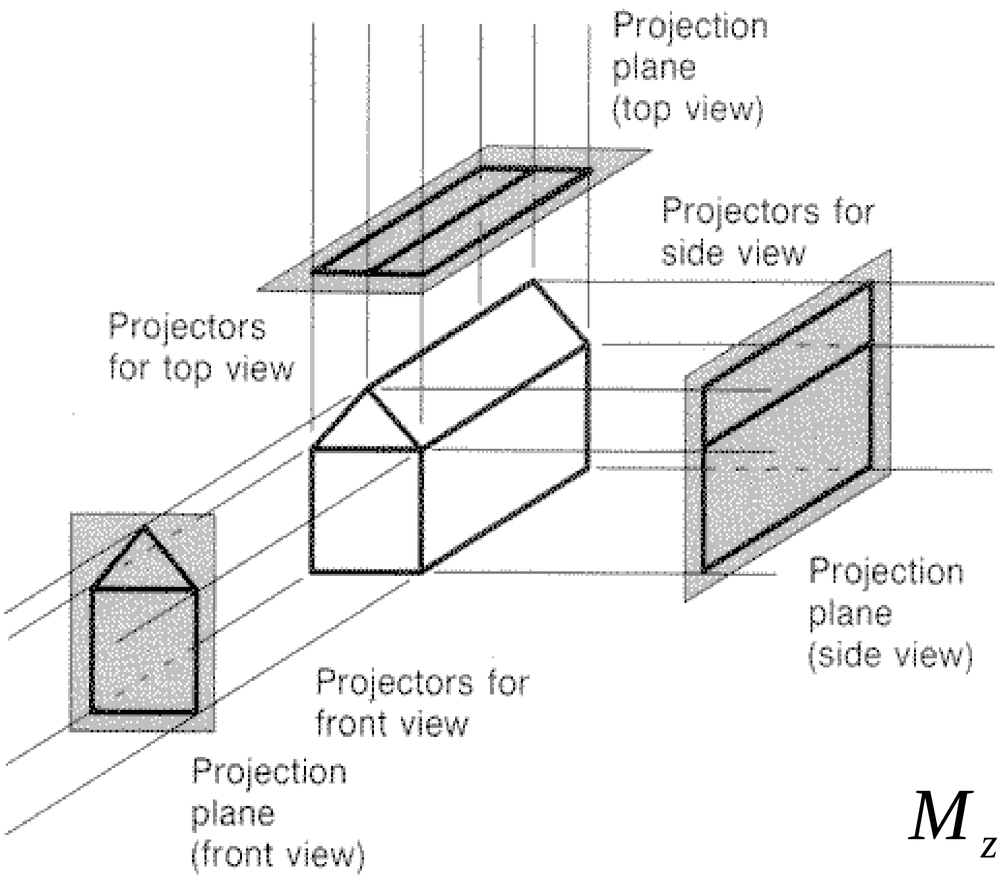
$$p=P+\mathbf{d}\frac{\mathbf{n}\cdot(B-P)}{\mathbf{n}\cdot\mathbf{d}}.$$

Условия $\mathbf{n}=(0,0,1)$, $B=(0,0,0)$.

$$p=\left(P_x-d_x\frac{P_z}{d_z},\ P_y-d_y\frac{P_z}{d_z},\ 0\right).$$

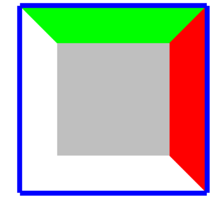
$$\begin{pmatrix}p_x\\p_y\\p_z\\1\end{pmatrix}=\begin{pmatrix}1&0&-\frac{d_x}{d_z}&0\\0&1&\frac{-d_y}{d_z}&0\\0&0&0&0\\0&0&0&1\end{pmatrix}\begin{pmatrix}P_x\\P_y\\P_z\\1\end{pmatrix}.$$

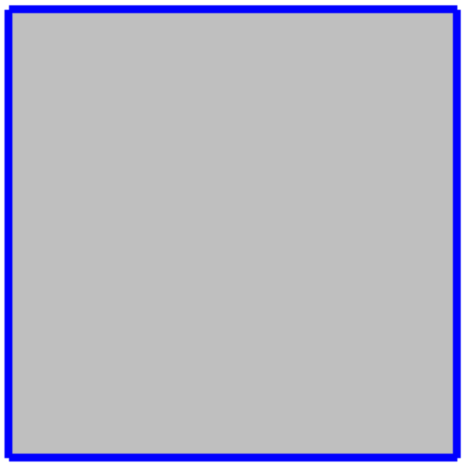
Ортографическая проекция (виды)



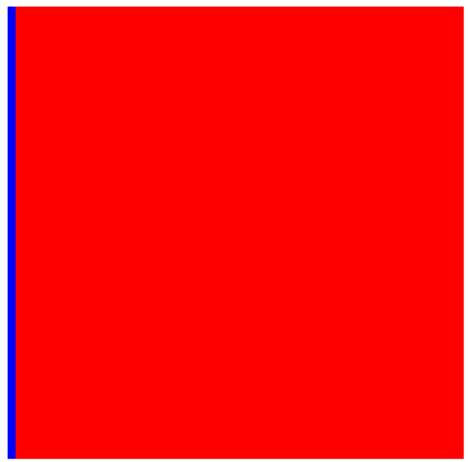
$$M_z = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & d = -Z_0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Ортографическая проекция: пример

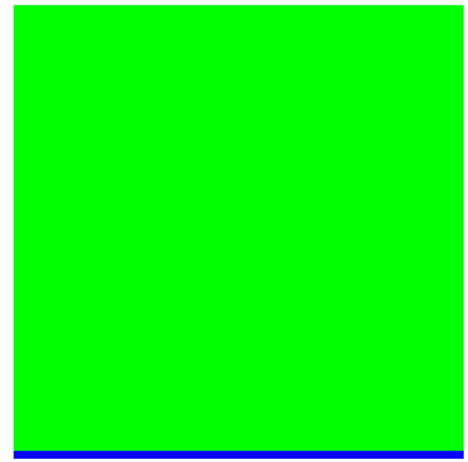




вид спереди
(front view)



вид слева
(side view)



вид сверху
(top view)

$$M_z = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$M_x = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$M_y = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Прямоугольные аксонометрические проекции

$$[T][U] = [T]\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} x_x^* & x_y^* & x_z^* \\ y_x^* & y_y^* & y_z^* \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

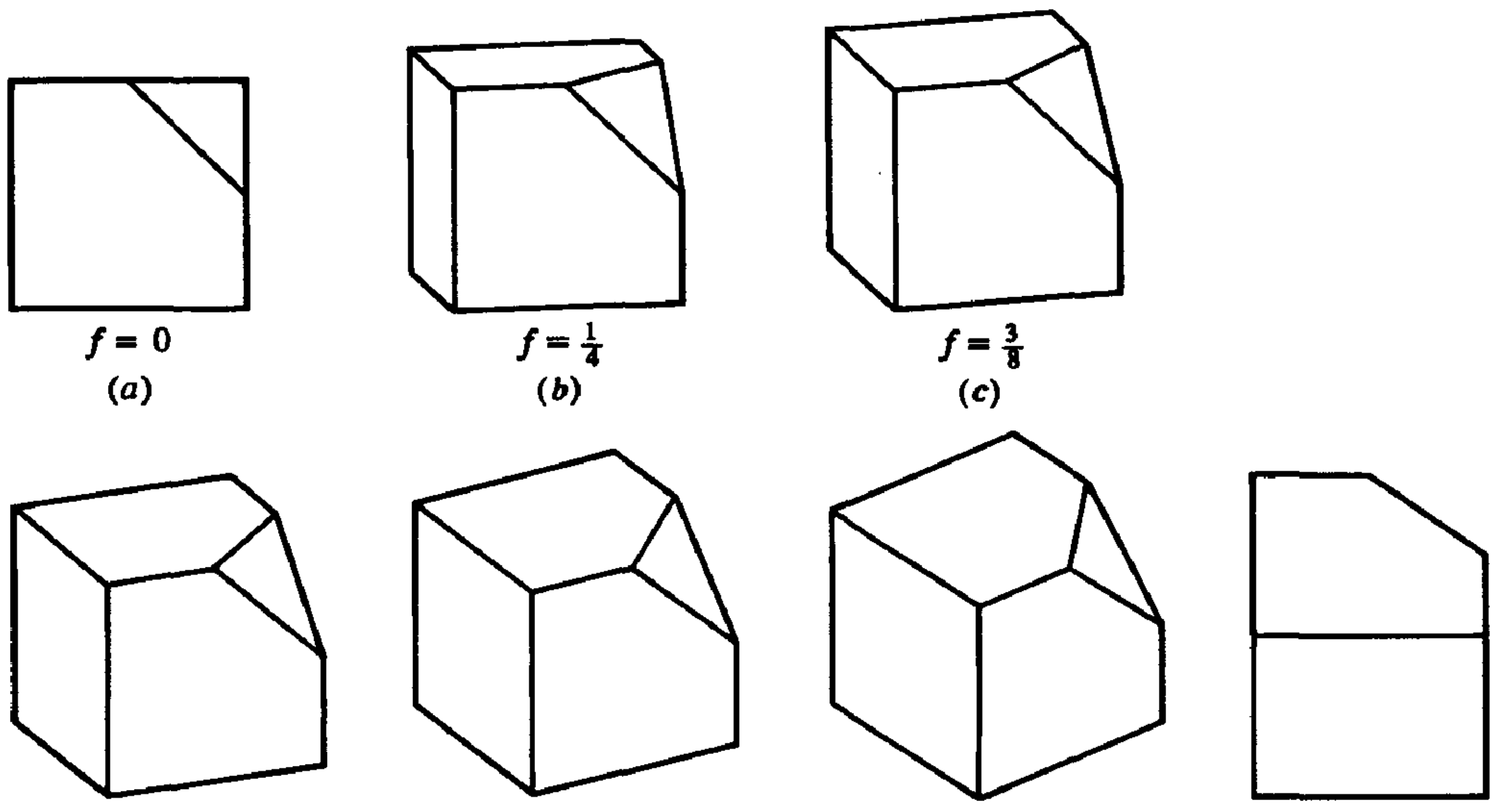
$$f_x = \sqrt{x_x^{*2} + y_x^{*2}},$$
$$f_y = \sqrt{x_y^{*2} + y_y^{*2}},$$
$$f_z = \sqrt{x_z^{*2} + y_z^{*2}}.$$

