Аффинные преобразования и проецирование в 3D

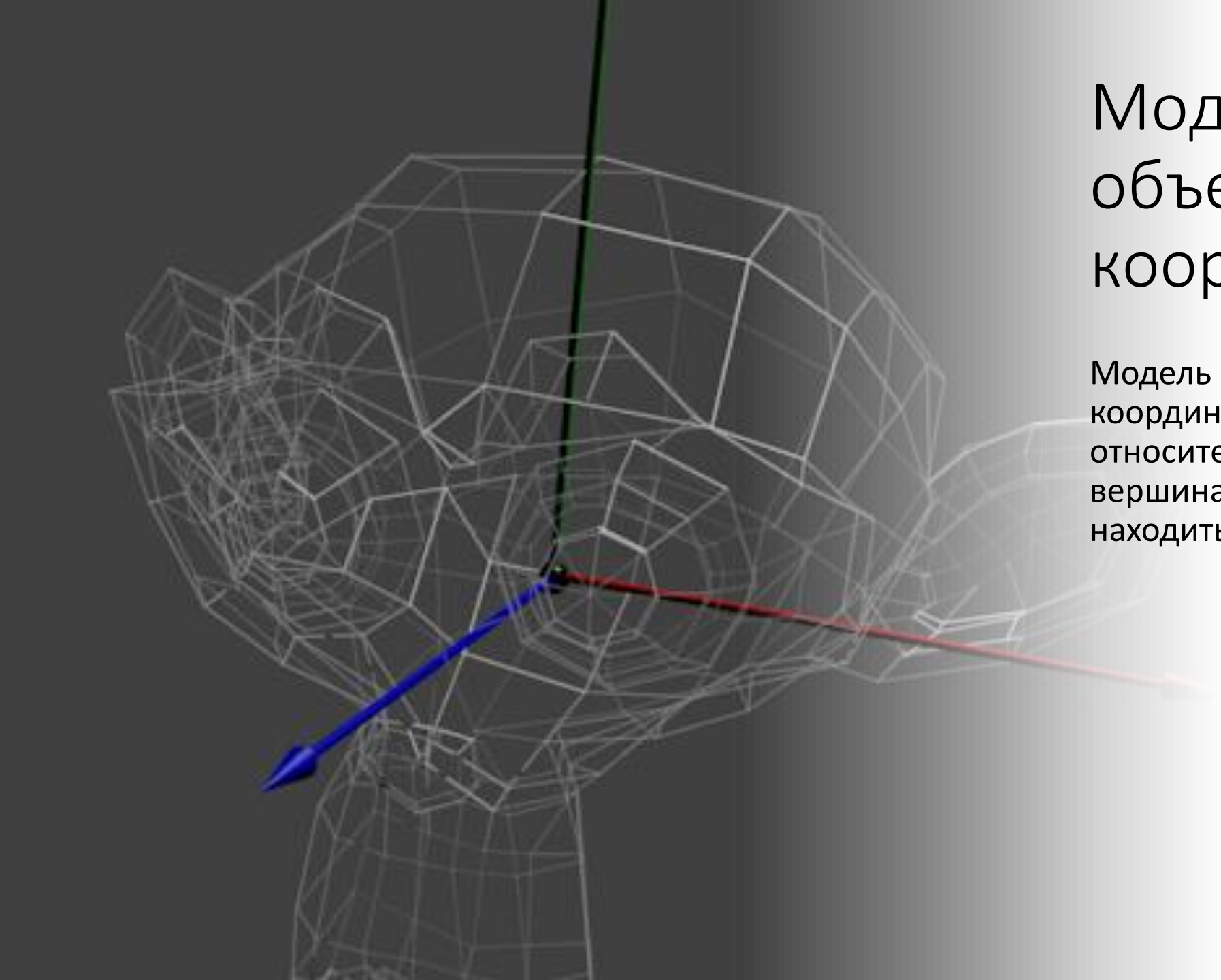
Компьютерная графика

Лектор: Демяненко Я.М., 4-й курс ФИИТ, 2026



Вопрос

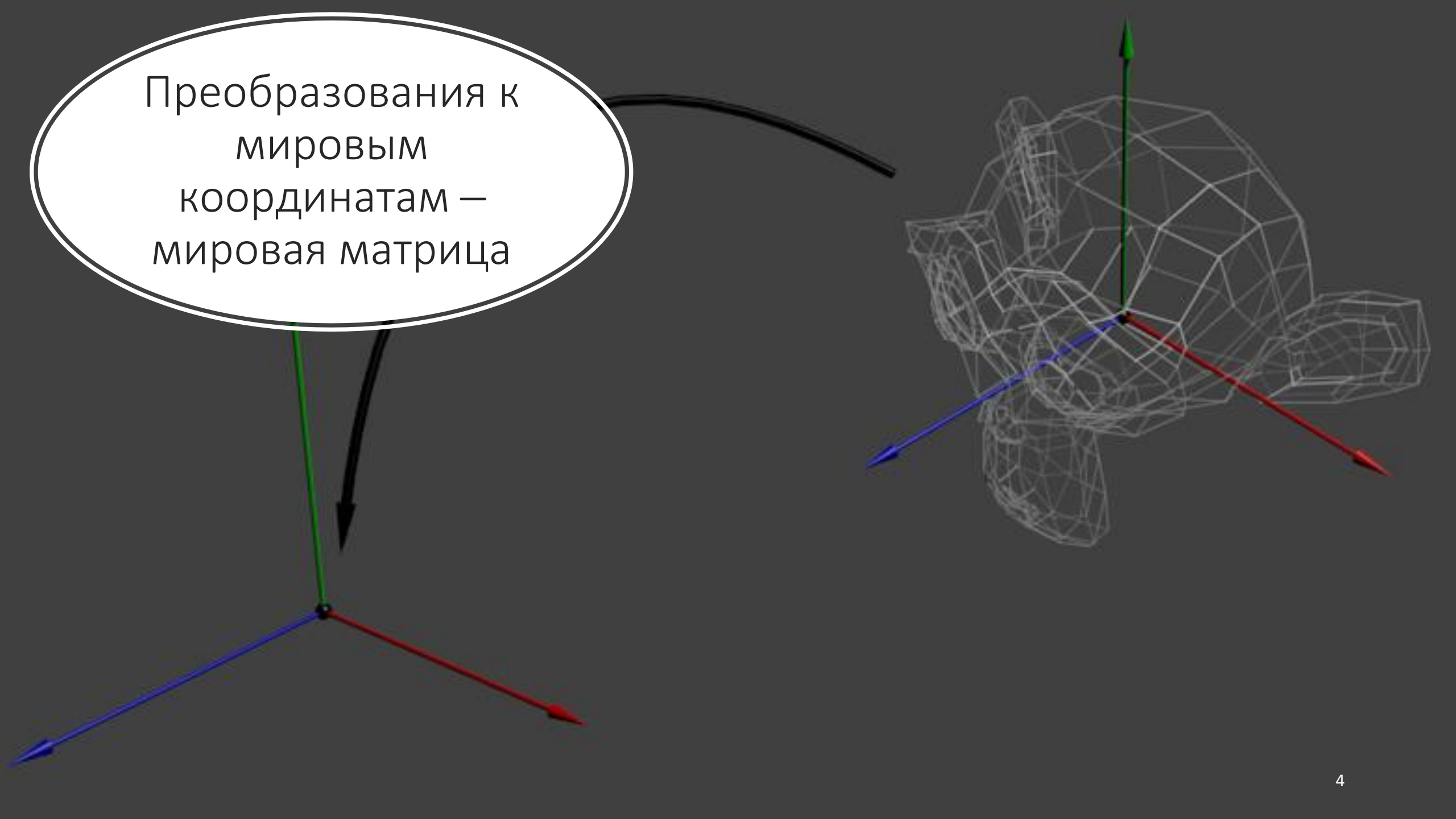
Как 3D объект нарисовать на 2D экране?



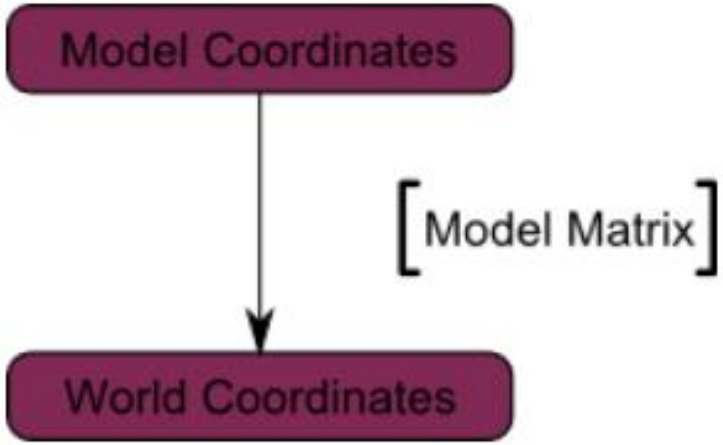
