Инженерная и компьютерная графика
6 семестр (диф.зачет)

Лектор:

Таранцев Игорь Геннадьевич

Доцент ФИТ НГУ, ИАиЭ, «СофтЛаб-НСК»

Создатели курса:

Дебелов Виктор Алексеевич

Валеев Тагир Фаридович

Козлов Дмитрий Сергеевич

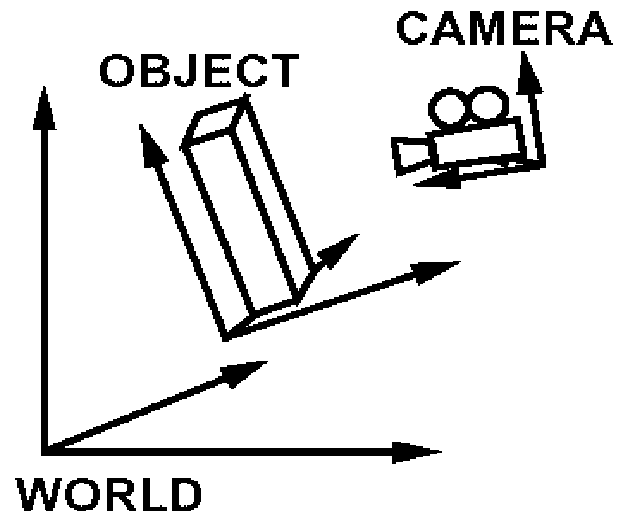
Лекция №8

Преобразования

Использовались материалы курса 15-462 (Heckbert)

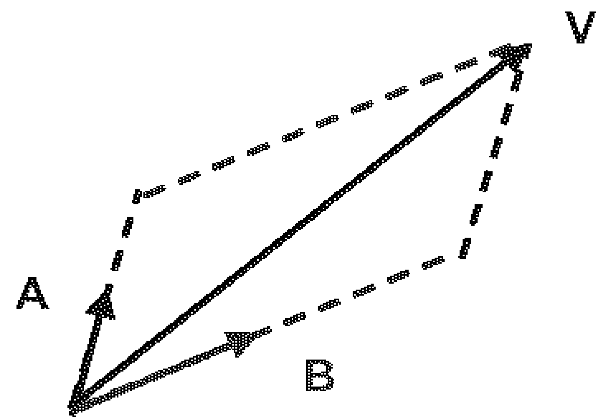
Преобразования

- Модельные
  - Создание сложных моделей из простых компонент путем позиционирования
  - Преобразование из объектных координат в мировые
- Видовые
  - Размещение виртуальной камеры в мире, т.е. Спецификация преобразования мировых координат в координаты камеры
- Анимации
  - Варьирование координат во времени для создания движений



Векторная алгебра

- Линейная комбинация векторов
- Линейная независимость векторов
- Базис
- Ортонормированный базис
- Компоненты = координаты вектора
- Вектор в базисе представляется однозначно
- При смене базиса вектор не меняется, меняются его компоненты



$$V = v_1 A + v_2 B;\ \ v_1, v_2 \in \Re$$

$$V = v_1 E_1 + v_2 E_2 + \ldots + v_n E_n$$

$$V = \begin{pmatrix} v_1 \\ v_2 \\ \ldots \\ v_n \end{pmatrix}$$

Линейные и аффинные преобразования

- F линейное, если $F(A+B)=F(A)+F(B)$ и $F(kA)=kF(A)$
- Любое линейное преобразование полностью специфицируется его действием на базисные векторы

$$
\begin{aligned}
V &= v_1E_1 + v_2E_2 + v_3E_3 \\
F(V) &= F(v_1E_1 + v_2E_2 + v_3E_3) \\
&= F(v_1E_1) + F(v_2E_2) + F(v_3E_3) \\
&= v_1F(E_1) + v_2F(E_2) + v_3F(E_3)
\end{aligned}
$$

- F аффинное, если оно линейное + сдвиг $\Rightarrow y = mx + b$

Линейное преобразование векторов

Преобразование базисных векторов

$$F(E_1) = f_{11}E_1 + f_{21}E_2 + f_{31}E_3$$
$$F(E_2) = f_{12}E_1 + f_{22}E_2 + f_{32}E_3$$
$$F(E_3) = f_{13}E_1 + f_{23}E_2 + f_{33}E_3$$

Преобразование вектора

$$\begin{aligned} F(V) &= v_1 F(E_1) + v_2 F(E_2) + v_3 F(E_3) \\ &= (f_{11}E_1 + f_{21}E_2 + f_{31}E_3)v_1 + (f_{12}E_1 + f_{22}E_2 + f_{32}E_3)v_2 + (f_{13}E_1 + f_{23}E_2 + f_{33}E_3)v_3 \\ &= (f_{11}v_1 + f_{12}v_2 + f_{13}v_3)E_1 + (f_{21}v_1 + f_{22}v_2 + f_{23}v_3)E_2 + (f_{31}v_1 + f_{32}v_2 + f_{33}v_3)E_3 \end{aligned}$$

Или в другой записи

$$\hat{v}_1 = f_{11}v_1 + f_{12}v_2 + f_{13}v_3$$
$$\hat{v}_2 = f_{21}v_1 + f_{22}v_2 + f_{23}v_3$$
$$\hat{v}_3 = f_{31}v_1 + f_{32}v_2 + f_{33}v_3$$

Получили формулу произведения матриц $\hat{v}_i = \sum_j f_{ij} v_j$

Матрицы для надежности

- Матрица $n \times n$ представляет линейную функцию в n-мерном пространстве
  - i-й столбец показывает как она действует на соответствующий вектор базиса
- Преобразование = линейная комбинация столбцов матрицы
- Обычно вычисляют по другому: скалярно умножают строку i на исходный вектор, чтобы получить компоненту i выходного вектора

$$\begin{pmatrix} v_1' \\ v_2' \\ v_3' \end{pmatrix} = \begin{pmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}$$

Основные 2D преобразования

Преобразование	Формула		Управляющий параметр
Сдвиг Translate	$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$	$v' = v + t$	t
Масштабирование Scale	$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$	$v' = Sv$	S
Поворот Rotate	$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$	$v' = R(\theta)v$	θ

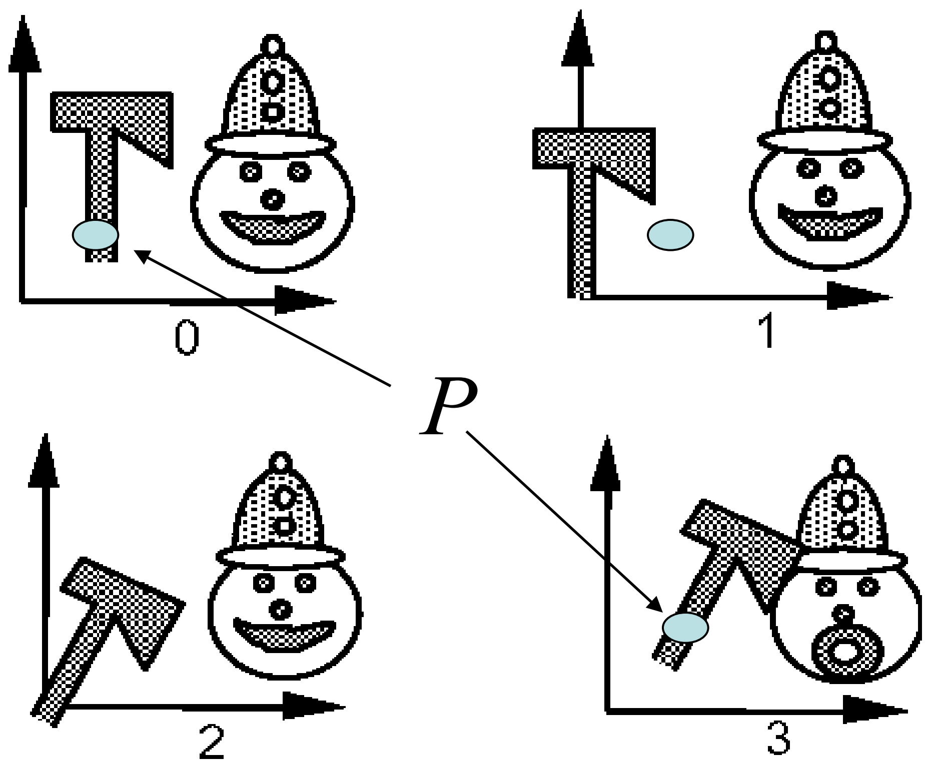
Составные преобразования

- Применять последовательно основные преобразования.
- Последовательность линейных (поворот и масштабирование) можно представить одним – свернуть их матрицы в одну результирующую матрицу.
- Наличие сдвига не дает применять матрицы, т.к. нелинейное преобразование.

$$v' = Sv \qquad v' = R(\theta)v \qquad v' = v + t$$

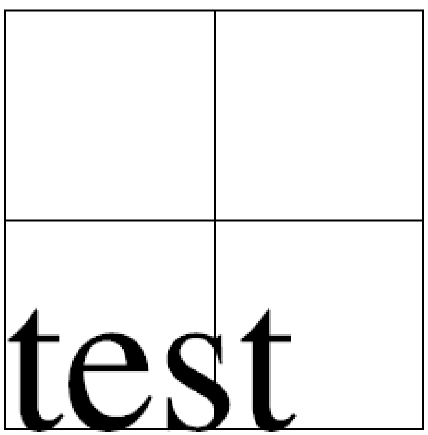
Поворот вокруг точки

1) Переносим топор в начало координат;
2) Поворачиваем на нужный угол;
3) Переносим в повернутом виде обратно.

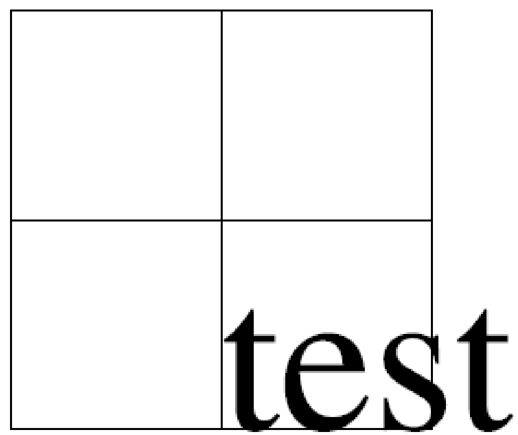


PostScript (постфиксная запись)

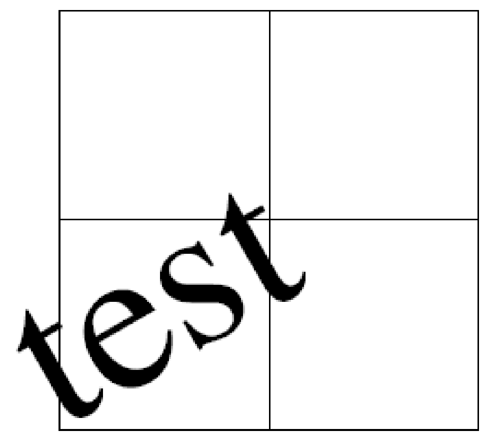
```
0 0 moveto
(test) show
```



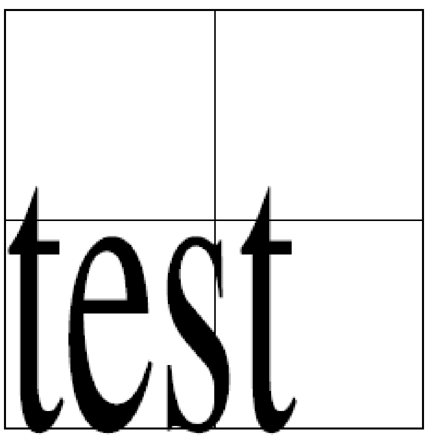

```
1 0 translate
0 0 moveto
(test) show
```



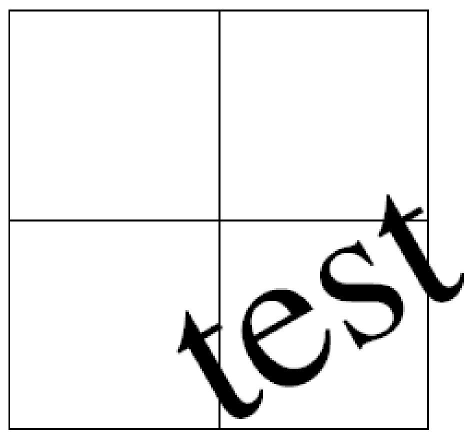

```
30 rotate
0 0 moveto
(test) show
```



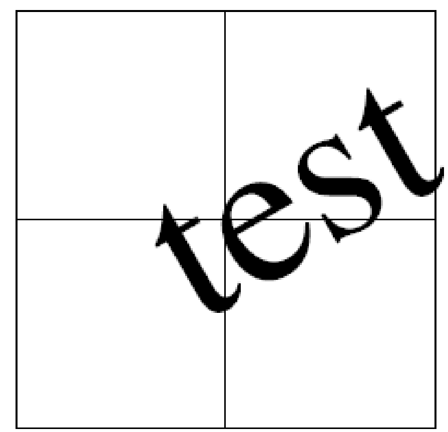
```
1 2 scale
0 0 moveto
(test) show
```