коэффициенты искажения вдоль главных осей

$$[T] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} \cos\varphi & 0 & \sin\varphi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\varphi & 0 & \cos\varphi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Диметрия

$$\theta = \arcsin(\pm f_z/\sqrt{2}). \quad \phi = \arcsin(\pm f_z/\sqrt{2 - f_z^2}).$$



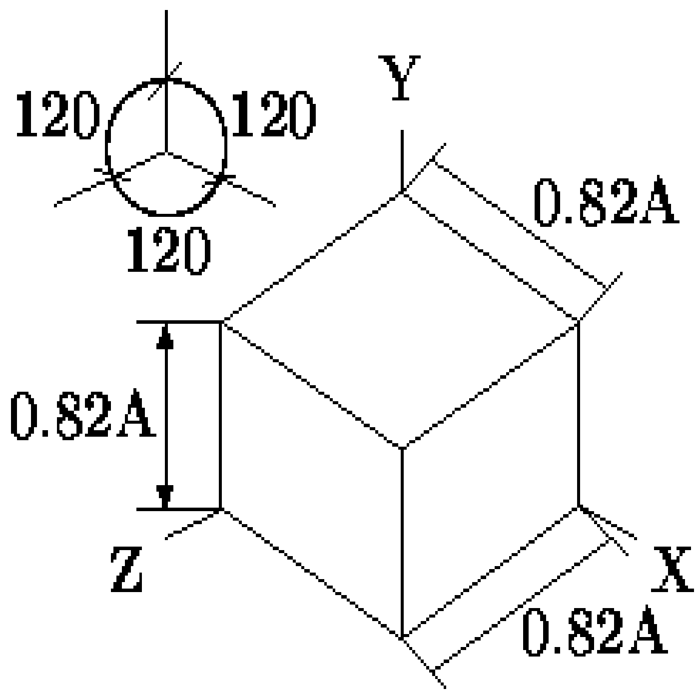
$f = 0$ (a) $f = \frac{1}{4}$ (b) $f = \frac{3}{8}$ (c)

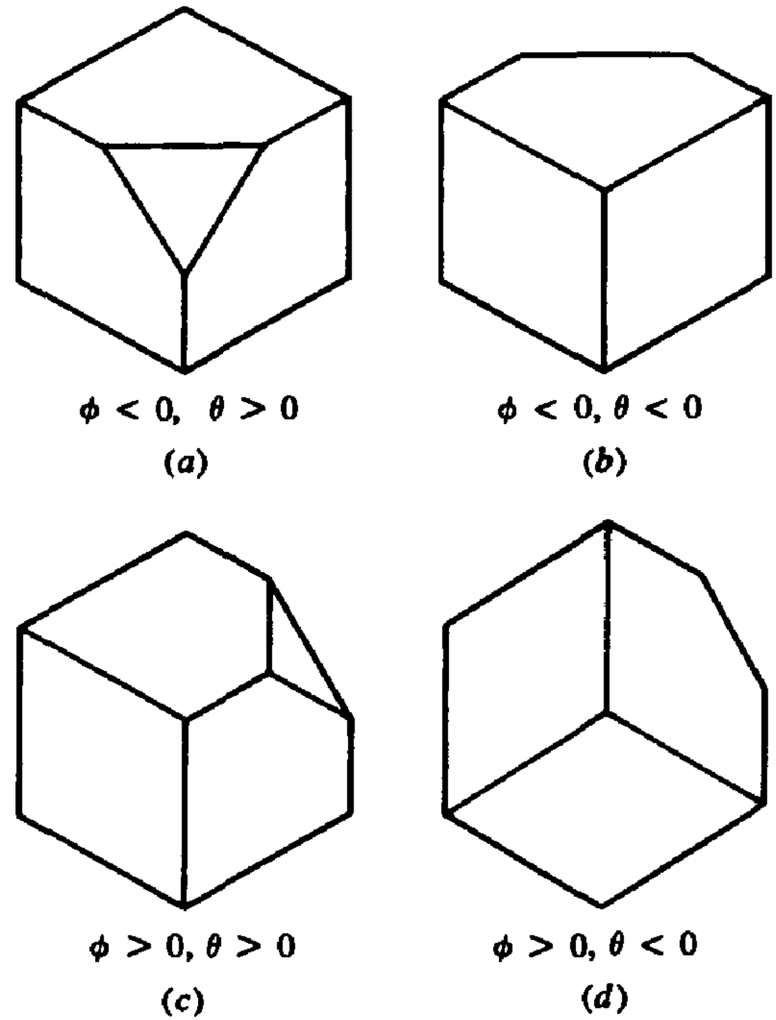
$f = \frac{1}{2}$ (d) $f = \frac{5}{8}$ (e) $f = \frac{3}{4}$ (f) $f = 1$ (g)