Модель в объектных координатах

Модель задается множеством вершин, координаты которых заданы относительно центра объекта, т. е. вершина с координатами (0, 0, 0) будет находиться в центре объекта.

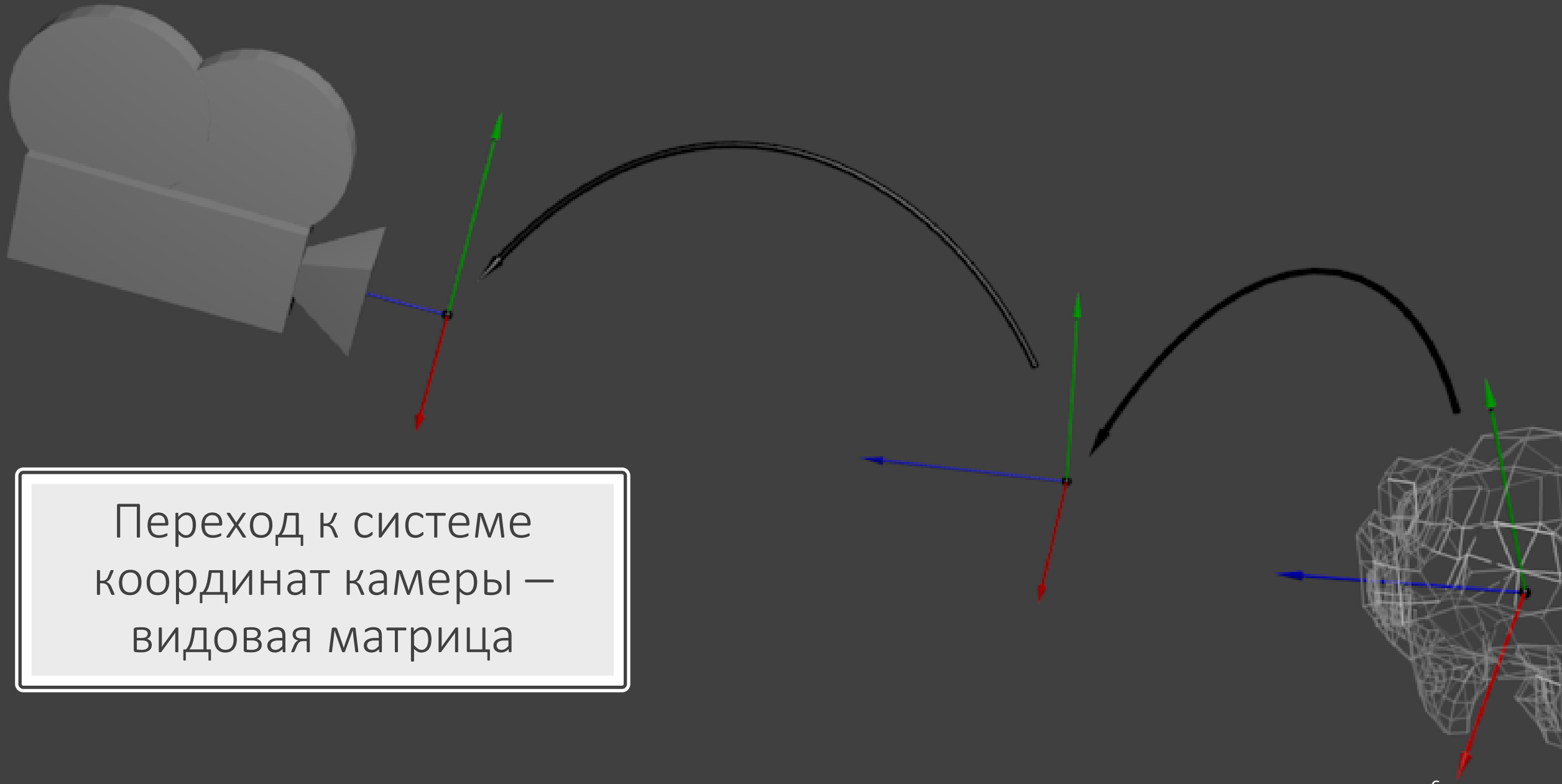
Преобразования к мировым координатам – мировая матрица