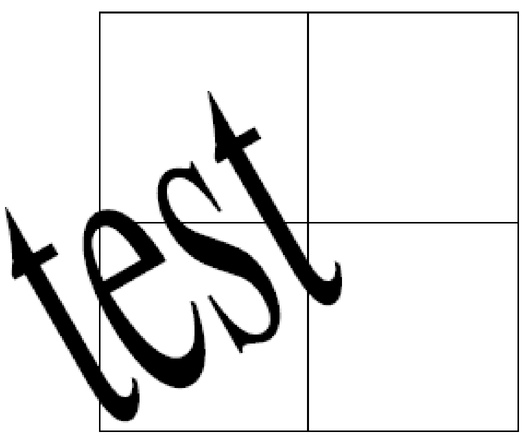

```
1 0 translate
30 rotate
0 0 moveto
(test) show
```



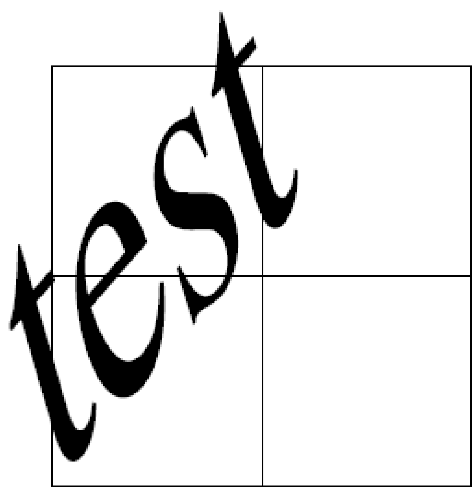

```
30 rotate
1 0 translate
0 0 moveto
(test) show
```



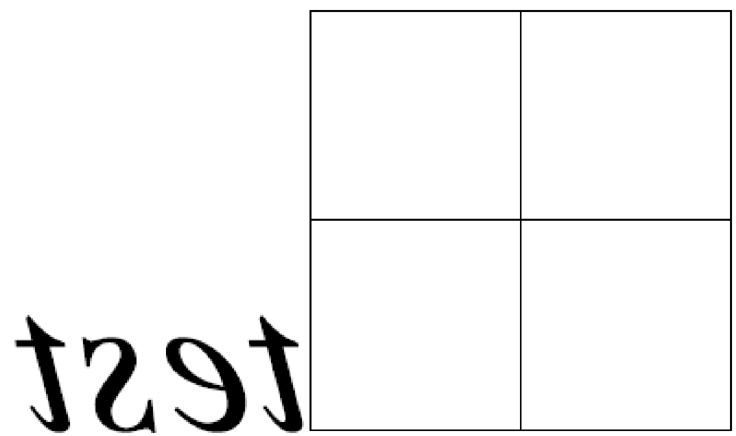
```
30 rotate
1 2 scale
0 0 moveto
(test) show
```




```
1 2 scale
30 rotate
0 0 moveto
(test) show
```




```
-1 1 scale
0 0 moveto
(test) show
```



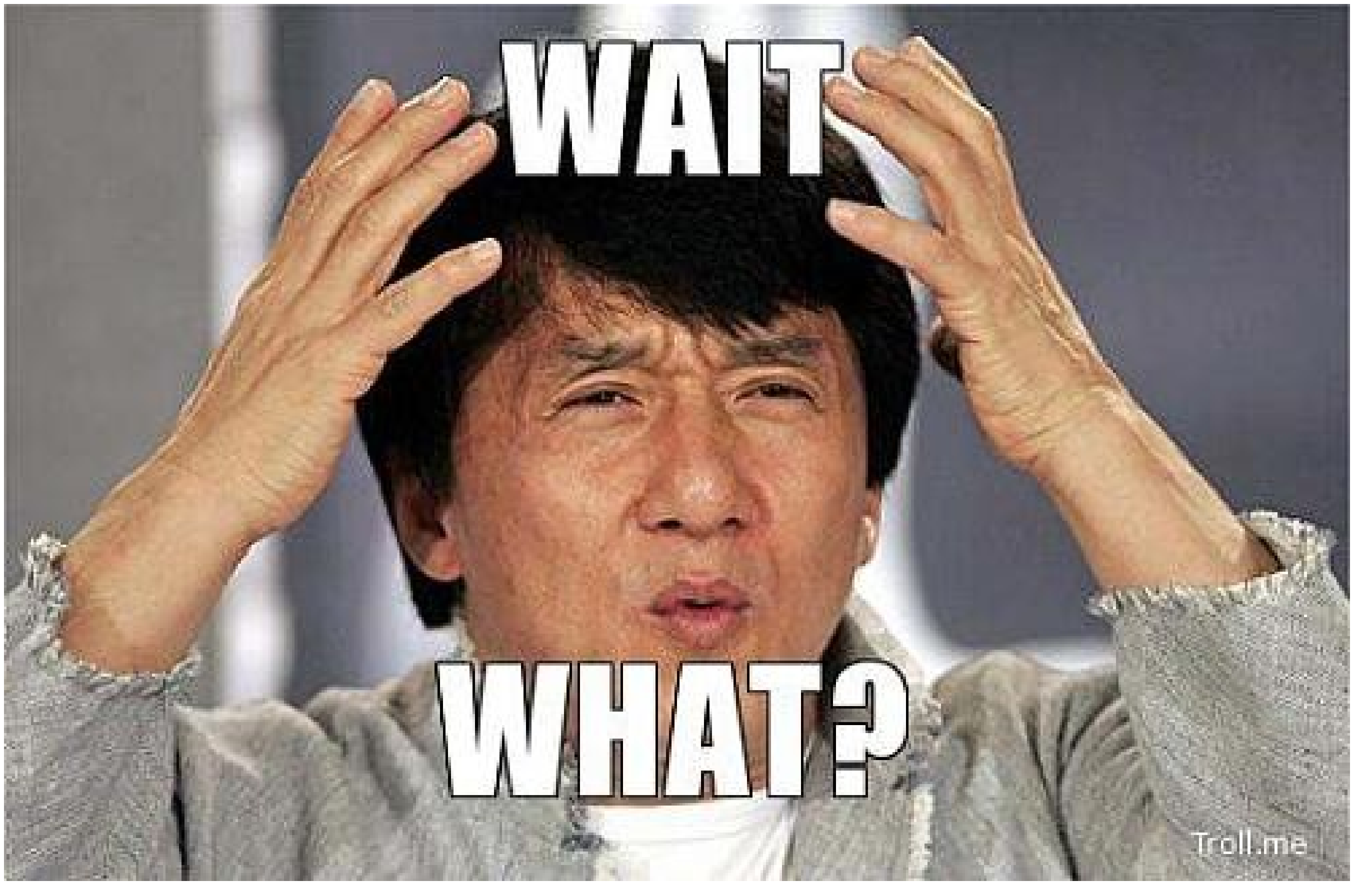
Однородные координаты

- Как бы представить сдвиг в матричном виде
- «трюк» – добавим по лишней координате к векторам

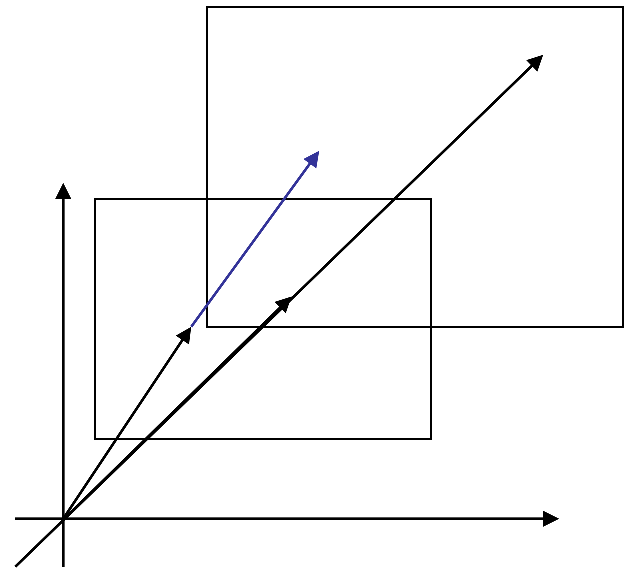
$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

- Это *однородная* координата w
- После преобразований отбрасываем её ($w = 1$) и всё
- Такие матрицы определяют *однородные* преобразования

W? WHAT?!



W? WHAT?!



- Практический ответ:
  - W – это алгебраический трюк
  - Сильно не переживай, w = 1.0 пока что
  - Если не равно 1.0, то просто дели на него
- Академический ответ:
  - (x, y, w) координаты из 3D проективного пространства
  - все ненулевые кратные w(x, y, 1) определяют одну и ту же точку в декартовых координатах (x, y)
  - w = 0 означает точку в бесконечности, что эквивалентно направлению
  - в 3D графике используются четырехмерные однородные координаты (x, y, z, w)

Грубо: z = w

Однородные 2D преобразования

Translate

$$\begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix}$$

Scale

$$\begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

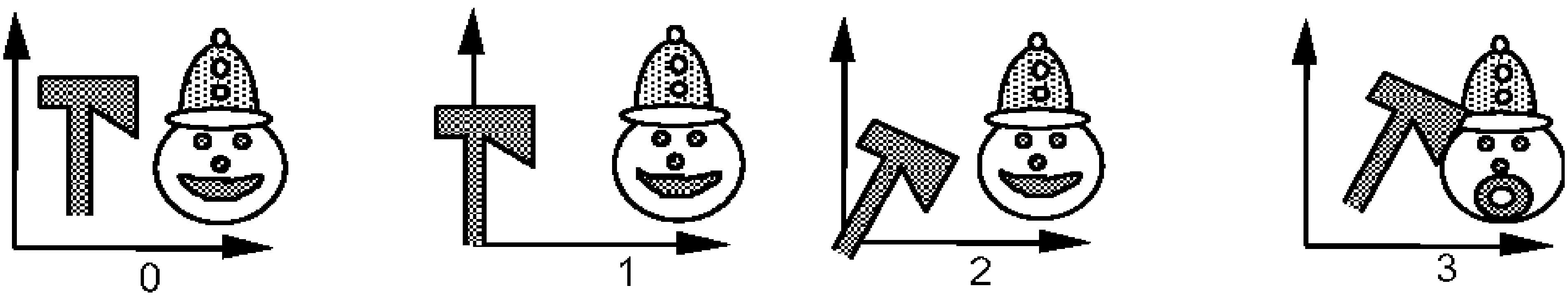
Rotate

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

Теперь любая последовательность сдвигов/вращений/масштабирований на плоскости может быть выражена одной матрицей

Поворот вокруг точки

1) Переносим топор в начало координат;
2) Поворачиваем на нужный угол;
3) Переносим в повернутом виде обратно.

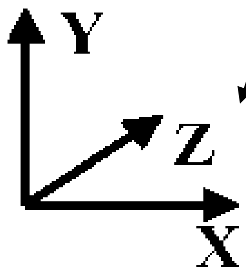


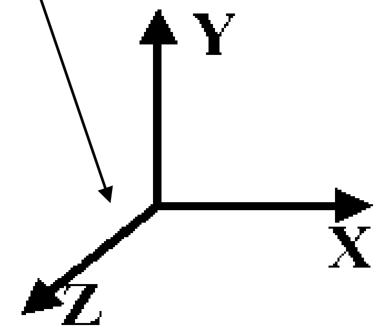
Translate(P) $\times$ Rotate(Alpha) $\times$ Translate($-$P) применим к каждой точке

Соглашения:

вектор – столбец	вектор – строка
$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$	$\begin{pmatrix} x' & y' & 1 \end{pmatrix} = \begin{pmatrix} x & y & 1 \end{pmatrix} \begin{pmatrix} m_{11} & m_{21} & m_{31} \\ m_{12} & m_{22} & m_{32} \\ m_{13} & m_{23} & m_{33} \end{pmatrix}$
$p' = ABCDp$	$p'^T = p^T D^T C^T B^T A^T$

Левосторонние и правосторонние системы координат



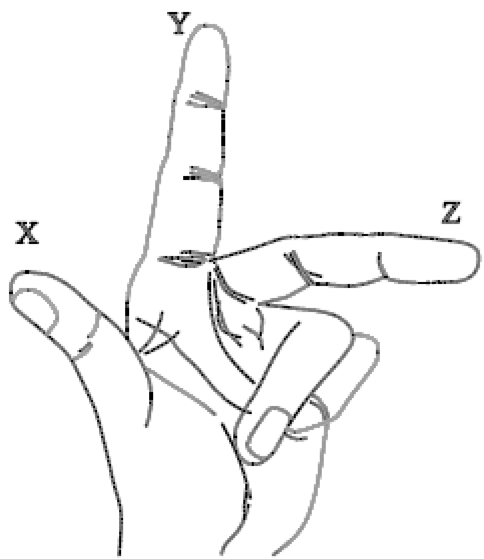


$$Z = X \times Y = \begin{pmatrix} x_2 y_3 - x_3 y_2 \\ x_3 y_1 - x_1 y_3 \\ x_1 y_2 - x_2 y_1 \end{pmatrix}$$