Изометрия

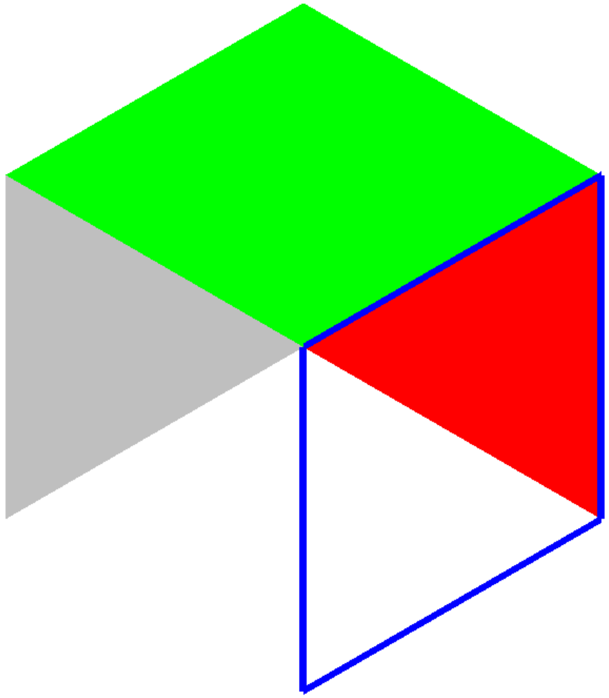
$$\theta = \pm 35,26^\circ$$

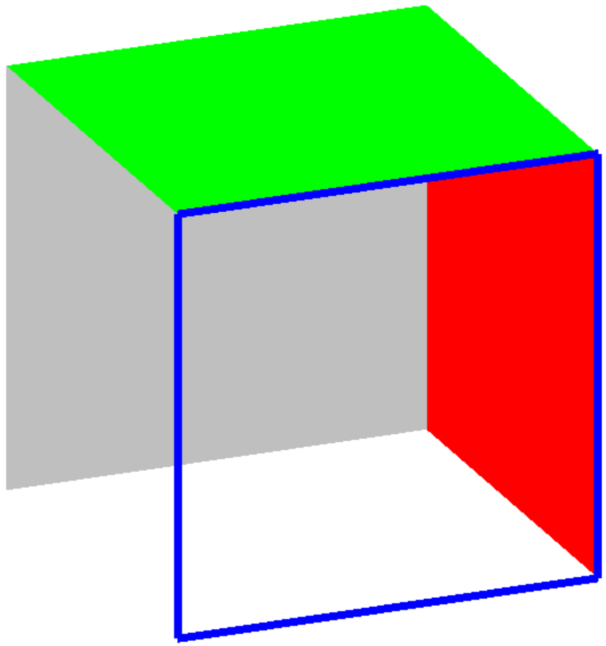
$$\phi = \pm 45^\circ$$





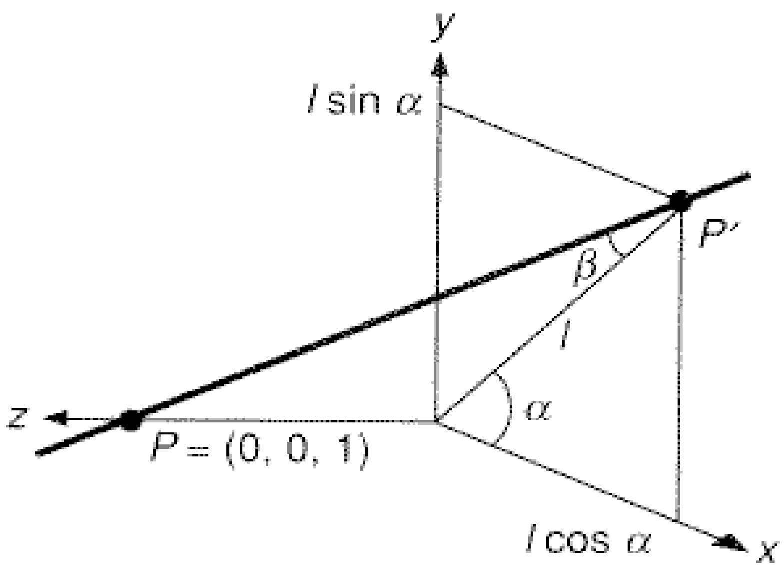
Изометрия и диметрия: пример





$$[T] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\varphi & 0 & \sin\varphi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\varphi & 0 & \cos\varphi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \cos\varphi & 0 & \sin\varphi & 0 \\ \sin\varphi\sin\theta & \cos\theta & -\cos\varphi\sin\theta & 0 \\ \sin\varphi\cos\theta & -\sin\theta & -\cos\varphi\cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Косоугольные аксонометрические проекции



- Направление проецирования

  $(l\cos\alpha, l\sin\alpha, -1)$
- Угол между косыми проекторами и плоскостью проекции

  $\beta = arcctg(l)$
- Военная проекция (кавалье)

  $\beta = arcctg(1) = 45°$
- Кабинетная проекция

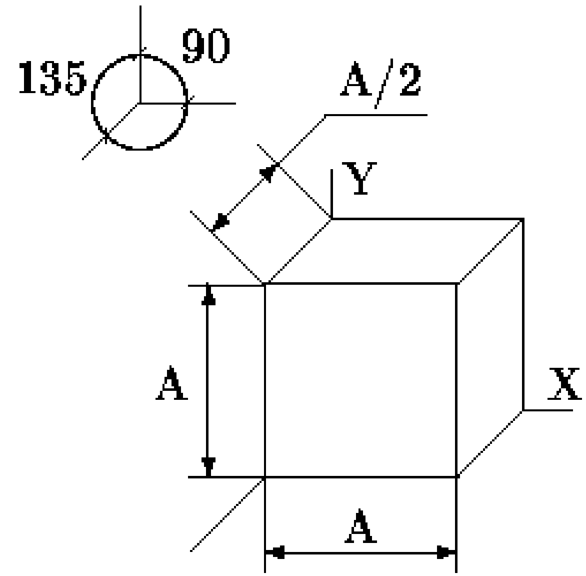
  $\beta = arcctg(1/2) = 63.435°$

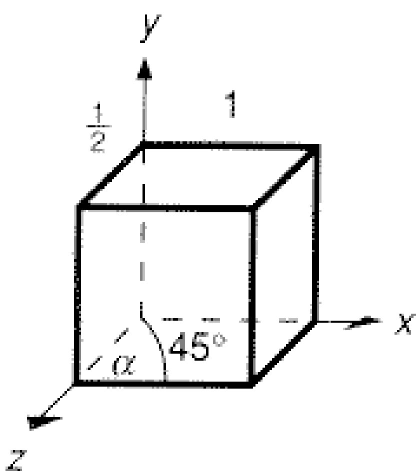
$$[T] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & -l*\cos\alpha & 0 \\ 0 & 1 & -l*\sin\alpha & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} p_x \\ p_y \\ p_z \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & -\frac{d_x}{d_z} & 0 \\ 0 & 1 & \frac{-d_y}{d_z} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} P_x \\ P_y \\ P_z \\ 1 \end{pmatrix}.$$

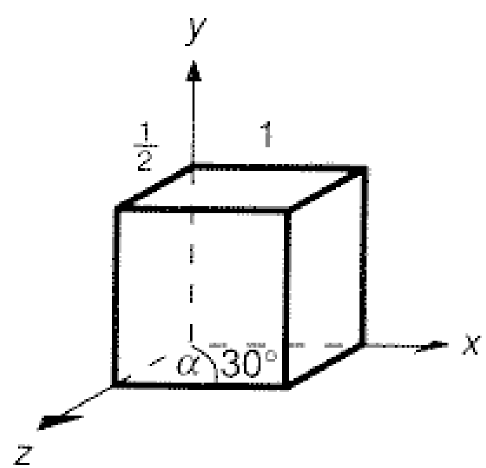
Фронтальная диметрия (Cabinet)

- плоскость проецирования перпендикулярна оси Z





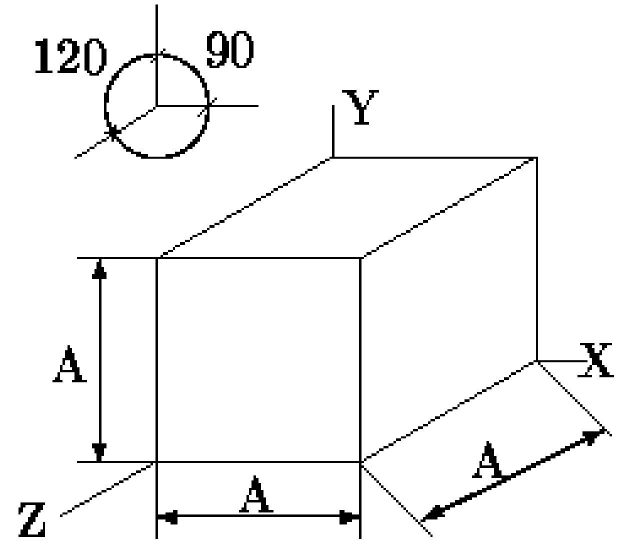
(a)

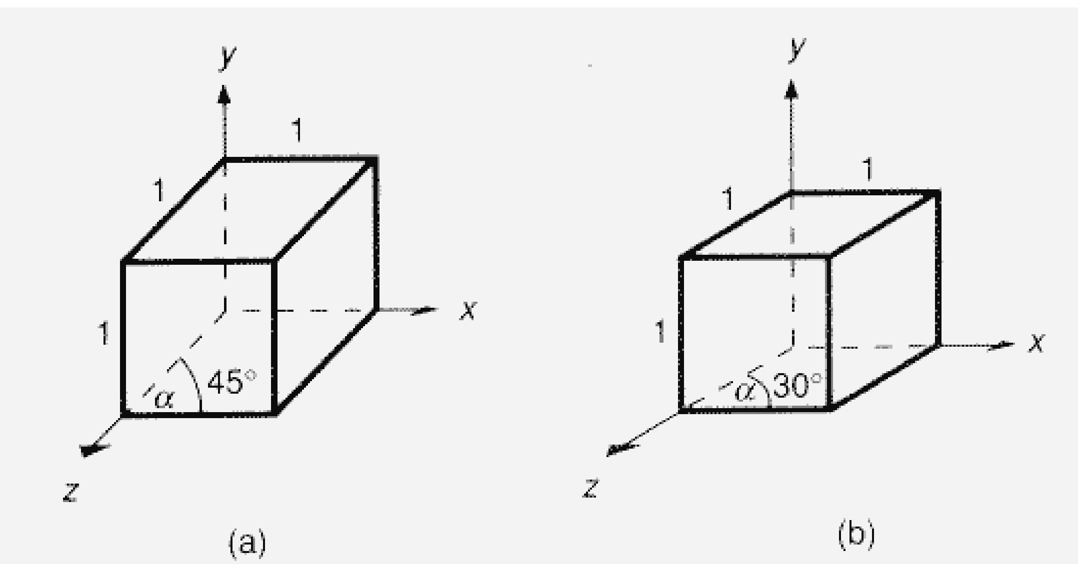


(b)