Преобразования к мировым координатам – мировая матрица

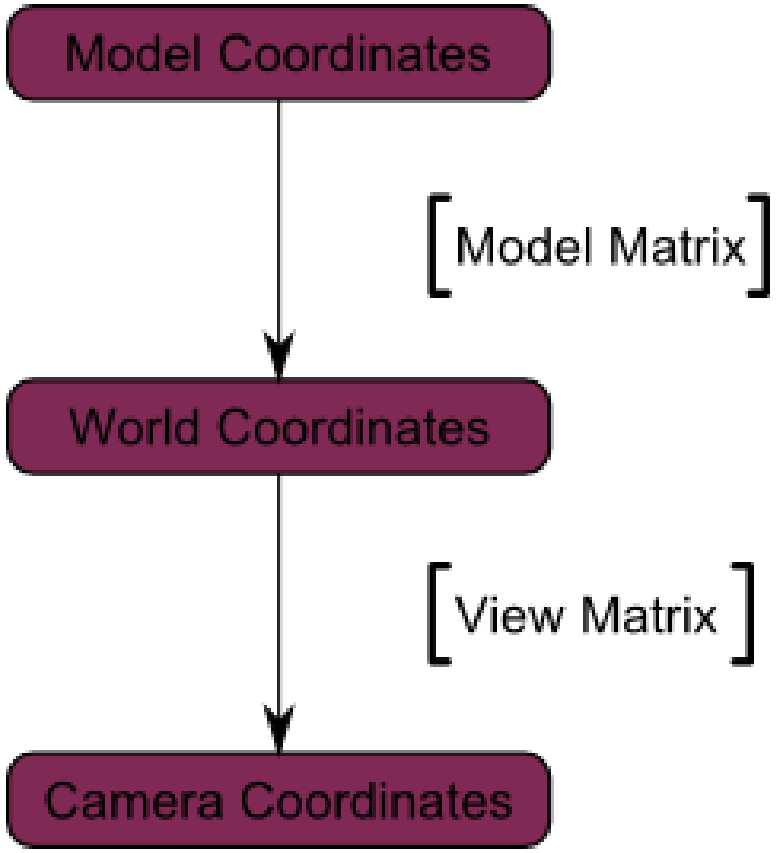


Мы перешли из пространства объекта
(все вершины заданы относительно центра объекта)
к мировому пространству
(все вершины заданы относительно центра мира)



Переход к системе координат камеры – видовая матрица

Преобразования к системе координат камеры – видовая матрица



Мы перешли из мировой системы координат
(все вершины заданы относительно центра мировой системы)
к системе координат камеры
(все вершины заданы относительно камеры)

Переход из пространства камеры в однородное пространство



API	X	Y	Z
OpenGL	[-1, 1]	[-1, 1]	[-1, 1]
DirectX	[-1, 1]	[-1, 1]	[0, 1]
Vulkan	[-1, 1]	[-1, 1]	[0, 1]

NDC — это Normalized Device Coordinates (нормализованные координаты устройства)

NDC — это промежуточное пространство координат между камерой и экраном

NDC — это куб (или параллелепипед), в котором все видимые объекты находятся внутри

Зачем нужен NDC?

Проблема: разные экраны — разные разрешения

У вас есть монитор 1920×1080, телефон 1080×2400, VR-шлем 2160×2160

Как один и тот же 3D-объект отобразить на разных экранах?