- Ось Z определяется через оси X и Y на основе векторного произведения, которое определяется правилом левой или правой руки (согласно выбранной системе)

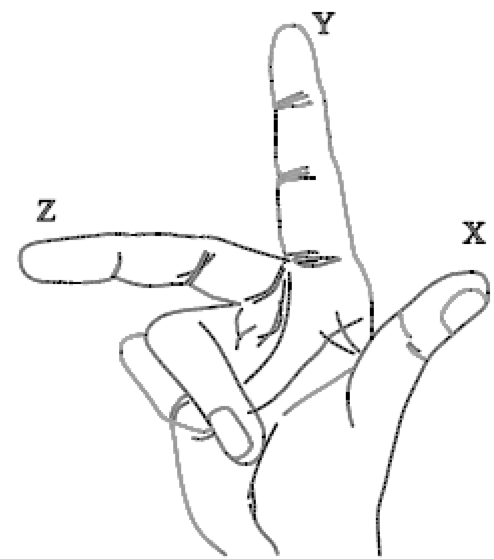
Левосторонние и правосторонние системы координат

Левая рука



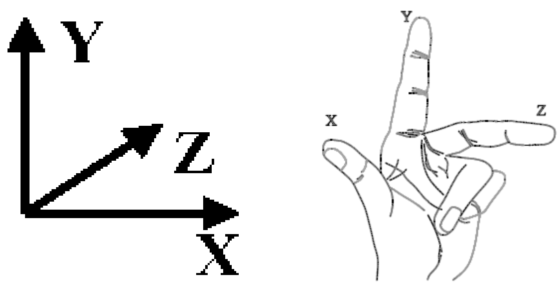
$$\vec{Y} \times \vec{X} = \vec{Z}$$

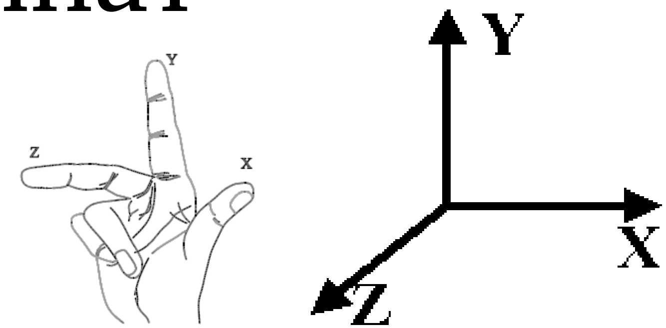
Правая рука



$$\vec{X} \times \vec{Y} = \vec{Z}$$

Левосторонние и правосторонние системы координат





- В отличие от плоскости в пространстве есть понятие оси вращения
- Направление вращения зависит от выбранной системы координат (левой или правой)
- Положительное вращение (верно для обеих систем):
  - Вокруг оси X: от +Y к +Z
  - Вокруг оси Y: от +Z к +X
  - Вокруг оси Z: от +X к +Y

Ломание головы

Однородные 3D преобразования

Translate

$$T(t_x,t_y,t_z)=\begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Scale

$$S(s_x,s_y,s_z)=\begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_y & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Любая последовательность сдвигов/вращений/масштабирований в пространстве может быть выражена одной матрицей

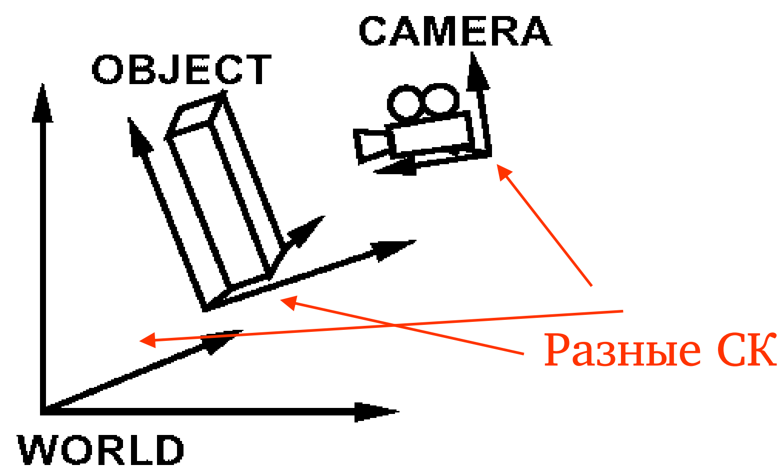
Однородные 3D преобразования

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

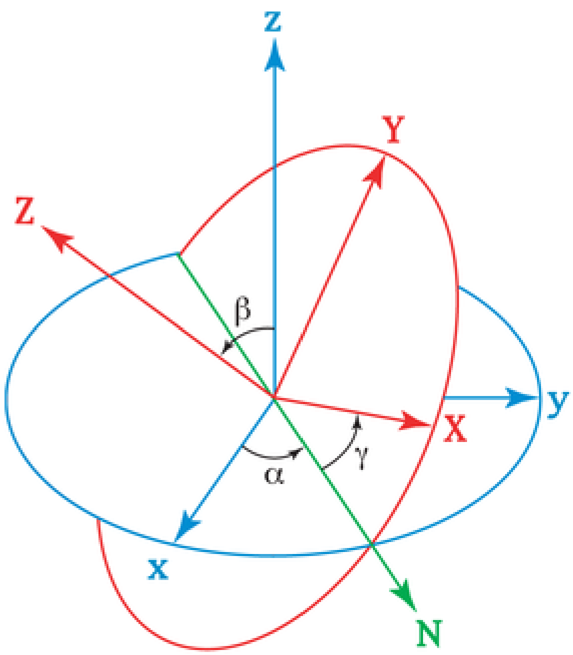
$$R_y(\theta) = \begin{pmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

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

- Углы Эйлера
- Любой поворот – это комбинация поворотов вокруг координатных осей
- Неоднозначность представления
- Анимации получаются дерганные
- Представлены матрицы вращений для правосторонней системы координат
- Для левосторонней СК их надо транспонировать
- Для отрезков, граней и т.п достаточно преобразовывать только опорные вершины



Углы Эйлера



Углы Эйлера определяют три поворота системы, которые позволяют привести любое положение системы к текущему. Обозначим начальную систему координат как (x, y, z), конечную как (X, Y, Z). Пересечение координатных плоскостей xy и XY называется линией узлов N.

Угол α между осью x и линией узлов.
Угол β между осями z и Z.
Угол γ между осью X и линией узлов.

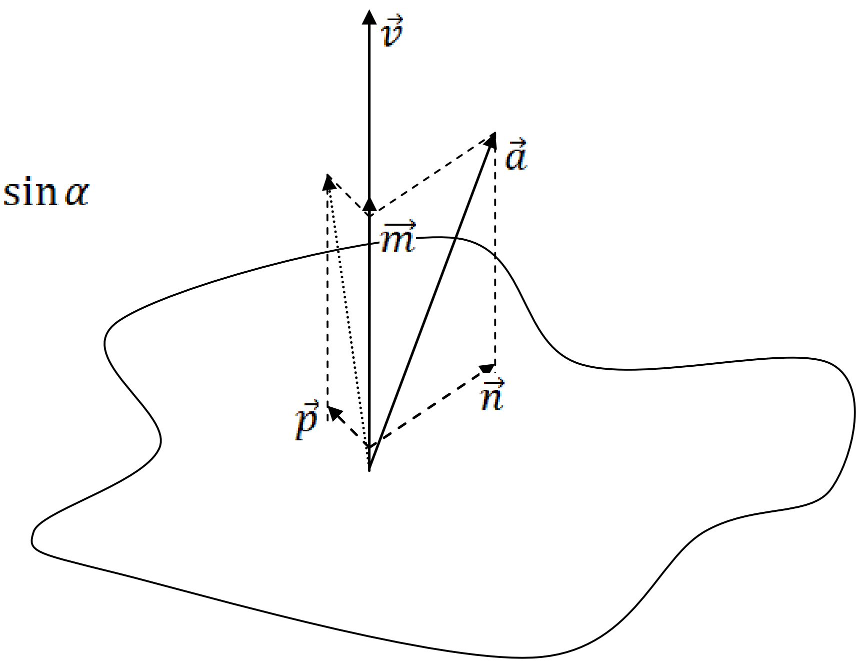
Повороты системы на эти углы называются прецессия, нутация и поворот на собственный угол (вращение). Такие повороты некоммутативны и конечное положение системы зависит от порядка, в котором совершаются повороты. В случае углов Эйлера это последовательность 3, 1, 3 (Z, X, Z).

Вращение вокруг оси

$\vec{a}$	– вектор, который поворачиваем
$\vec{v}$	– ось вращения
$\vec{m} = (\vec{a}, \vec{v}) \cdot \vec{v}$	– проекция на ось вращения
$\vec{n} = \vec{a} - (\vec{a}, \vec{v}) \cdot \vec{v}$	– проекция на плоскость, перпендикулярную оси вращения
$\vec{p} = \vec{v} \times \vec{a}$	– проекция вектора, повёрнутого на 90°

Поворот на произвольный угол α:

$$\overrightarrow{a_{\propto}} = \vec{m} + \vec{n} \cdot \cos\alpha + \vec{p} \cdot \sin\alpha =$$
$$(\vec{a}, \vec{v}) \cdot \vec{v} + (\vec{a} - (\vec{a}, \vec{v}) \cdot \vec{v}) \cdot \cos\alpha + \vec{v} \times \vec{a} \cdot \sin\alpha$$



Оператор вращения вокруг оси

$$\overrightarrow{a_\propto} = \vec{m} + \vec{n}\cos\alpha + \vec{p}\sin\alpha = (\vec{a},\vec{v})\vec{v} + (\vec{a} - (\vec{a},\vec{v})\vec{v})\cos\alpha + \vec{v}\times\vec{a}\sin\alpha$$

$$(\vec{a},\vec{v}) = a_x v_x + a_y v_y + a_z v_z$$

$$(\vec{a},\vec{v})\vec{v} = \begin{pmatrix} a_x v_x v_x + a_y v_y v_x + a_z v_z v_x \\ a_x v_x v_y + a_y v_y v_y + a_z v_z v_y \\ a_x v_x v_z + a_y v_y v_z + a_z v_z v_z \end{pmatrix} = \begin{pmatrix} v_x v_x & v_y v_x & v_z v_x \\ v_x v_y & v_y v_y & v_z v_y \\ v_x v_z & v_y v_z & v_z v_z \end{pmatrix}\begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix}$$

$$\begin{pmatrix} v_x v_x & v_y v_x & v_z v_x \\ v_x v_y & v_y v_y & v_z v_y \\ v_x v_z & v_y v_z & v_z v_z \end{pmatrix} = \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix}\begin{pmatrix} v_x & v_y & v_z \end{pmatrix} = vv^T$$

$$(\vec{a},\vec{v})\vec{v} = vv^T\begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix}$$

Оператор вращения вокруг оси

$$\overrightarrow{a_{\propto}} = \vec{m} + \vec{n}\cos\alpha + \vec{p}\sin\alpha = (\vec{a},\vec{v})\vec{v} + (\vec{a} - (\vec{a},\vec{v})\vec{v})\cos\alpha + \vec{v}\times\vec{a}\sin\alpha$$

$$(\vec{a} - (\vec{a},\vec{v})\vec{v}) = \left(\begin{pmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{pmatrix} - vv^T\right)\begin{pmatrix}a_x\\ a_y\\ a_z\end{pmatrix} = \left(I - vv^T\right)\begin{pmatrix}a_x\\ a_y\\ a_z\end{pmatrix}$$

$$I = \begin{pmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{pmatrix}$$

$$(\vec{a} - (\vec{a},\vec{v})\vec{v})\cos\alpha = \left(I - vv^T\right)\cos a\begin{pmatrix}a_x\\ a_y\\ a_z\end{pmatrix}$$

Оператор вращения вокруг оси

$$\overrightarrow{a_{\propto}} = \vec{m} + \vec{n}\cos\alpha + \vec{p}\sin\alpha = (\vec{a},\vec{v})\vec{v} + (\vec{a} - (\vec{a},\vec{v})\vec{v})\cos\alpha + \vec{v}\times\vec{a}\sin\alpha$$

$$\vec{v}\times\vec{a} = \begin{pmatrix} v_y a_z - v_z a_y \\ v_z a_x - v_x a_z \\ v_x a_y - v_y a_x \end{pmatrix} = \begin{pmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{pmatrix}\begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} = v^*\begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix}$$

$$v^* = \begin{pmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{pmatrix}$$