Горизонтальная изометрия (Cavalier, военная перспектива)

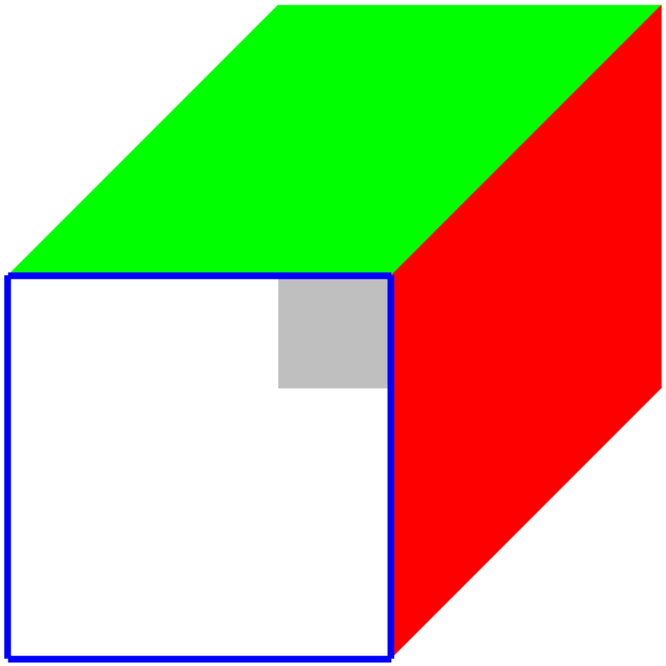
- плоскость проецирования перпендикулярна оси Y

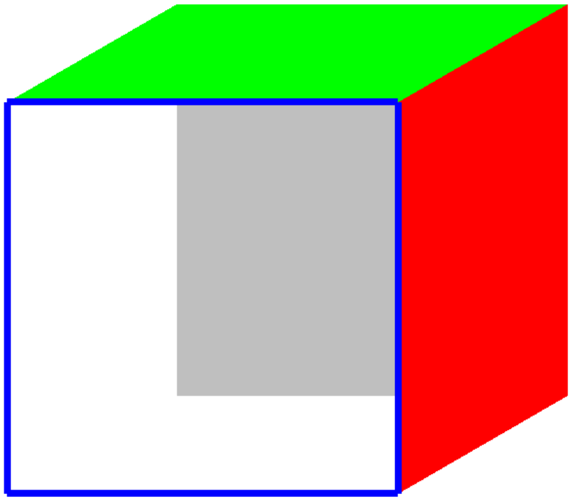




(a)
(b)

Горизонтальная изометрия и фронтальная диметрия: пример

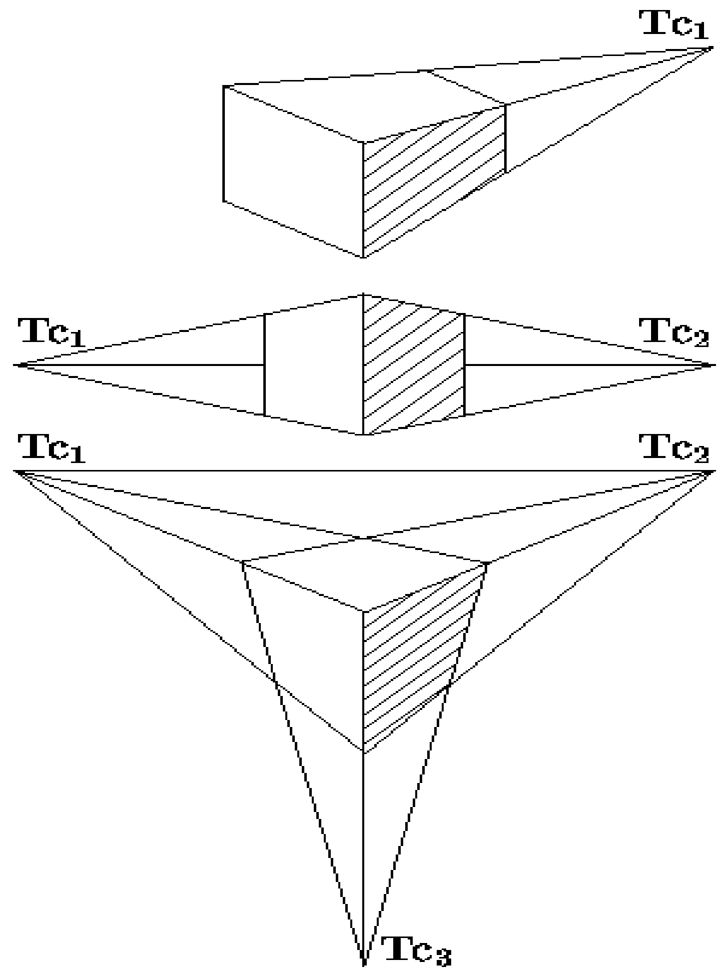




$$[T]=\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & -1 & 0\\ 0 & 0 & 0 & 1\end{pmatrix}\begin{pmatrix}1 & 0 & -l*\cos\alpha & 0\\ 0 & 1 & -l*\sin\alpha & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{pmatrix}=\begin{pmatrix}1 & 0 & -l*\cos\alpha & 0\\ 0 & 1 & -l*\sin\alpha & 0\\ 0 & 0 & -1 & 0\\ 0 & 0 & 0 & 1\end{pmatrix}$$

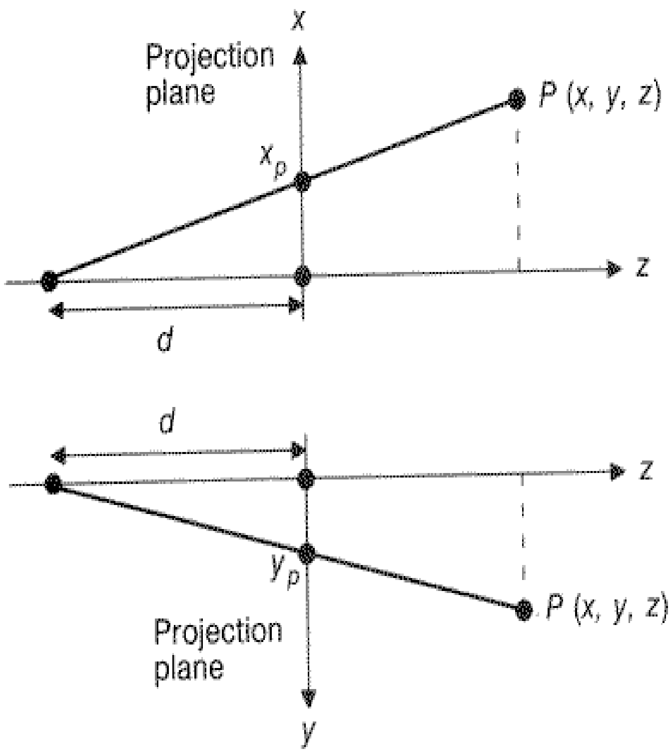
Центральные (перспективные) проекции

Центральные проекции параллельных прямых, не параллельных плоскости проекции будут сходиться в *точке схода,* количество которых (для прямых, параллельных осям координат), зависит от числа координатных осей, которые пересекает плоскость проекции



$$[T]=\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ p & q & r & 1 \end{pmatrix}$$