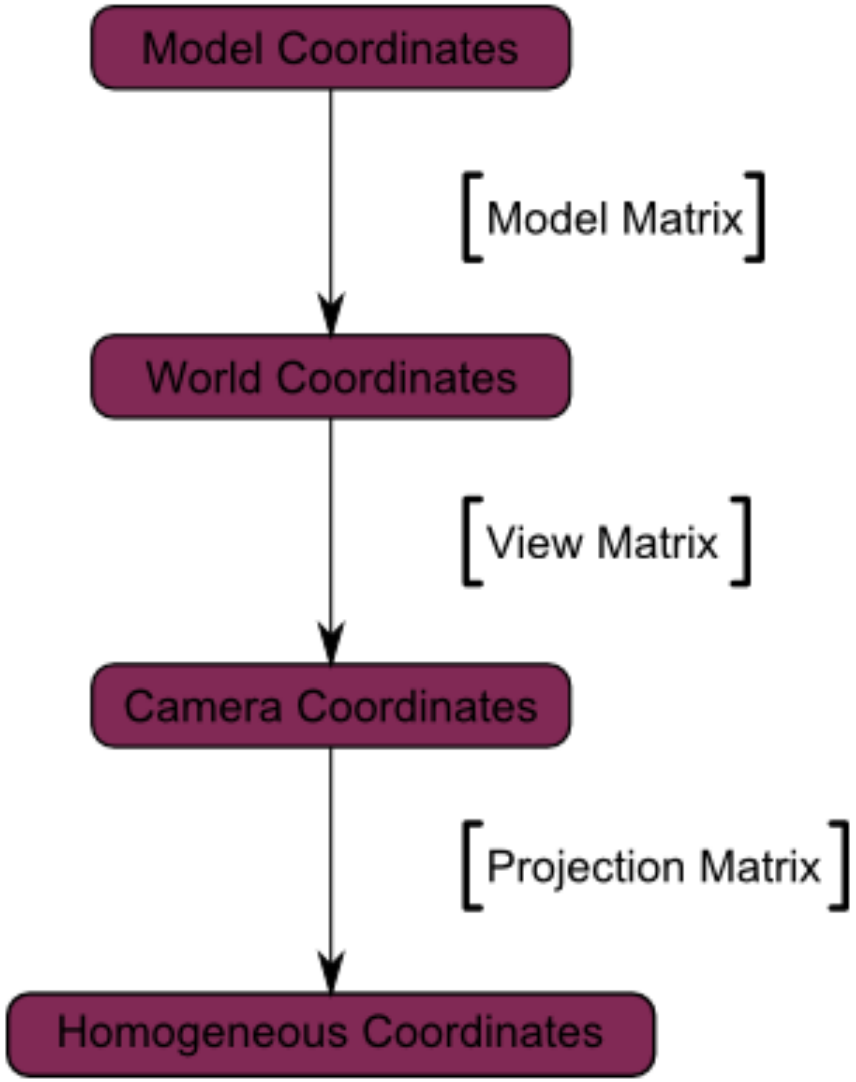
Решение: сначала нормализовать координаты в единичный куб (NDC), а потом масштабировать под конкретный экран

Преимущества NDC

Преимущество	Объяснение
Независимость от разрешения	Один и тот же NDC-куб → любой экран
Отсечение (Clipping)	Всё, что вне куба [-1,1] — отбрасывается
Перспективное деление	Деление на w автоматически даёт NDC
Простота viewport	Просто масштабировать из [-1,1] в [0, width]

Преобразования к однородным координатам

Мы перешли из пространства камеры
(все вершины заданы относительно камеры)
в однородное пространство
(все вершины находятся в небольшом кубе.
Все, что находится внутри куба - выводится на экран).