Векторное произведение представлено умножением на двойственную (dual) матрицу

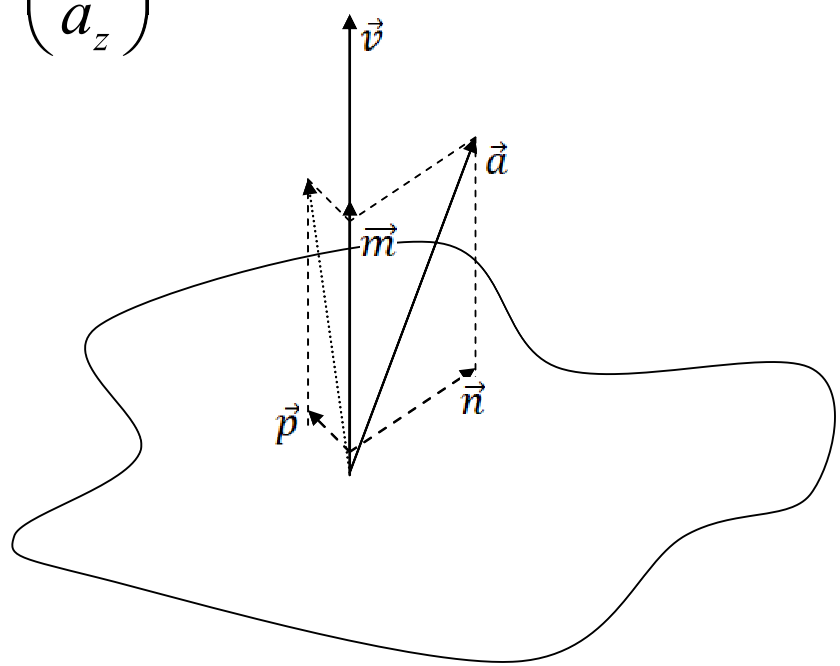
$$\vec{v}\times\vec{a}\sin\alpha = v^*\sin a\begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix}$$

Оператор вращения вокруг оси

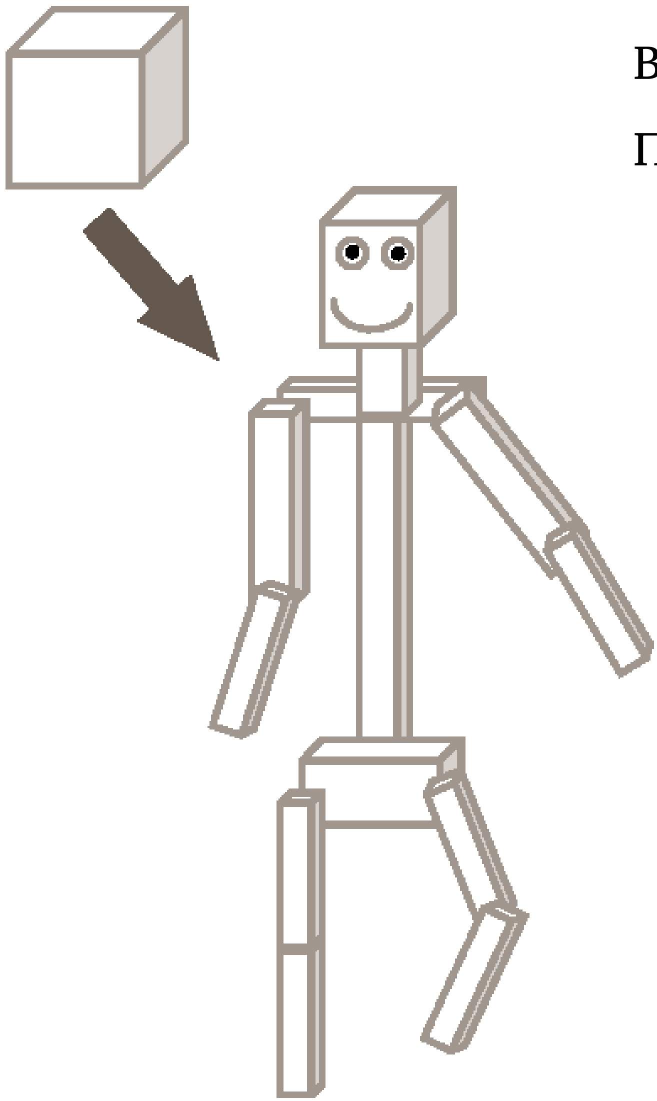
$$\overrightarrow{a_{\propto}} = \vec{m} + \vec{n}\cos\alpha + \vec{p}\sin\alpha = (\vec{a},\vec{v})\vec{v} + (\vec{a} - (\vec{a},\vec{v})\vec{v})\cos\alpha + \vec{v}\times\vec{a}\sin\alpha$$

$$\overline{a_{\propto}} = \left(vv^T + \left(I - vv^T\right)\cos a + v^*\sin a\right)\begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix}$$

$$\boxed{R = vv^T + \left(I - vv^T\right)\cos a + v^*\sin a}$$



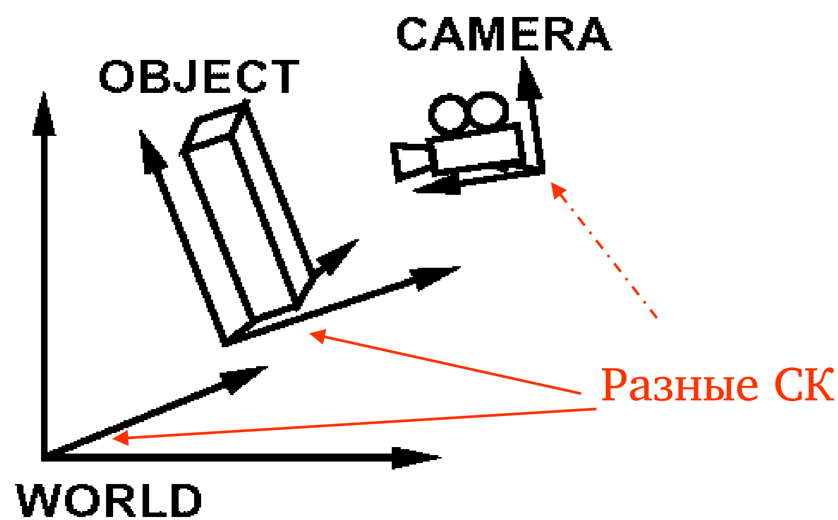
Модельные преобразования



Все из 1 базового кубика: повороты, масштабы, сдвиги

Прекрасно, пока не захотим анимировать – пусть ходит

Физически корректные деформации



Преобразование через базис

(after Peter Shirly)

$\mathbf{o}, \mathbf{x}, \mathbf{y}, \mathbf{z}$ – это наша неявная система координат

$$P = (3,4,2) \Rightarrow P = \mathbf{o} + \mathbf{3x} + \mathbf{4y} + \mathbf{2z}$$

$\mathbf{o}' = (o'_x, o'_y, o'_z)$ – новое начало системы координат

$$P = (x, y, z) \Rightarrow P = (x', y', z')$$

$$(x', y', z') = (x - o'_x, y - o'_y, z - o'_z)$$

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & -o'_x \\ 0 & 1 & 0 & -o'_y \\ 0 & 0 & 1 & -o'_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Преобразование через базис (преобразование камеры)

(after Peter Shirly)

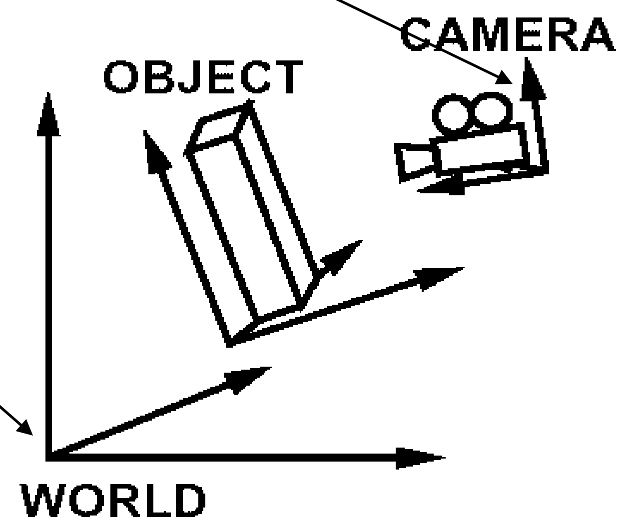
$P = (x, y, z) \Rightarrow P = (u, v, w)$ из мировой СК в СК камеры, например

$\mathbf{u} = (u_x, u_y, u_z)$ $\mathbf{v} = (v_x, v_y, v_z)$ $\mathbf{w} = (w_x, w_y, w_z)$

$$\mathbf{o} + x\mathbf{x} + y\mathbf{y} + z\mathbf{z} = \mathbf{o} + u\mathbf{u} + v\mathbf{v} + w\mathbf{w}$$

$$x = uu_x + vv_x + ww_x$$
$$y = uu_y + vv_y + ww_y$$
$$z = uu_z + vv_z + ww_z$$

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} u_x & v_x & w_x \\ u_y & v_y & w_y \\ u_z & v_z & w_z \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix}$$



Преобразование через базис (преобразование камеры)

(after Peter Shirly)

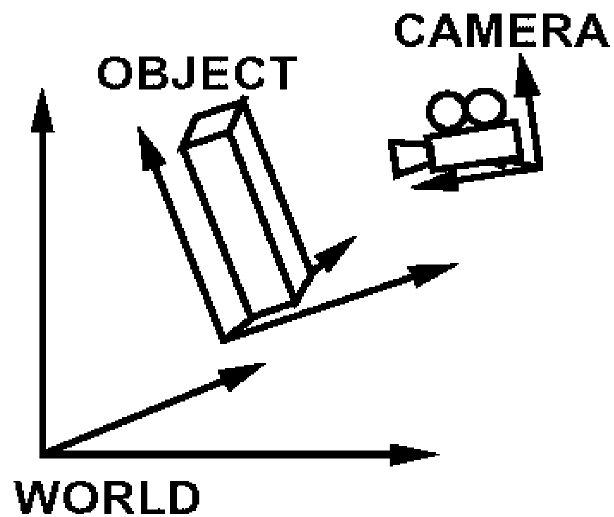
$$P = (x, y, z) \Rightarrow P = (u, v, w)$$

из мировой СК в СК камеры, например

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} u_x & v_x & w_x \\ u_y & v_y & w_y \\ u_z & v_z & w_z \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix}$$

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} u_x & u_y & u_z \\ v_x & v_y & v_z \\ w_x & w_y & w_z \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

$$\begin{pmatrix} u \\ v \\ w \\ 1 \end{pmatrix} = \begin{pmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ w_x & w_y & w_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$



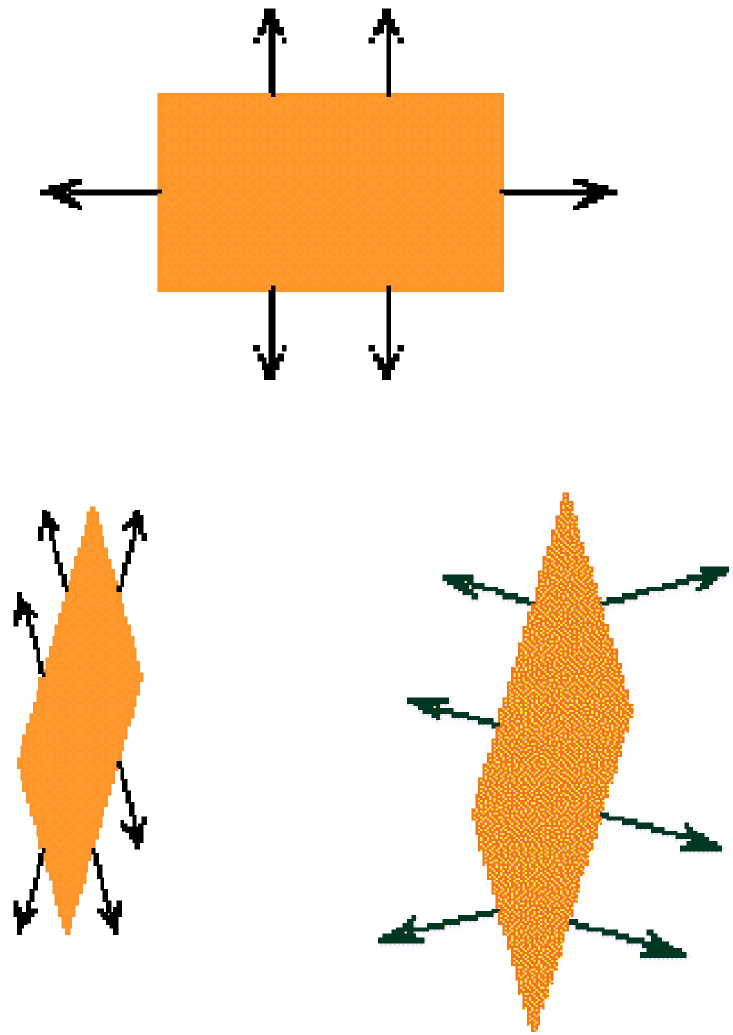
Rigid body transformations
преобразования твердого тела

$$\begin{pmatrix} m_{11} & m_{12} & m_{13} & t_x \\ m_{21} & m_{22} & m_{23} & t_y \\ m_{31} & m_{32} & m_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Преобразование твердого тела не содержит деформаций и является комбинацией любого числа поворотов и сдвигов. Матрица 3×3 – это матрица вращения (ортогональная матрица).