Перспективная проекция: матричное представление



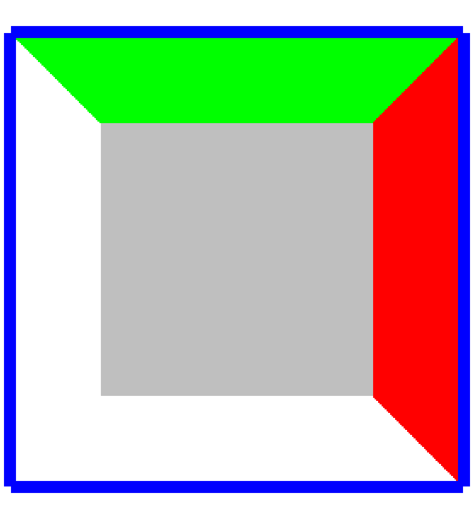
$$M_{per} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1/d & 1 \end{bmatrix}$$

$$x_p = \frac{d \cdot x}{z + d} = \frac{x}{(z/d) + 1}$$

$$y_p = \frac{d \cdot y}{z + d} = \frac{y}{(z/d) + 1}$$

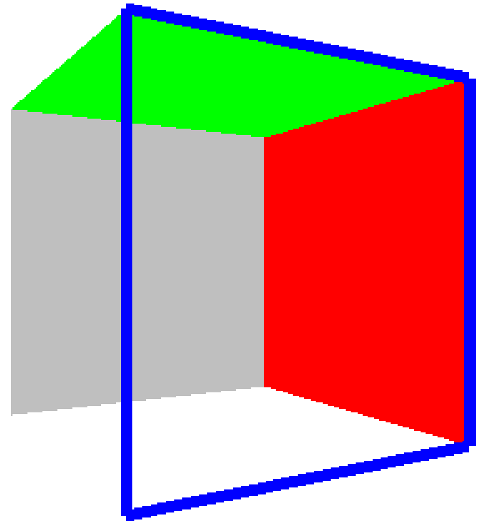
$$\frac{x_p}{d} = \frac{x}{z + d}, \frac{y_p}{d} = \frac{y}{z + d},$$

Перспективная проекция: пример



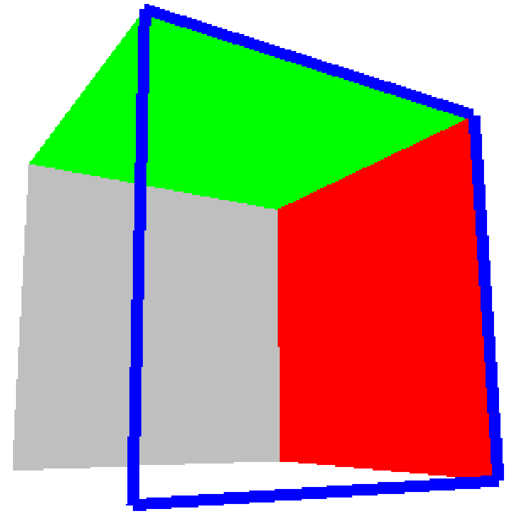
- одноточечная

$$M_{1P} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -2 & 1 \\ 0 & 0 & -1 & 2 \end{pmatrix}$$



- двухточечная

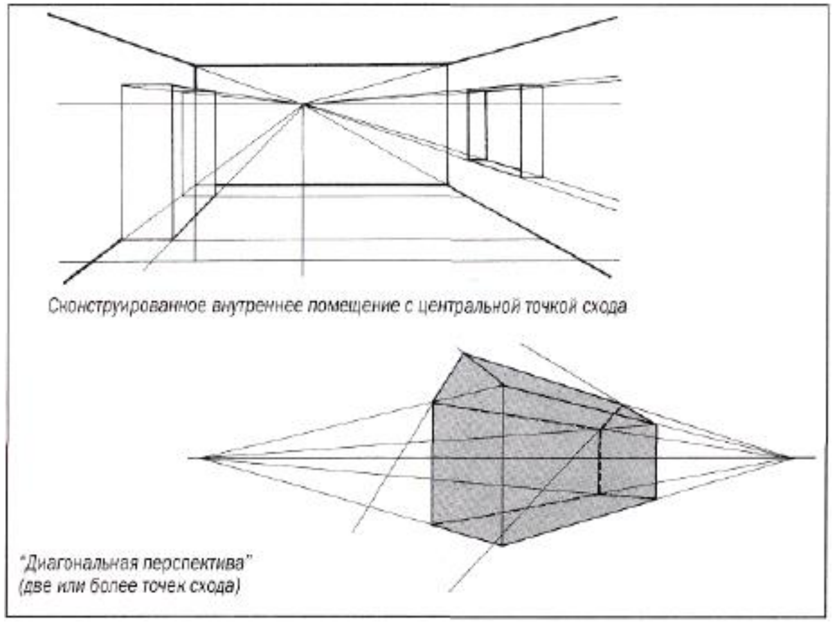
$$M_{2P} = \begin{pmatrix} 0{,}87 & 0 & 0{,}5 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & -1{,}73 & 1 \\ 0{,}5 & 0 & -0{,}87 & 2 \end{pmatrix}$$



- трёхточечная

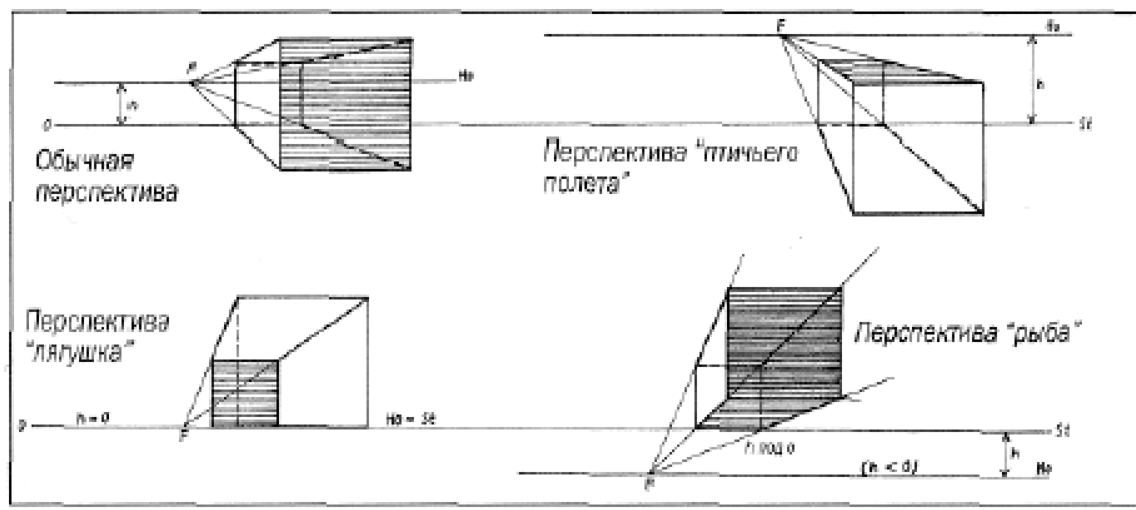
$$M_{3P} = \begin{pmatrix} 0{,}87 & 0 & 0{,}5 & 0 \\ -0{,}09 & 0{,}98 & 0{,}15 & 0 \\ 0{,}98 & 0{,}35 & -1{,}7 & 1 \\ 0{,}49 & 0{,}17 & -0{,}85 & 2 \end{pmatrix}$$

Центральная перспектива

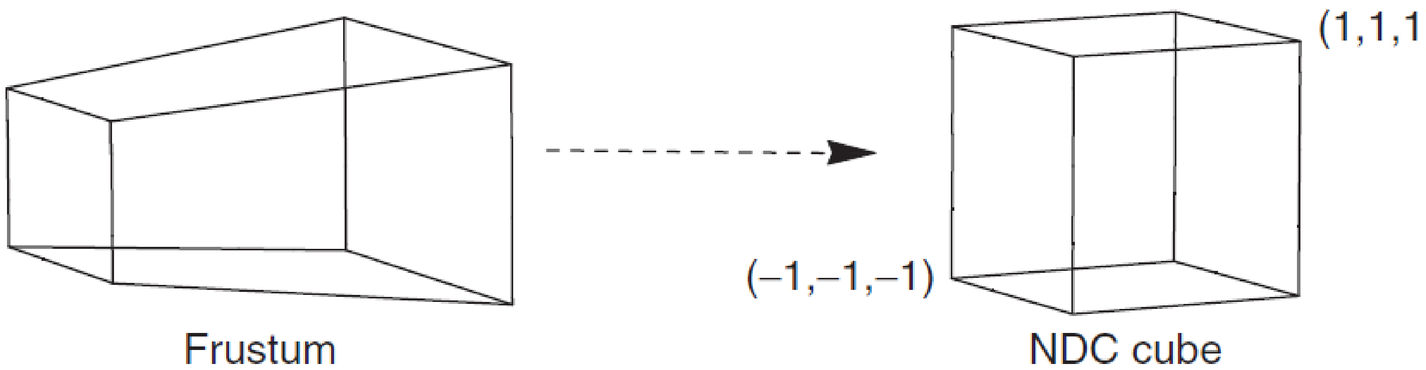


искаженная перспектива

- выбор центра проецирования для достижения желаемого эффекта при восприятии
- несоответствие положения наблюдателя и центра проекции