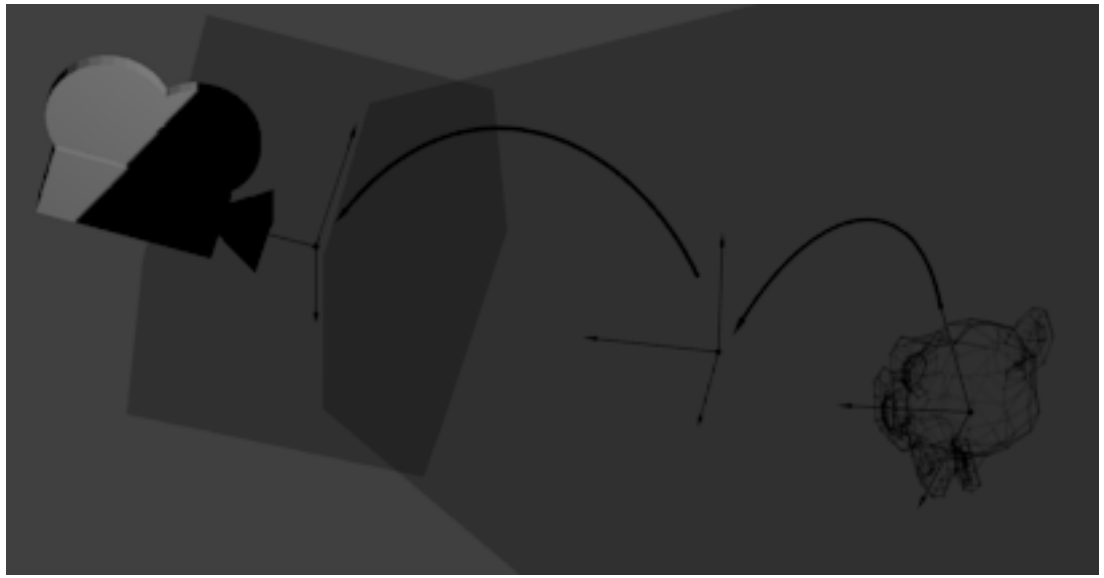


Перспективная проекция

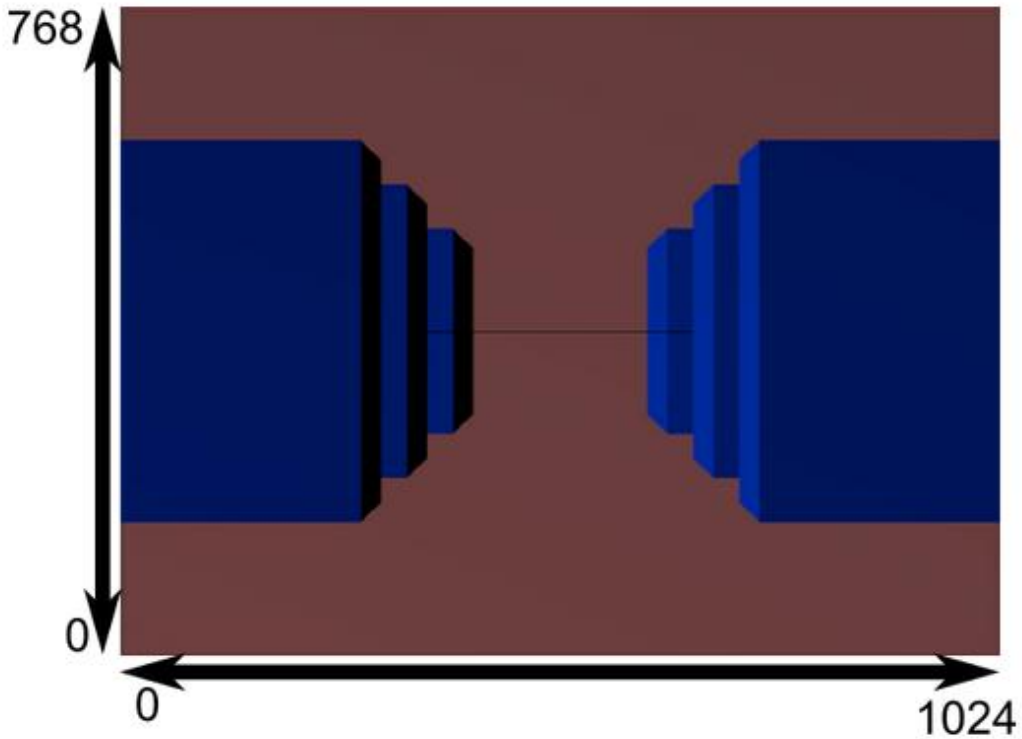
Вершина, которая получит координаты x == 0 и y == 0 будет отображаться по центру экрана

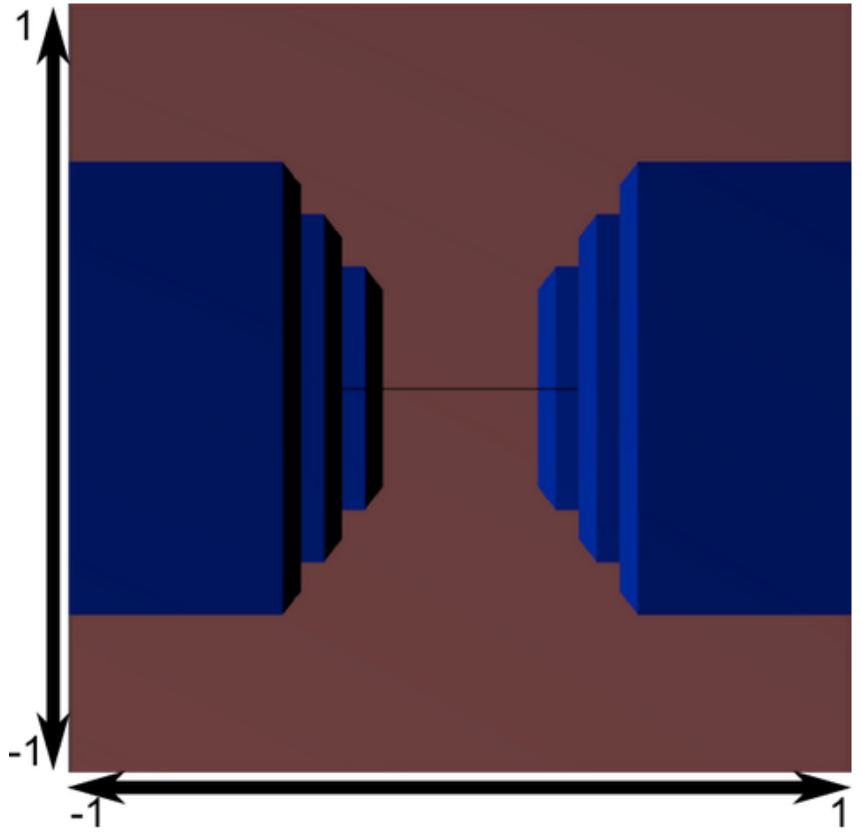
Однако, при отображении объекта огромную роль играет также дистанция до камеры (z)

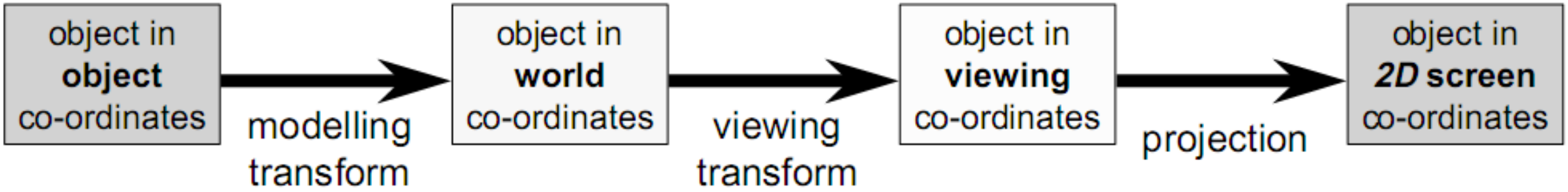
Для двух вершин, с одинаковыми x и y, вершина имеющая большее значение по z будет отображаться ближе, чем другая.