Преобразование нормалей

- Можно подумать, что нормали, как иголки ежа, почему бы их не обрабатывать как точки
- Но не так, рассмотрим пример в 2D
- Значит для нормалей нужно свое правило



Преобразование нормалей

- Еще раз! Не как точки!
- Преобразование $\mathbf{M}$
- Преобразование точки $p = (x \quad y \quad z)$

$$p' = \mathbf{M}p$$

- Нормаль $n = (n_x \quad n_y \quad n_z)$
- Нельзя $n' = \mathbf{M}n$
- Эта формула не работает для произвольного преобразования

Преобразование нормалей

$$ax + by + cz + d = 0$$

$$\mathbf{n} = \begin{pmatrix} a & b & c & d \end{pmatrix}^T$$

$$\mathbf{p} = \begin{pmatrix} x & y & z & 1 \end{pmatrix}^T$$

$$\mathbf{n} \cdot \mathbf{p} = \mathbf{n}^T \mathbf{p} = 0$$

$\begin{pmatrix} a & b & c \end{pmatrix}$ нормаль плоскости

d Расстояние от начала СК

Преобразование нормалей

$$0 = \mathbf{n}^T \mathbf{I} \mathbf{p} = \mathbf{n}^T (\mathbf{M}^{-1} \mathbf{M}) \mathbf{p} = (\mathbf{n}^T \mathbf{M}^{-1})(\mathbf{M} \mathbf{p}) = \mathbf{n}'^T \mathbf{p}'$$

$$\mathbf{p}' = \mathbf{M} \mathbf{p}$$

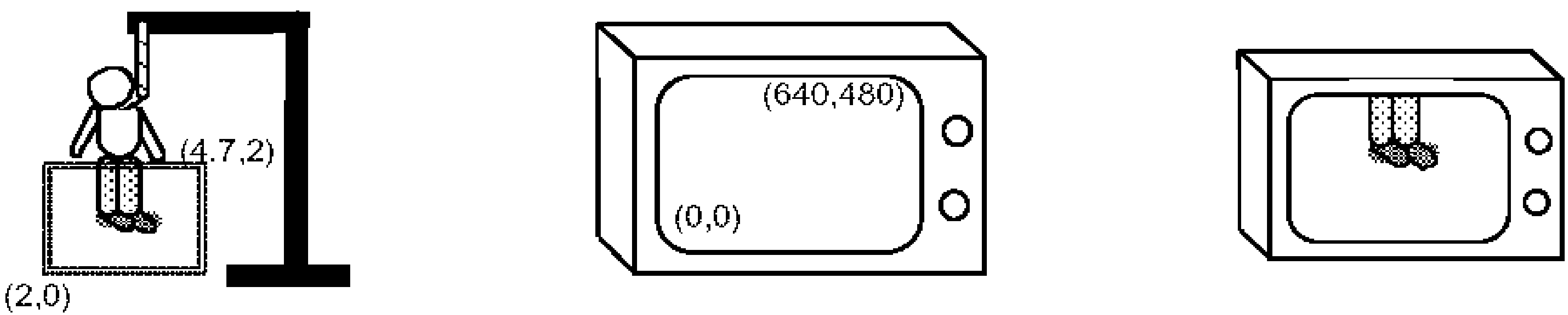
$$\mathbf{n}' = (\mathbf{n}^T \mathbf{M}^{-1})^T = (\mathbf{M}^{-1})^T \mathbf{n}$$

1. Для преобразований общего вида (например, включающих перспективу) надо использовать формулу с верным значением d
2. Для аффинных преобразований d не играет роли, можно положить равным 0
3. Для вращений (обратная и транспонированная матрица равна исходной матрице) можно преобразовывать как точки.

Преобразование порта вывода

Viewport transformation

- Определение видимой части мира в окне на экране
- Часто по умолчанию порт вывода – это все окно

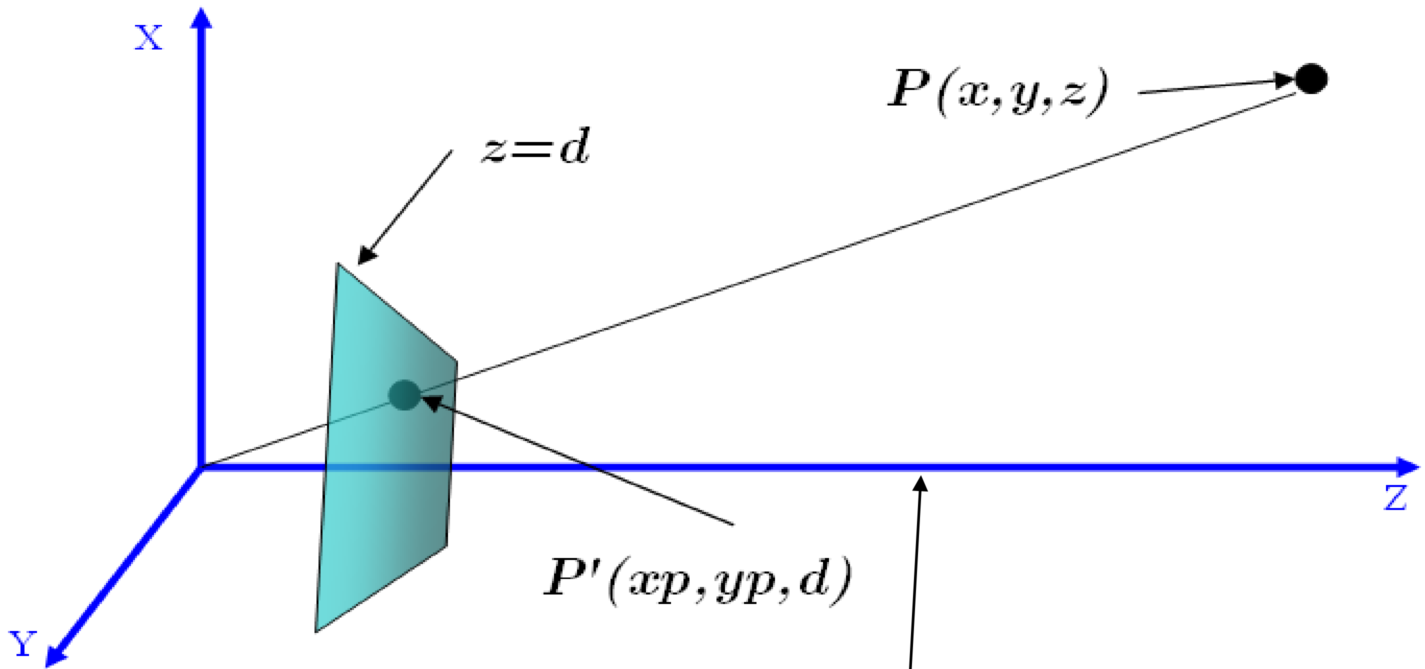


Видовое преобразование

Под видовым преобразованием понимается преобразование координат из мировой системы (сцены) в систему координат результирующего изображения – порта вывода. *Порт вывода (viewport)* – это прямоугольная область экрана или клиентской области окна. Основное преобразование – это перевод мировых координат в координаты *картинной плоскости*, которая, как правило, перпендикулярна оси Z. При этом:

- При параллельном проецировании – это плоскость $z = 0$
- При перспективном проецировании – плоскость $z = d > 0$

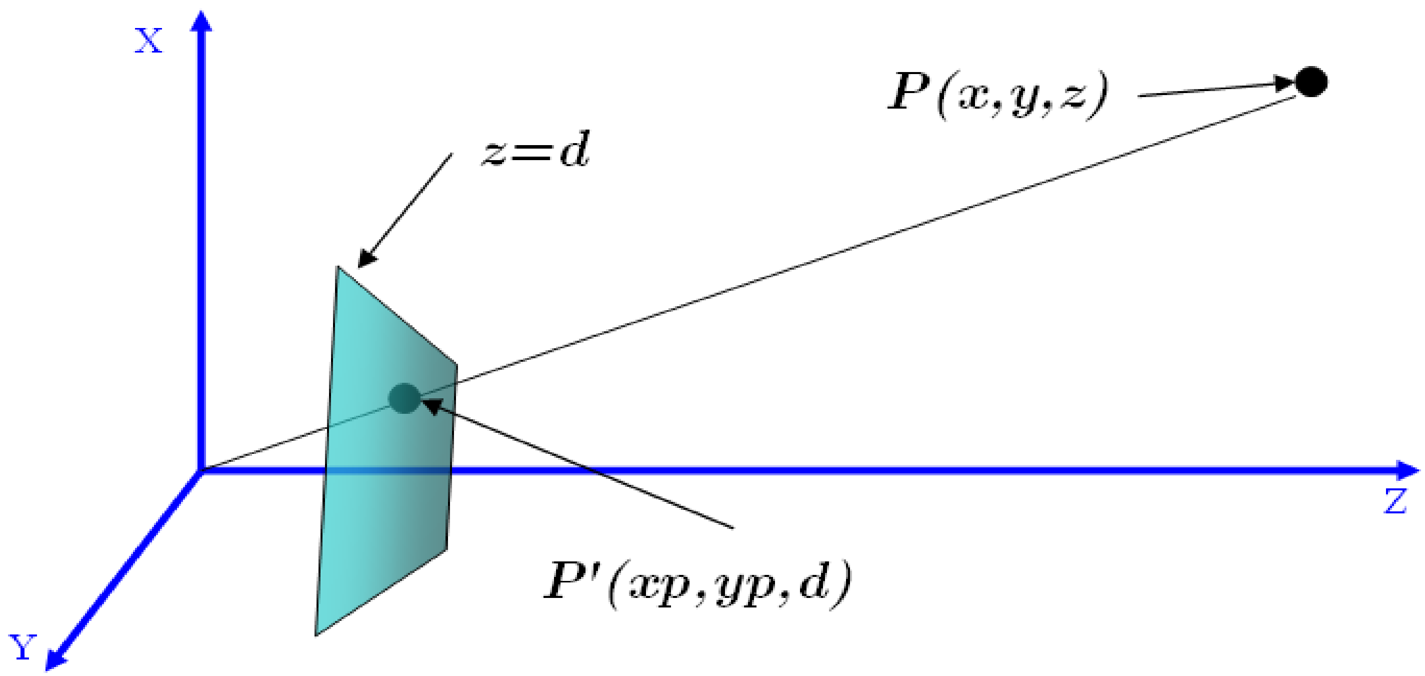
Видовое преобразование



Реально видовое преобразование раскладывается на несколько более простых, одно из которых – это преобразование камеры. Рассмотрим схему на рисунке. Здесь считается, что координаты объектов заданы в системе координат камеры. Сама камера находится в начале координат, картинная плоскость.

Направление вида (View direction) – ось Z.