Нормализованные координаты



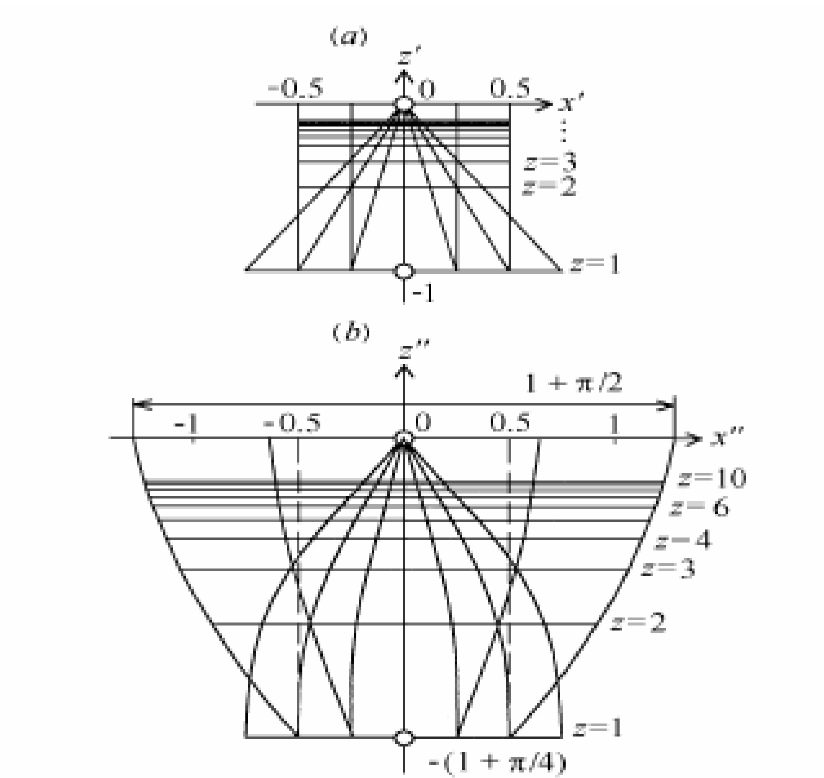
$$M_{per} = \begin{bmatrix} \dfrac{-2z_n}{w_r - w_l} & 0 & \dfrac{w_r + w_l}{w_r - w_l} & 0 \\ 0 & \dfrac{-2z_n}{w_t - w_b} & \dfrac{w_t + w_b}{w_t - w_b} & 0 \\ 0 & 0 & \dfrac{z_n + z_f}{z_n - z_f} & -\dfrac{2z_n z_f}{z_n - z_f} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

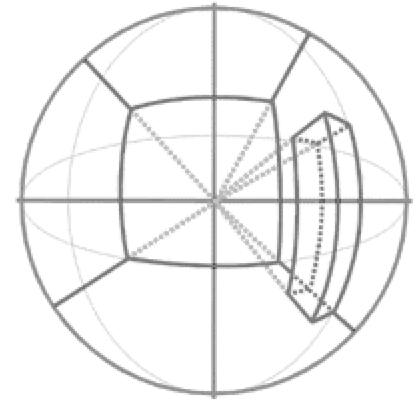
- интерполяция с учётом перспективы

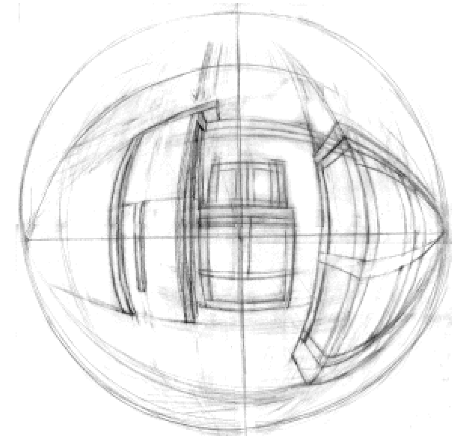
$$\frac{1}{z_2} = (1 - s)\frac{1}{z_1} + s\frac{1}{z_3}$$

$$\frac{c_2}{z_2} = (1 - s)\frac{c_1}{z_1} + s\frac{c_3}{z_3}$$

Нелинейная перспектива







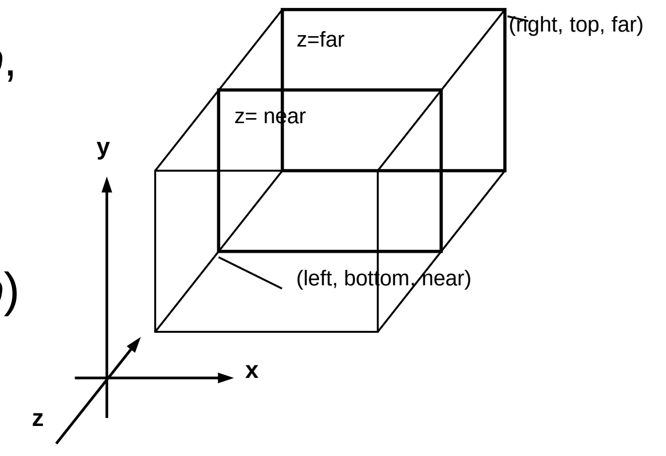
Пространство изображений:
а – линейное,
б – нелинейное.

Сферическая перспектива (сохраняет угловые размеры объектов)

Ортографическое проективное преобразование в OpenGL

void **glOrtho** (GLdouble *left*, GLdouble *right*,
GLdouble *bottom*, GLdouble *top*,
GLdouble *near*, GLdouble *far*)

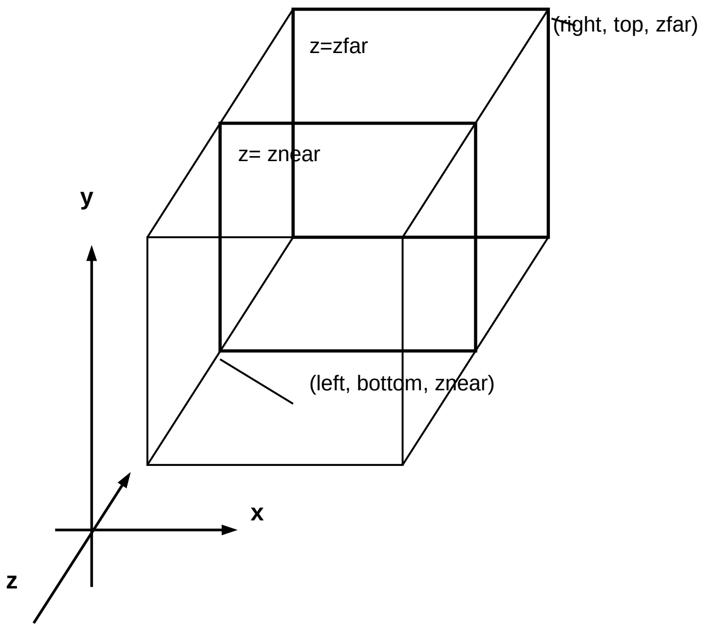
void **gluOrtho2D** (GLdouble *left*, GLdouble *right*,
GLdouble *bottom*, GLdouble *top*)
// near = -1, far = 1



Определяет параллелепипед видимости, ребра которого направлены вдоль осей координат, с точкой наблюдения в (0, 0, 0), используется левосторонняя система координат. Параметры команды задают точки (left, bottom, znear) и (right, top, zfar), которые отвечают левому нижнему и правому верхнему углам окна вывода. Параметры near и far задают расстояние до ближней и дальней плоскостей отсечения по удалению от точки (0,0,0) и могут быть отрицательными.

Матрица ортографического проективного преобразования в OpenGL

$$\begin{pmatrix} \frac{2}{right-left} & 0 & 0 & t_x \\ 0 & \frac{2}{top-bottom} & 0 & t_y \\ 0 & 0 & \frac{-2}{far-near} & t_z \\ 0 & 0 & 0 & -1 \end{pmatrix}$$



$$t_x = -\frac{right+left}{right-left}$$

$$t_y = -\frac{top+bottom}{top-bottom}$$

$$t_z = -\frac{far+near}{far-near}$$

Определяет параллелепипед видимости, ребра которого направлены вдоль осей координат, с точкой наблюдения в (0, 0, 0), используется левосторонняя система координат.