Последний шаг







- модельно-видовая как одна матрица (мировая x видовая)
- как мировая, так и видовая могут быть единичными

Разнообразие преобразований – мировая, видовая и проекционная матрицы

Аффинные преобразования в 3D

- аналогичны аффинным преобразования в 2D
- однородные координаты (x, y, z, w) → (x/w, y/w, z/w)
- базовые преобразования
  - повороты вокруг осей
  - смещения
  - растяжения
  - отражения

Базовые аффинные преобразования

translation

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t_x & t_y & t_z & 1 \end{bmatrix}$$

identity

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

rotation about x-axis

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

scale

$$\begin{bmatrix} m_x & 0 & 0 & 0 \\ 0 & m_y & 0 & 0 \\ 0 & 0 & m_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

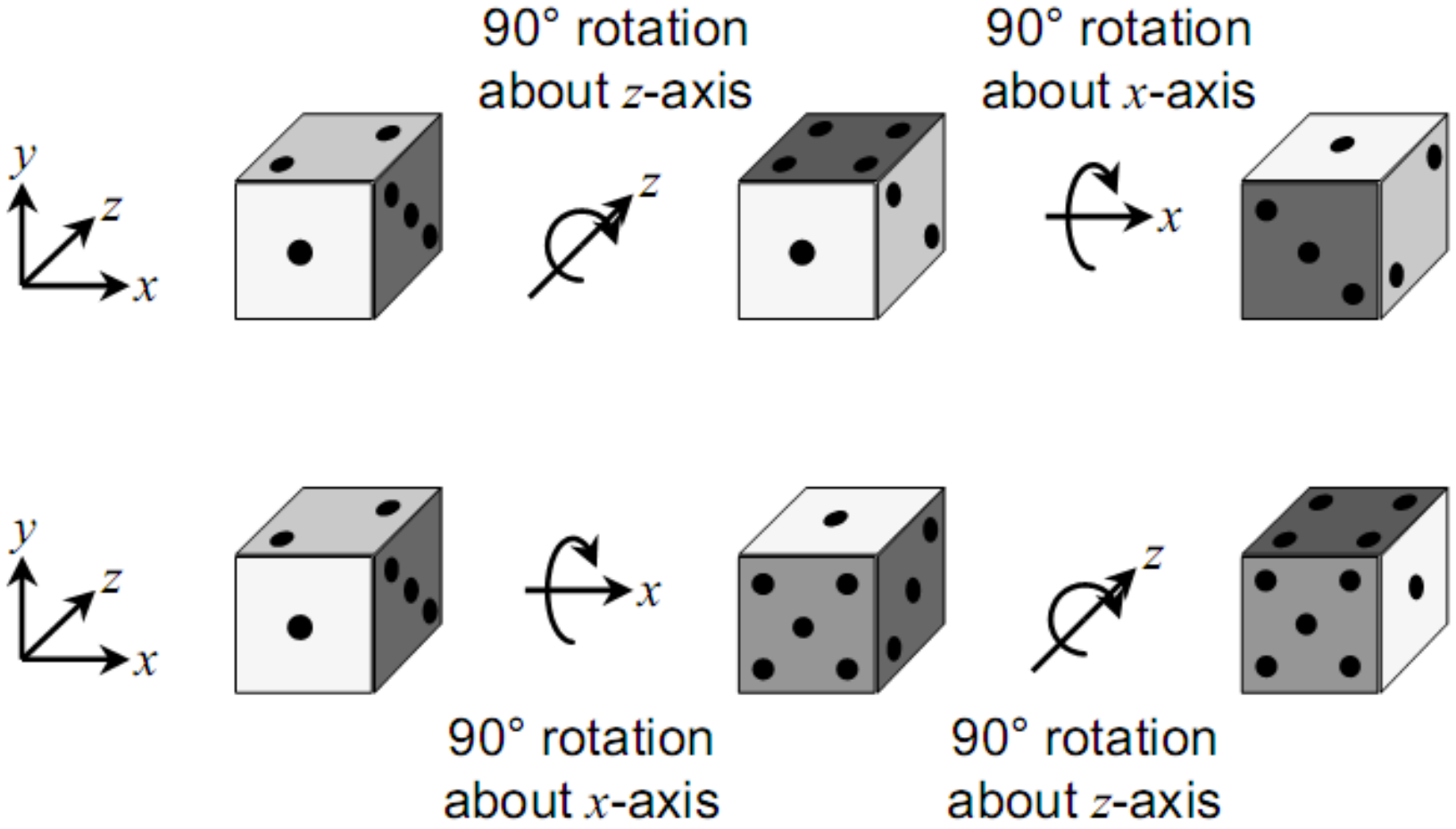
rotation about z-axis

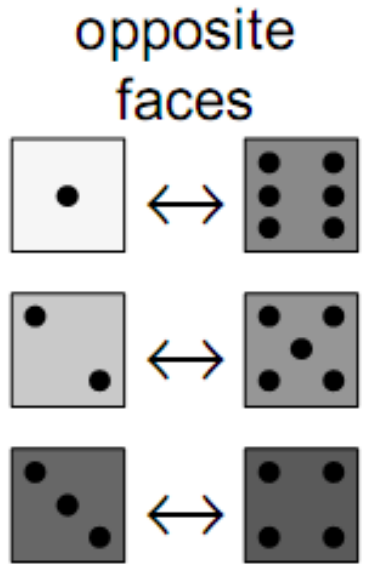
$$\begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

rotation about y-axis

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

Преобразования некоммутативны





Задача

- Повернуть объект вокруг произвольной прямой L в пространстве на заданный угол
- Объект задаётся списком вершин и списком рёбер
- Прямая задаётся точкой A (a,b,c) , через которую она проходит, и единичным вектором (l,m,n)