Вывод матрицы проецирования



$$\frac{x_p}{d} = \frac{x}{z}; \frac{y_p}{d} = \frac{y}{z}; x_p = \frac{x}{z/d}; y_p = \frac{y}{z/d}; z \neq 0$$

Вывод матрицы проецирования

$$\frac{x_p}{d} = \frac{x}{z}; \frac{y_p}{d} = \frac{y}{z}; x_p = \frac{x}{z/d}; y_p = \frac{y}{z/d}; z \neq 0$$

$$M_{psp}(d) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{d} & 0 \end{pmatrix} \Rightarrow \begin{pmatrix} X_p \\ Y_p \\ Z_p \\ W_p \end{pmatrix} = M_{psp}(d) \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \\ \frac{z}{d} \end{pmatrix}$$

Делим на $W = \frac{z}{d}$ для перехода в R^3

Вывод матрицы проецирования

$$M_{psp}(d) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{d} & 0 \end{pmatrix} \Rightarrow \begin{pmatrix} X_p \\ Y_p \\ Z_p \\ W_p \end{pmatrix} = M_{psp}(d) \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \\ \frac{z}{d} \end{pmatrix}$$

$$\begin{pmatrix} X_p / W_p \\ Y_p / W_p \\ Z_p / W_p \\ W_p / W_p \end{pmatrix} = \begin{pmatrix} \frac{x}{z/d} \\ \frac{y}{z/d} \\ d \\ 1 \end{pmatrix} = \begin{pmatrix} x_p \\ y_p \\ d \\ 1 \end{pmatrix} = P'$$

Вывод матрицы проецирования

Если z = 0, а камера в (0, 0, −d)

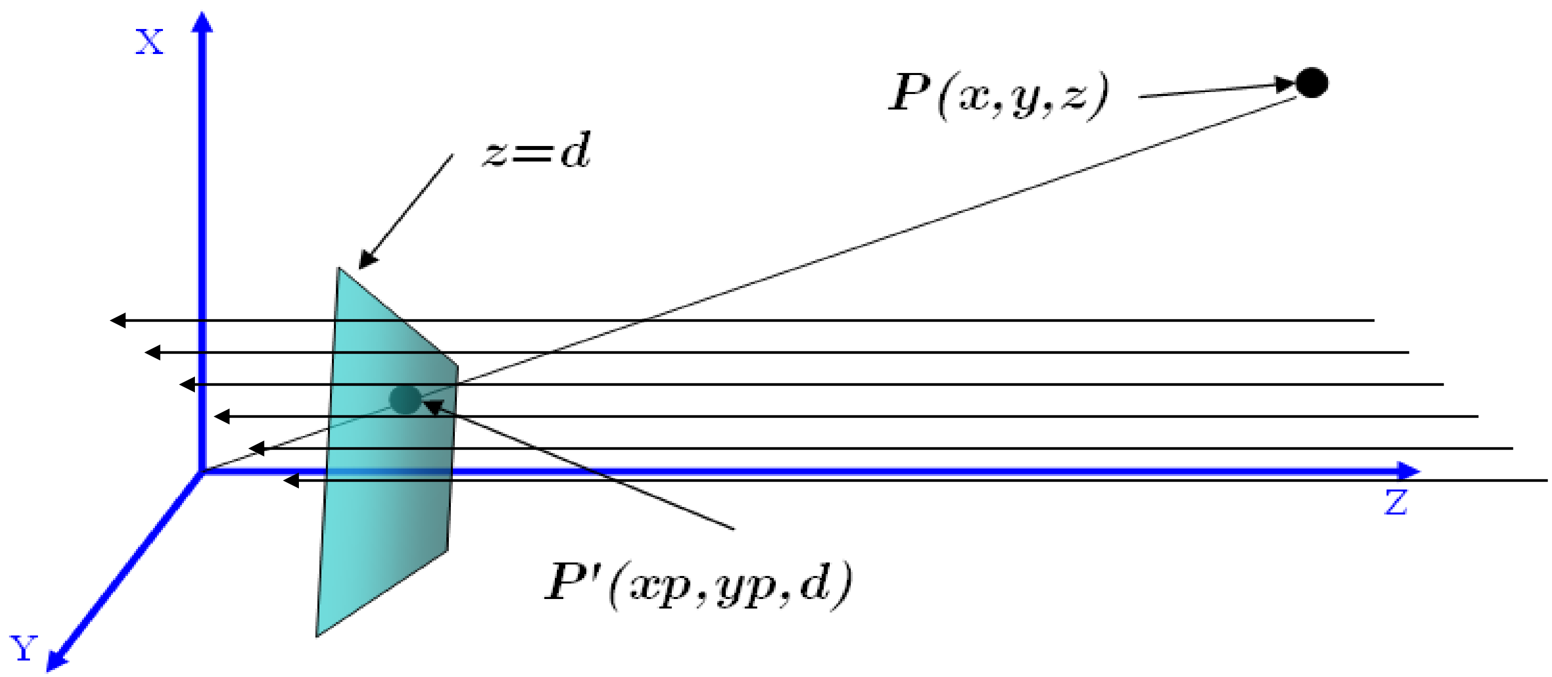
$$M_{psp}(d) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{d} & 0 \end{pmatrix} \Rightarrow M'_{psp}(d) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{d} & 1 \end{pmatrix}$$

$$M'_{psp} = T(0,0,-d)M_{psp}T(0,0,d)$$

Вывод матрицы проецирования

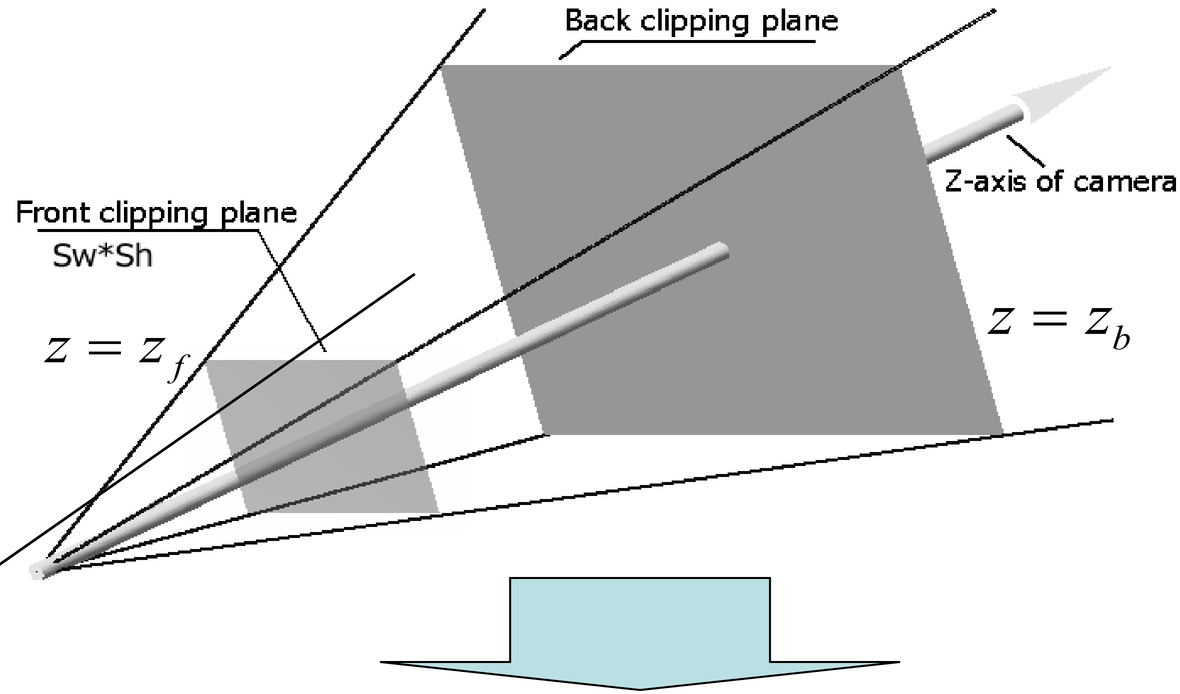
$$M_{ort} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Если параллельное проецирование



Преобразование к полукубу

$$M_{psp}(d) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{d} & 0 \end{pmatrix}$$



$$M_{psp}(z_f)$$

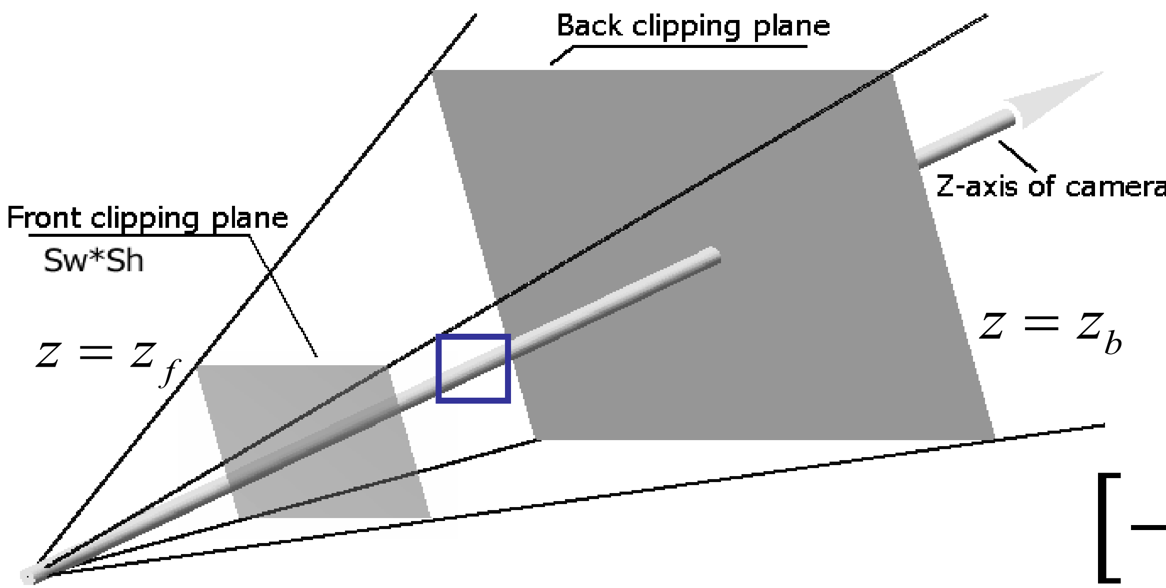
$$\left[-\frac{S_w}{2}, \frac{S_w}{2}\right] \times \left[-\frac{S_h}{2}, \frac{S_h}{2}\right] \times \left[z_f, z_f\right]$$

$$[-1,1] \times [-1,1] \times [0,1]$$

Преобразование к полукубу

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \to \begin{pmatrix} \frac{2}{S_w} x \cdot \frac{z_f}{z} \\ \frac{2}{S_h} y \cdot \frac{z_f}{z} \\ ? \\ \\ \end{pmatrix}$$

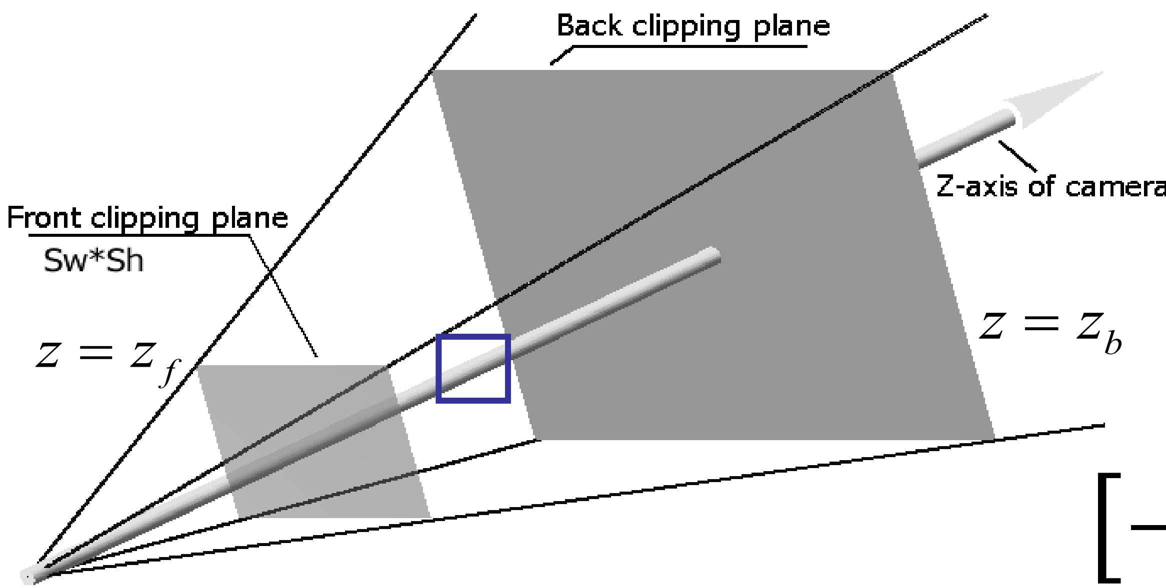
$$M_{proj}(S_w, S_h, z_f, z_b) \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{2}{S_w} x \cdot z_f \\ \frac{2}{S_h} y \cdot z_f \\ f\left(z, z_f, z_b\right) \\ z \end{pmatrix}$$



$$[-1,1] \times [-1,1] \times [0,1]$$

Преобразование к полукубу

$$M_{proj}(S_w, S_h, z_f, z_b) = \begin{pmatrix} \frac{2}{S_w} z_f & 0 & 0 & 0 \\ 0 & \frac{2}{S_h} z_f & 0 & 0 \\ 0 & 0 & a & b \\ 0 & 0 & 1 & 0 \end{pmatrix} \qquad M_{proj} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{2}{S_w} x \cdot z_f \\ \frac{2}{S_h} y \cdot z_f \\ az + b \\ z \end{pmatrix}$$