Перспективное проективное преобразование в OpenGL (определение пирамиды видимости)

void glFrustum(GLdouble *left*, **GLdouble** *right*, **GLdouble** *bottom*, **GLdouble** *top*, **GLdouble** *znear*, **GLdouble** *zfar* **);**

$$\begin{pmatrix} \frac{2\,near}{right-left} & 0 & A & 0 \\ 0 & \frac{2\,near}{top-bottom} & B & 0 \\ 0 & 0 & C & D \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

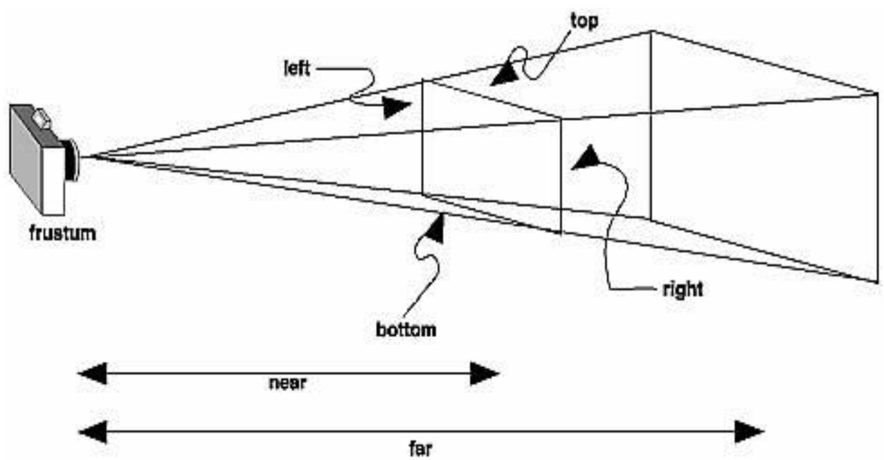
$$A = \frac{right+left}{right-left}$$

$$B = \frac{top+bottom}{top-bottom}$$

$$C = -\frac{far+near}{far-near}$$

$$D = -\frac{2\,far\,near}{far-near}$$

Определяет пирамиду видимости, расположенную вдоль оси Z с вершиной в точке (0, 0, 0), используется левосторонняя система координат.



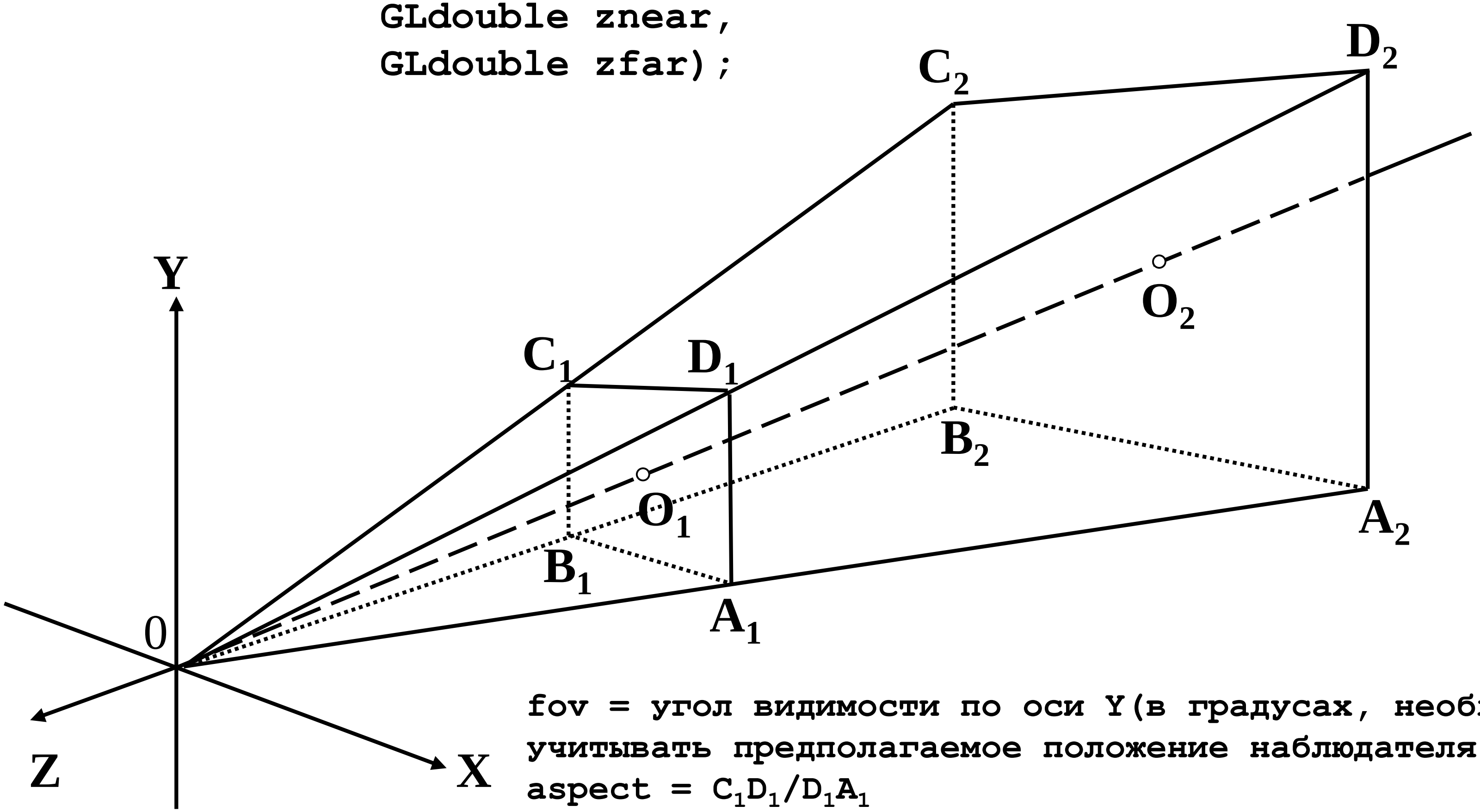
$$r = \frac{far}{near}$$

В силу операции перспективного деления, данное соотношение определяет точность операций с глубиной.

Перспективное проективное преобразование в OpenGL

```
void gluPerspective(GLdouble fov,
                    GLdouble aspect,
                    GLdouble znear,
                    GLdouble zfar);
```

фактически используется левосторонняя СК



fov = угол видимости по оси Y(в градусах, необходимо учитывать предполагаемое положение наблюдателя)

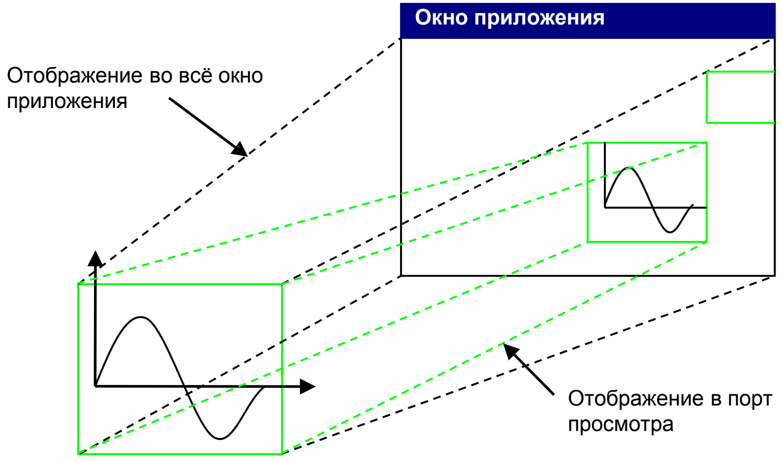
aspect = C_1D_1/D_1A_1

znear = $|OO_1|$

zfar = $|OO_2|$

Преобразование рабочей станции и порт просмотра

Мировое окно – та часть сцены, которую необходимо отобразить
Порт просмотра – область, в которую проецируется мировое окно



Усеченные координаты:
$(xc, yc, zc, wc)^T$

Нормализованные координаты (перспективное деление):
$(xn, yn, zn)^T = (xc/wc\ ,\ yc/wc,\ zc/wc)^T$