Результирующая матрица

$$\begin{pmatrix} l^2 + \cos\varphi(1 - l^2) & l(1 - \cos\varphi)m + n\sin\varphi & l(1 - \cos\varphi)n - m\sin\varphi & 0 \\ l(1 - \cos\varphi)m - n\sin\varphi & m^2 + \cos\varphi(1 - m^2) & m(1 - \cos\varphi)n + l\sin\varphi & 0 \\ l(1 - \cos\varphi)n + m\sin\varphi & m(1 - \cos\varphi)n - l\sin\varphi & n^2 + \cos\varphi(1 - n^2) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

План

1. Перенести прямую L в центр координат на –A (-a,-b,-c)
2. Совместить прямую L с одной из координатных осей
3. Выполнить поворот объекта вокруг прямой L
4. Выполнить преобразования 1 и 2 в обратной последовательности

Решение

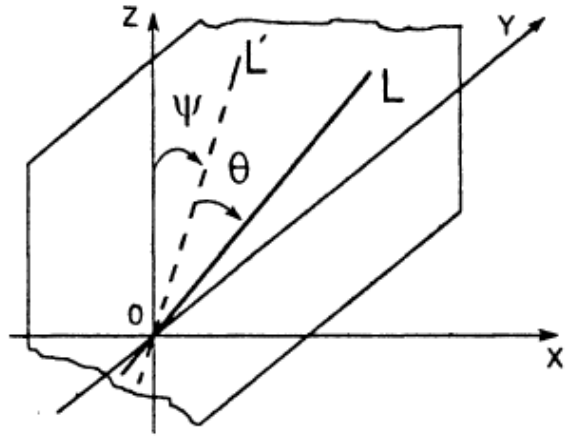
1. Перенести прямую L в центр координат на –A (-a,-b,-c)
2. Совместить прямую L с одной из координатных осей, например, Z
   - Повернуть прямую L вокруг Ox
   - Повернуть прямую L вокруг Oy
3. Выполнить поворот объекта вокруг прямой L
4. Выполнить повороты 2 в обратной последовательности на обратные углы
5. Выполнить перенос на A (a, b, c)

Перенос на –A (-a,-b,-c)

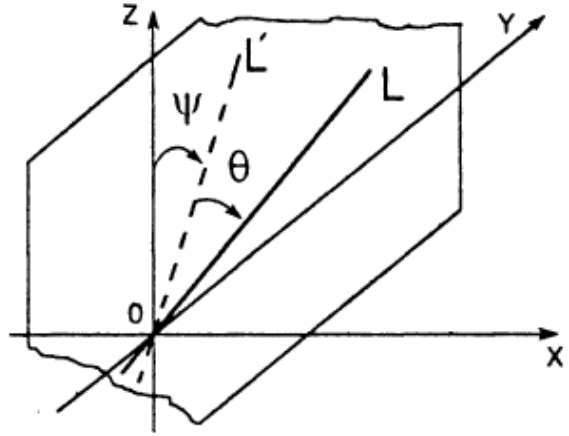
$$[T] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -a & -b & -c & 1 \end{bmatrix}$$

Совмещение прямой L с осью Z

- Повернуть прямую L вокруг Ox на угол ψ



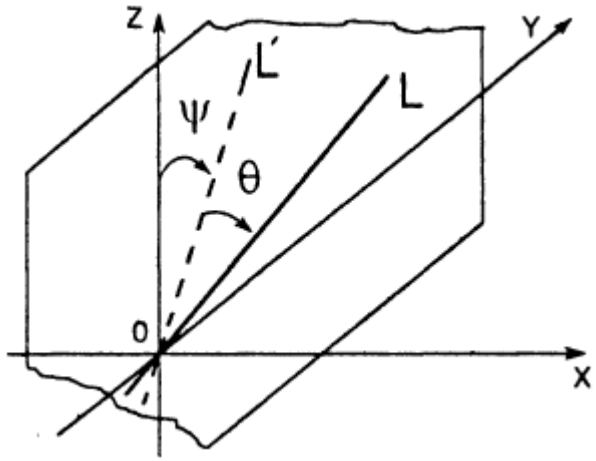
- Повернуть прямую L вокруг Oy на угол θ



Поворот прямой L вокруг Ox на угол ψ

Рассмотрим L' – проекцию на YZ – (0,m,n)

$$[R_x] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{n}{d} & \frac{m}{d} & 0 \\ 0 & -\frac{m}{d} & \frac{n}{d} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