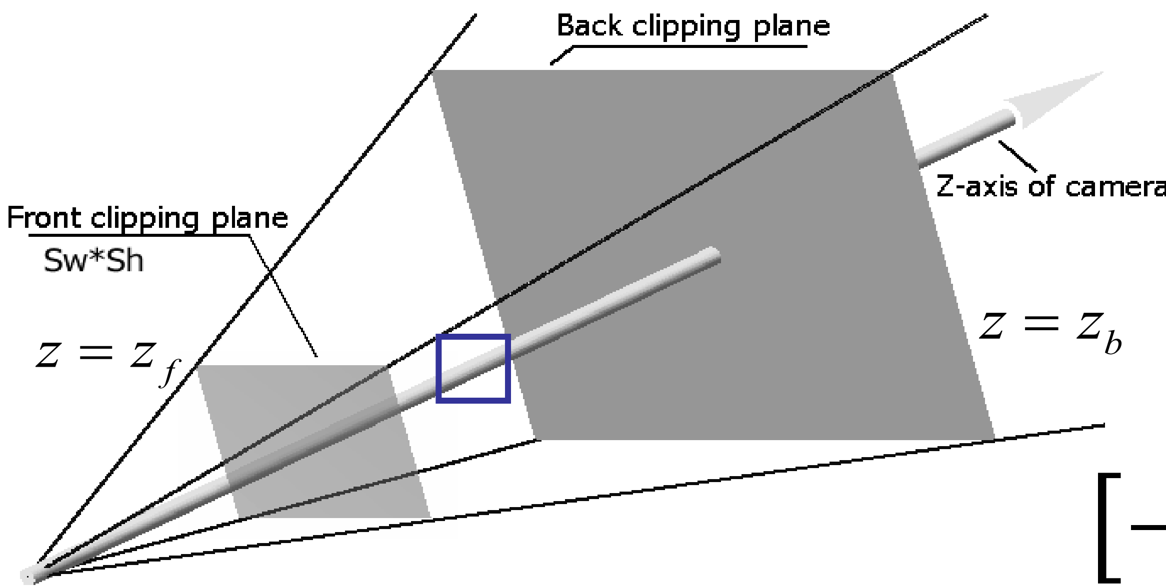


$$[-1,1] \times [-1,1] \times [0,1]$$

Преобразование к полукубу

$$z \to \frac{az+b}{z}$$

$$\begin{cases} z_f \to 0 \\ z_b \to 1 \end{cases} \Rightarrow \begin{cases} \dfrac{az_f+b}{z_f} = 0 \\ \dfrac{az_b+b}{z_b} = 1 \end{cases} \Rightarrow \begin{cases} b = -az_f \\ az_b - az_f = z_b \end{cases} \Rightarrow \begin{cases} a = \dfrac{z_b}{z_b - z_f} \\ b = \dfrac{-z_f z_b}{z_b - z_f} \end{cases}$$



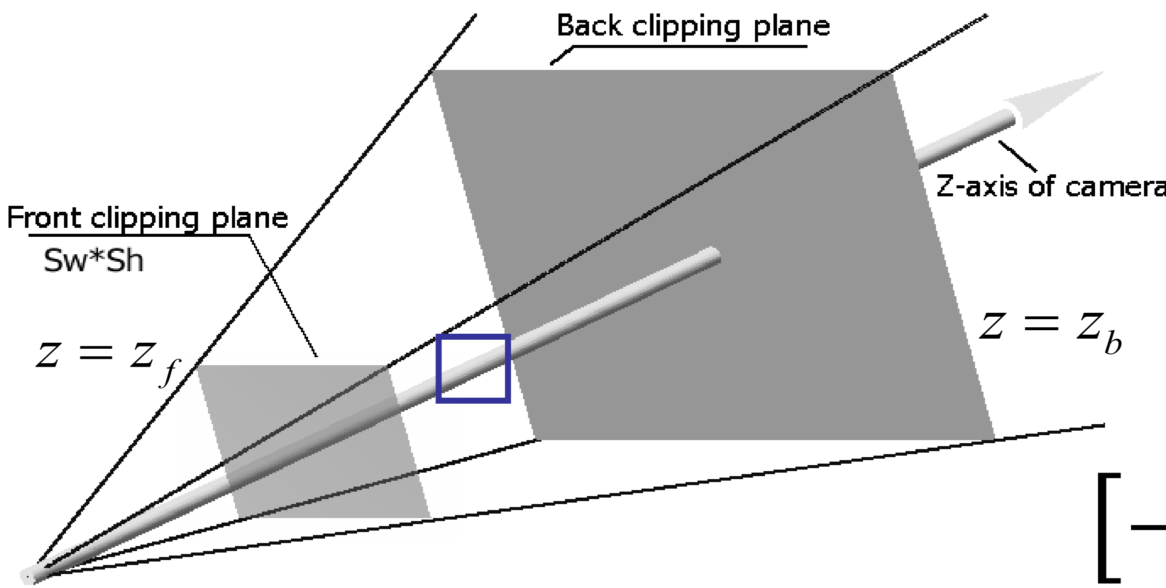
$$M_{proj} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} \dfrac{2}{S_w} x \cdot z_f \\ \dfrac{2}{S_h} y \cdot z_f \\ az+b \\ z \end{pmatrix}$$

$z = z_f$ $\quad z = z_b$

$$[-1,1] \times [-1,1] \times [0,1]$$

Преобразование к полукубу

$$M_{proj}(S_w, S_h, z_f, z_b) = \begin{pmatrix} \frac{2}{S_w} z_f & 0 & 0 & 0 \\ 0 & \frac{2}{S_h} z_f & 0 & 0 \\ 0 & 0 & \frac{z_b}{z_b - z_f} & \frac{-z_f z_b}{z_b - z_f} \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

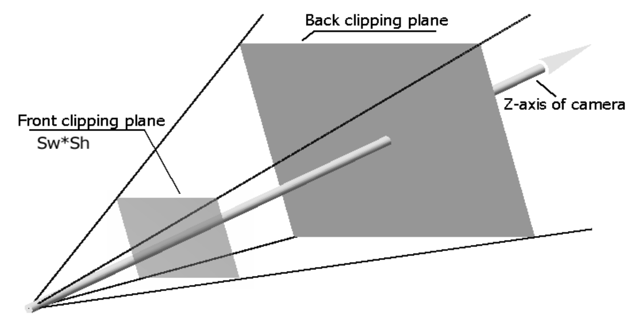


$$[-1,1] \times [-1,1] \times [0,1]$$

Преобразование к полукубу

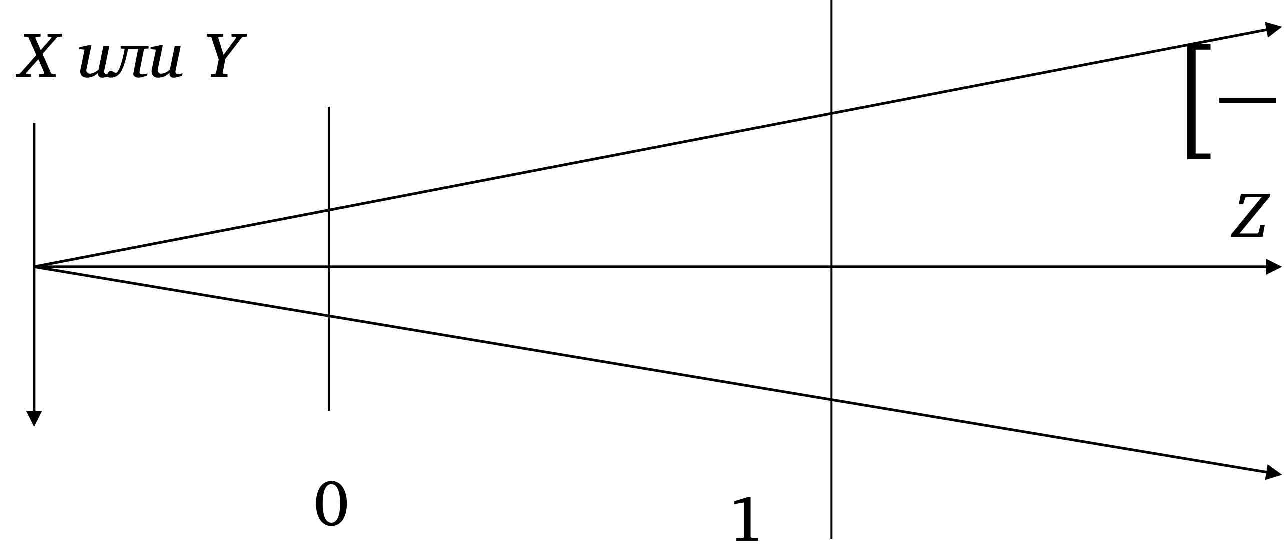
$$M_{proj}(S_w, S_h, z_f, z_b) = \begin{pmatrix} \frac{2}{S_w} z_f & 0 & 0 & 0 \\ 0 & \frac{2}{S_h} z_f & 0 & 0 \\ 0 & 0 & \frac{z_b}{z_b - z_f} & \frac{-z_f z_b}{z_b - z_f} \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$M_{proj}(S_w, S_h, z_f, z_b) \begin{pmatrix} x \\ y \\ z \\ \boxed{1} \end{pmatrix} = \begin{pmatrix} X_p \\ Y_p \\ Z_p \\ \boxed{W_p} \end{pmatrix} = \begin{pmatrix} x \frac{2z_f}{S_w} \\ y \frac{2z_f}{S_h} \\ \frac{z \cdot z_b}{z_b - z_f} - \frac{z_f \cdot z_b}{z_b - z_f} \\ z \end{pmatrix}$$

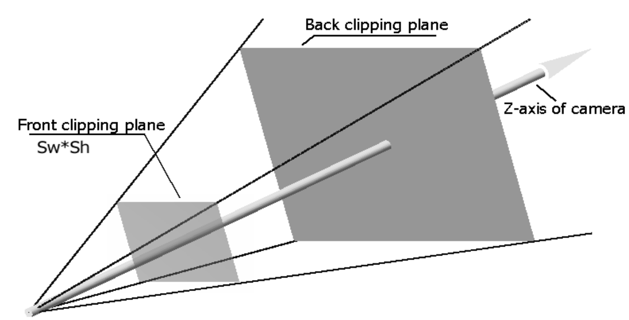


Преобразование к полукубу

$$M_{proj}(S_w,S_h,z_f,z_b)\begin{pmatrix}x\\y\\z\\\boxed{1}\end{pmatrix}=\begin{pmatrix}X_p\\Y_p\\Z_p\\\boxed{W_p}\end{pmatrix}=\begin{pmatrix}x\dfrac{2z_f}{S_w}\\[2mm] y\dfrac{2z_f}{S_h}\\[2mm] \dfrac{z\cdot z_b}{z_b-z_f}-\dfrac{z_f\cdot z_b}{z_b-z_f}\\[2mm] z\end{pmatrix}\Bigg\|\begin{pmatrix}x_p\\y_p\\z_p\\\boxed{1}\end{pmatrix}=\begin{pmatrix}x\dfrac{2z_f}{S_w}\dfrac{1}{z}\\[2mm] y\dfrac{2z_f}{S_h}\dfrac{1}{z}\\[2mm] \left(\dfrac{z\cdot z_b}{z_b-z_f}-\dfrac{z_f\cdot z_b}{z_b-z_f}\right)\dfrac{1}{z}\\[2mm] 1\end{pmatrix}$$



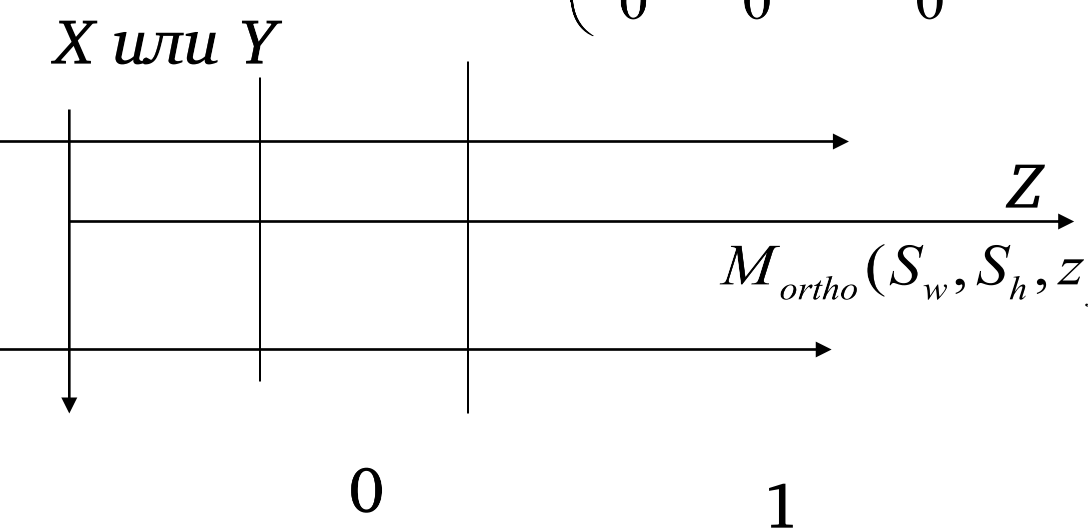
$$[-1,1]\times[-1,1]\times[0,1]$$



Если ортогональное проецирование

Преобразование к полукубу

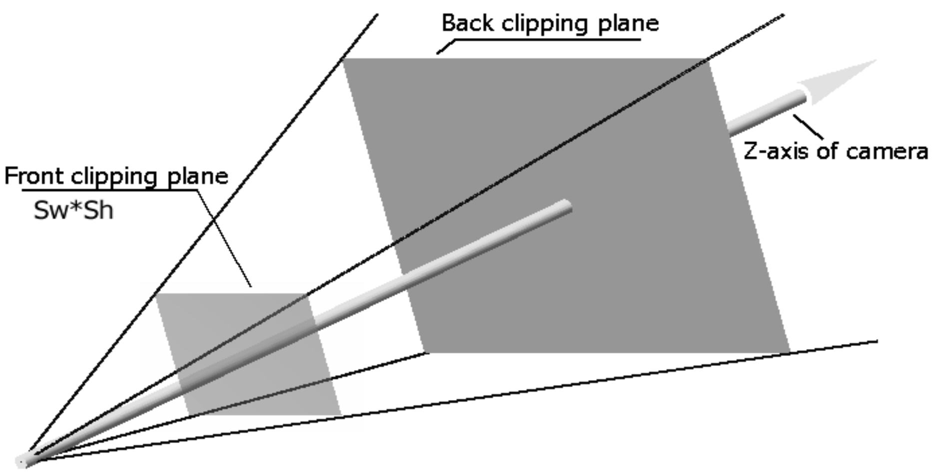
$$M_{ortho}(S_w, S_h, z_f, z_b) = \begin{pmatrix} \frac{2}{S_w} & 0 & 0 & 0 \\ 0 & \frac{2}{S_h} & 0 & 0 \\ 0 & 0 & \frac{1}{z_b - z_f} & \frac{-z_f}{z_b - z_f} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