Оконные координаты (нормализованные координаты приведены к диапазону [-1;1]):
$(xw, yw, zw)^T = (\ (px/2)\ xn + ox\ ,\ (py/2)\ yn + oy\ ,\ [(f-n)/2]\ zn+(n+f)/2\ )^T$, где

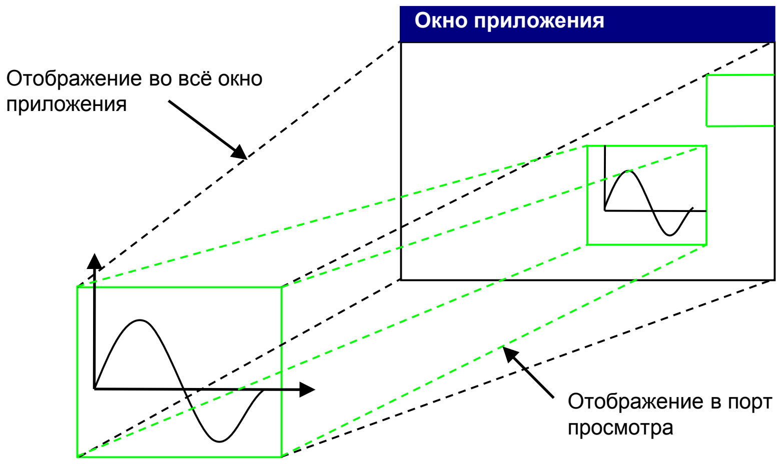
$px=width$, $py=height$

$ox=x+width/2$, $oy=y+height/2$ (оконные координаты центра области вывода)

n и f задают минимальную и максимальную глубину точки в окне

Преобразование порта просмотра в OpenGL

$$V = \begin{pmatrix} \frac{w}{2} & 0 & 0 & x_{left} + \frac{w}{2} \\ 0 & \frac{h}{2} & 0 & y_{bottom} + \frac{h}{2} \\ 0 & 0 & \frac{d}{2} & z_{near} + \frac{d}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



Задание порта просмотра:

glViewPort (GLint *x*, GLint *y*, GLint *width*, GLint *height*)

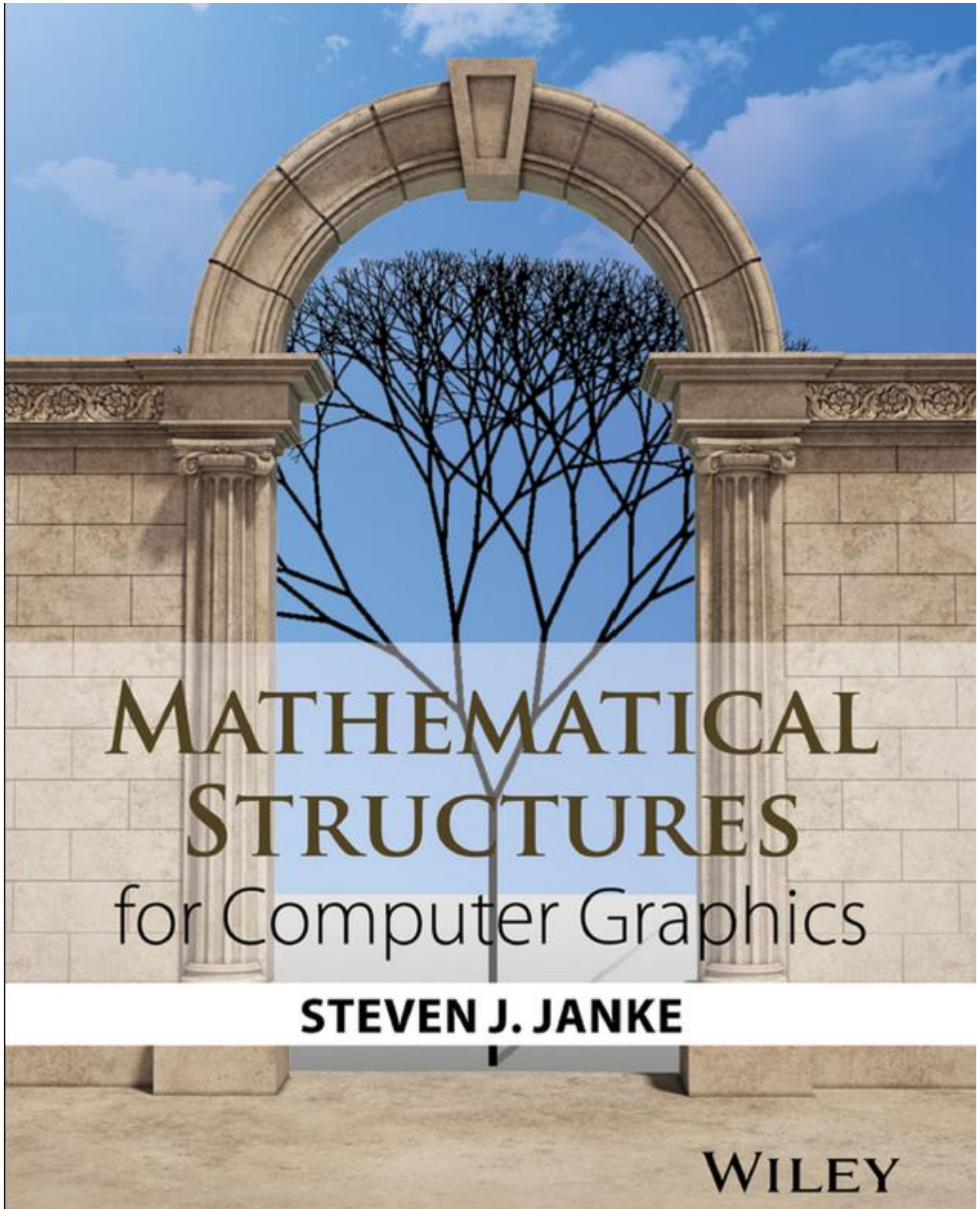
Определение значений n и f (по умолчанию равны 0 и 1 соответственно):

glDepthRange (GLclampd *n*, GLclampd *f*)

Получение образа и прообраза точки в OpenGL

```
int gluProject(  GLdouble objx, GLdouble objy, GLdouble objz,
                 const GLdouble modelMatrix[16],
                 const GLdouble projMatrix[16],
                 const GLint viewport[4],
                 GLdouble *winx, GLdouble *winy, GLdouble *winz );
```

```
int gluUnProject( GLdouble winx, GLdouble winy, GLdouble winz,
                  const GLdouble modelMatrix[16],
                  const GLdouble projMatrix[16],
                  const GLint viewport[4],
                  GLdouble *objx, GLdouble *objy, GLdouble *objz );
```


MATHEMATICAL
STRUCTURES
for Computer Graphics
STEVEN J. JANKE
WILEY

Лабораторная работа № 2 / 3

1. Определить параметризованную модель объекта сцены (в соответствии с вариантом). Для лабораторной работы №2 в качестве объекта использовать куб.
2. Определить преобразования, позволяющие получить заданный вид проекции (в соответствии с вариантом). Для демонстрации проекции добавить в сцену куб (в стандартной ориентации, не изменяемой при модельно-видовых преобразованиях основного объекта).
3. Реализовать изменение ориентации и размеров объекта (навигацию камеры) с помощью модельно-видовых преобразований (без gluLookAt). Управление производится интерактивно с помощью клавиатуры и/или мыши.
4. Предусмотреть возможность переключения между каркасным и твердотельным отображением модели (glFrontFace / glPolygonMode).

Вопросы к экзамену

- Линейные преобразования. Матричное представление преобразований. Однородные координаты. Аффинные преобразования и их свойства. Комбинирование и инвертирование преобразований. Преобразование нормали.
- Графический конвейер и преобразование координат. Модельные преобразования. Видовые преобразования. Виртуальная камера. Проективные преобразования.
- Двумерные преобразования и их матричное представление. Однородные координаты.
- Базовые трехмерные преобразования и их матричное представление. Однородные координаты.
- Отражение точки (вектора) от произвольной плоскости.
- Трёхмерный поворот относительно произвольной оси. Ориентация объекта.
- Классификация проекций. Виды параллельных проекций. Матричное представление ортографической проекции, аксонометрических косоугольных проекций.
- Классификация проекций. Виды параллельных проекций. Матричное представление аксонометрических прямоугольных проекций.
- Классификация проекций. Виды перспективных проекций. Матричное представление проективного преобразования.
- Нормализованные координаты и преобразование рабочей станции.
- Графический конвейер OpenGL и геометрические преобразования.