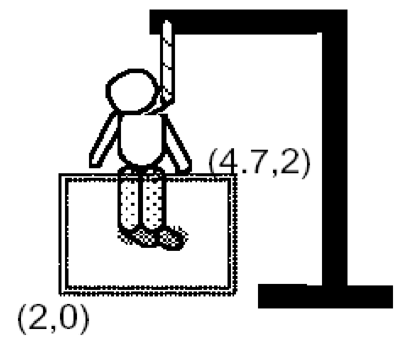


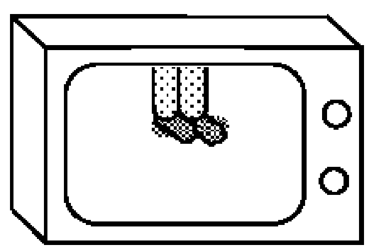
$$M_{ortho}(S_w, S_h, z_f, z_b) \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} X_p \\ Y_p \\ Z_p \\ W_p \end{pmatrix} = \begin{pmatrix} x \frac{2}{S_w} \\ y \frac{2}{S_h} \\ \frac{z}{z_b - z_f} - \frac{z_f}{z_b - z_f} \\ 1 \end{pmatrix}$$

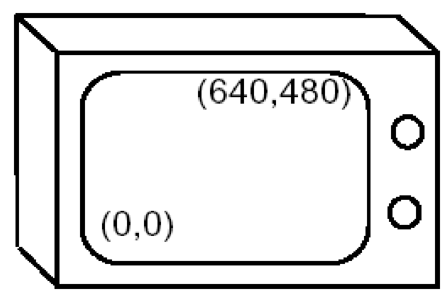
Порт вывода (viewport)

Порт вывода – это не что иное, как прямоугольник на передней клиппирующей плоскости.

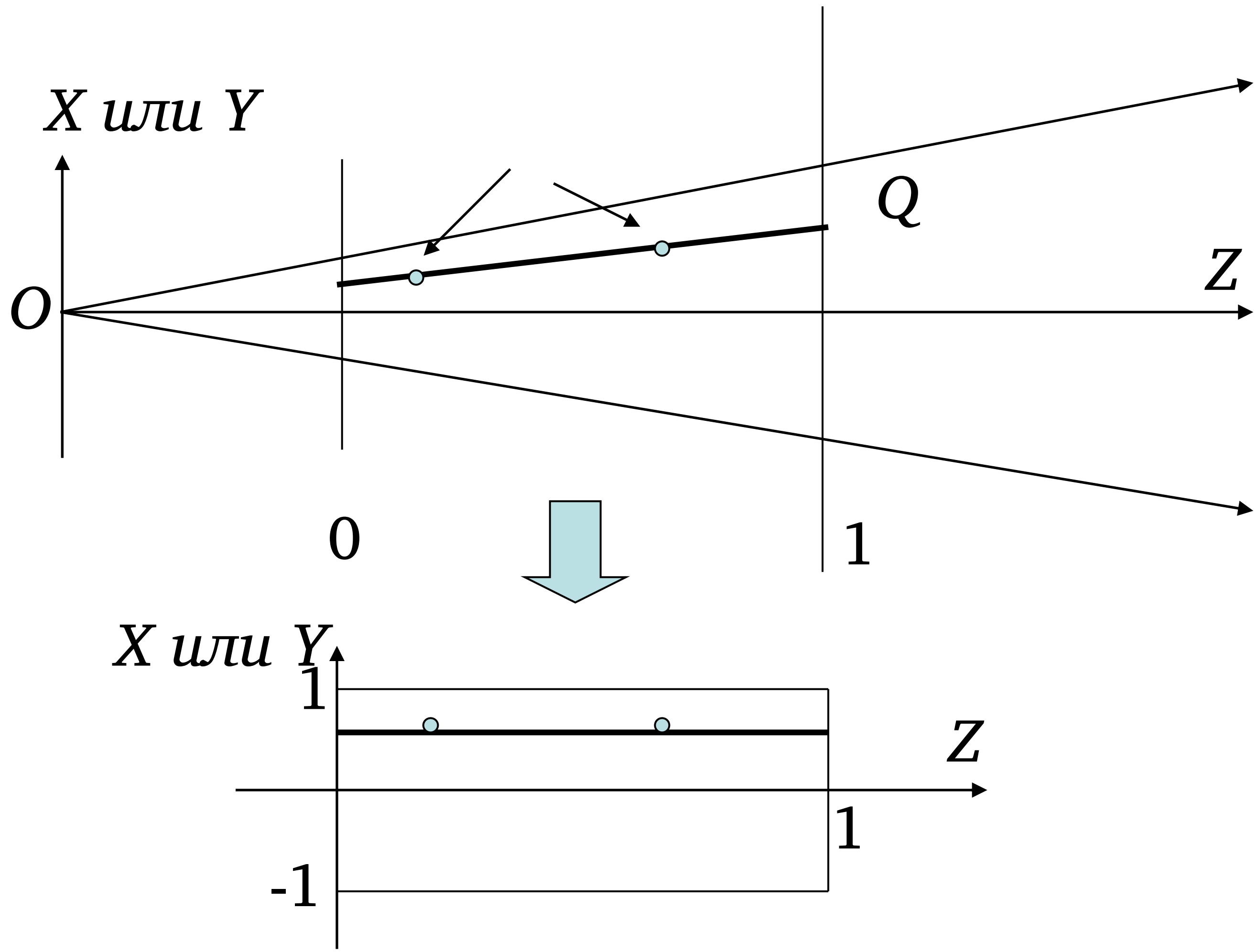




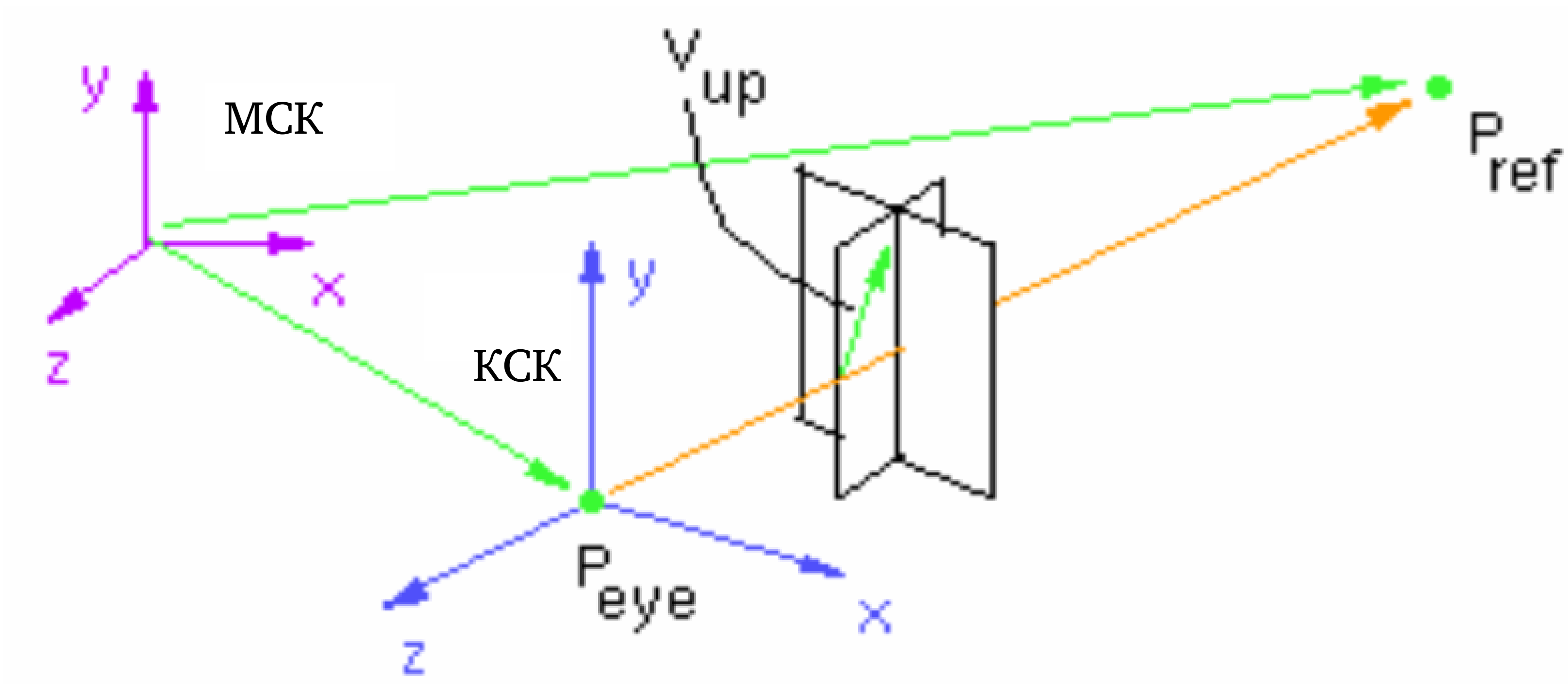




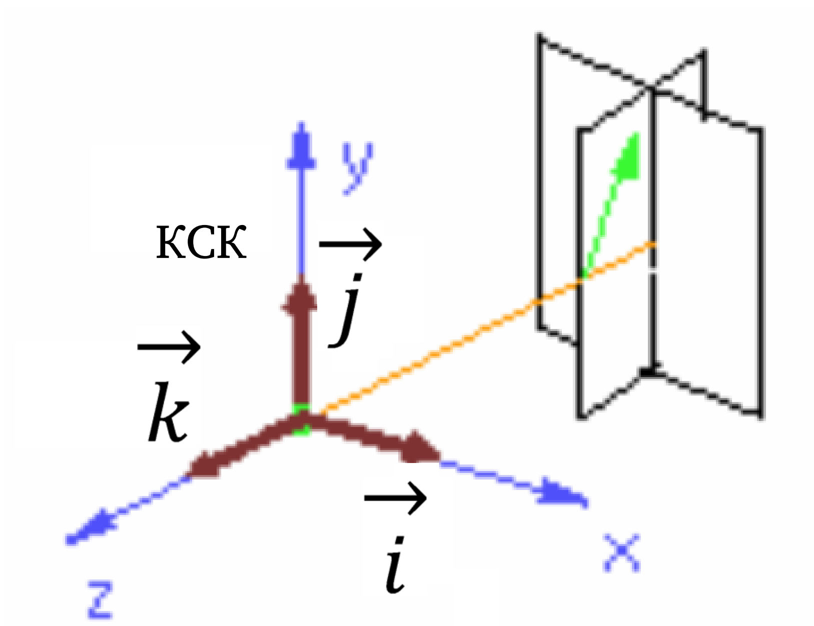
Алгоритм Z-буфера



Определение камеры



Определение камеры, орты



$$\vec{k} = \frac{P_{eye} - P_{ref}}{\|P_{ref} - P_{eye}\|}$$

$$\vec{I} = V_{up} \times \vec{k}$$

$$\vec{i} = \vec{I} / \|\vec{I}\|$$

$$\vec{j} = \vec{k} \times \vec{i}$$

Определение камеры, матрица

$$M_{cam} = \begin{pmatrix} i_x & i_y & i_z & 0 \\ j_x & j_y & j_z & 0 \\ k_x & k_y & k_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -P_{eye,x} \\ 0 & 1 & 0 & -P_{eye,y} \\ 0 & 0 & 1 & -P_{eye,z} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$M_{cam}^{-1} = \begin{pmatrix} 1 & 0 & 0 & P_{eye,x} \\ 0 & 1 & 0 & P_{eye,y} \\ 0 & 0 & 1 & P_{eye,z} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} i_x & j_x & k_x & 0 \\ i_y & j_y & k_y & 0 \\ i_z & j_z & k_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$