$$(l, m, n, 1)[R_x] = (l, 0, d, 1)$$

$$\cos\psi = \frac{n}{d}, \quad \sin\psi = \frac{m}{d},$$

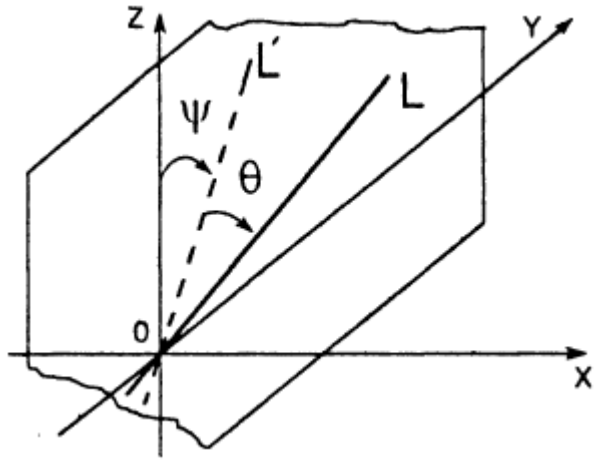
$$d = \sqrt{m^2 + n^2}.$$

Поворот прямой L вокруг Oy на угол θ

$$(l,\ m,\ n,\ 1)\left[R_x\right] = (l,\ 0,\ d,\ 1)$$

$$\cos\theta = l,\quad \sin\theta = -d.$$

$$\left[R_y\right] = \begin{bmatrix} l & 0 & d & 0 \\ 0 & 1 & 0 & 0 \\ -d & 0 & l & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



$$d = \sqrt{m^2 + n^2}\ .$$

Поворот объекта вокруг прямой L на угол ф

$$[R_z] = \begin{bmatrix} \cos\varphi & \sin\varphi & 0 & 0 \\ -\sin\varphi & \cos\varphi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Обратные преобразования

- Поворот прямой L вокруг Ox на угол -θ
- Поворот прямой L вокруг Oy на угол –ψ
- Перенос на A (a, b, c)

$$[T][R_x][R_y][R_z][R_y]^{-1}[R_x]^{-1}[T]^{-1}$$

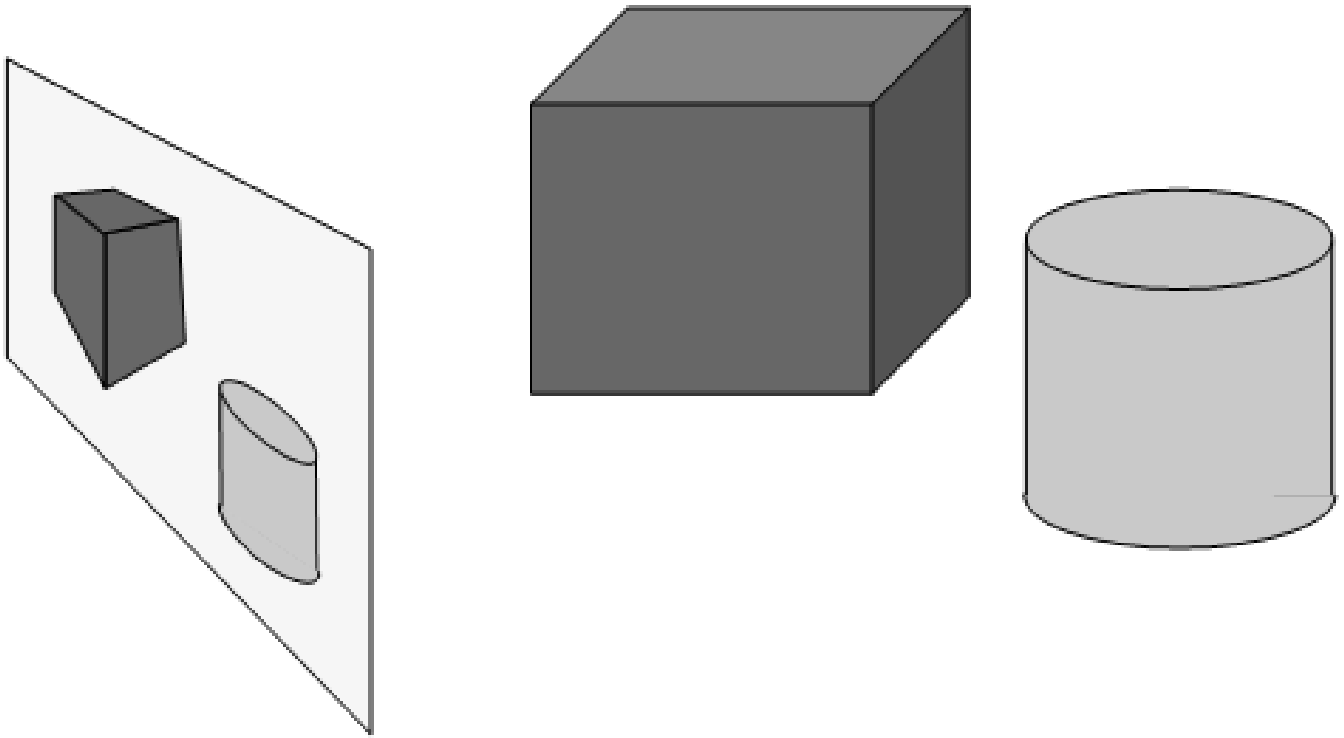
Результирующая матрица

$$\begin{pmatrix} l^2 + \cos\varphi(1 - l^2) & l(1 - \cos\varphi)m + n\sin\varphi & l(1 - \cos\varphi)n - m\sin\varphi & 0 \\ l(1 - \cos\varphi)m - n\sin\varphi & m^2 + \cos\varphi(1 - m^2) & m(1 - \cos\varphi)n + l\sin\varphi & 0 \\ l(1 - \cos\varphi)n + m\sin\varphi & m(1 - \cos\varphi)n - l\sin\varphi & n^2 + \cos\varphi(1 - n^2) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Для других задач (в общем виде)

$$[A] = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 & 0 \\ \beta_1 & \beta_2 & \beta_3 & 0 \\ \gamma_1 & \gamma_2 & \gamma_3 & 0 \\ \lambda & \mu & \nu & 1 \end{bmatrix}$$

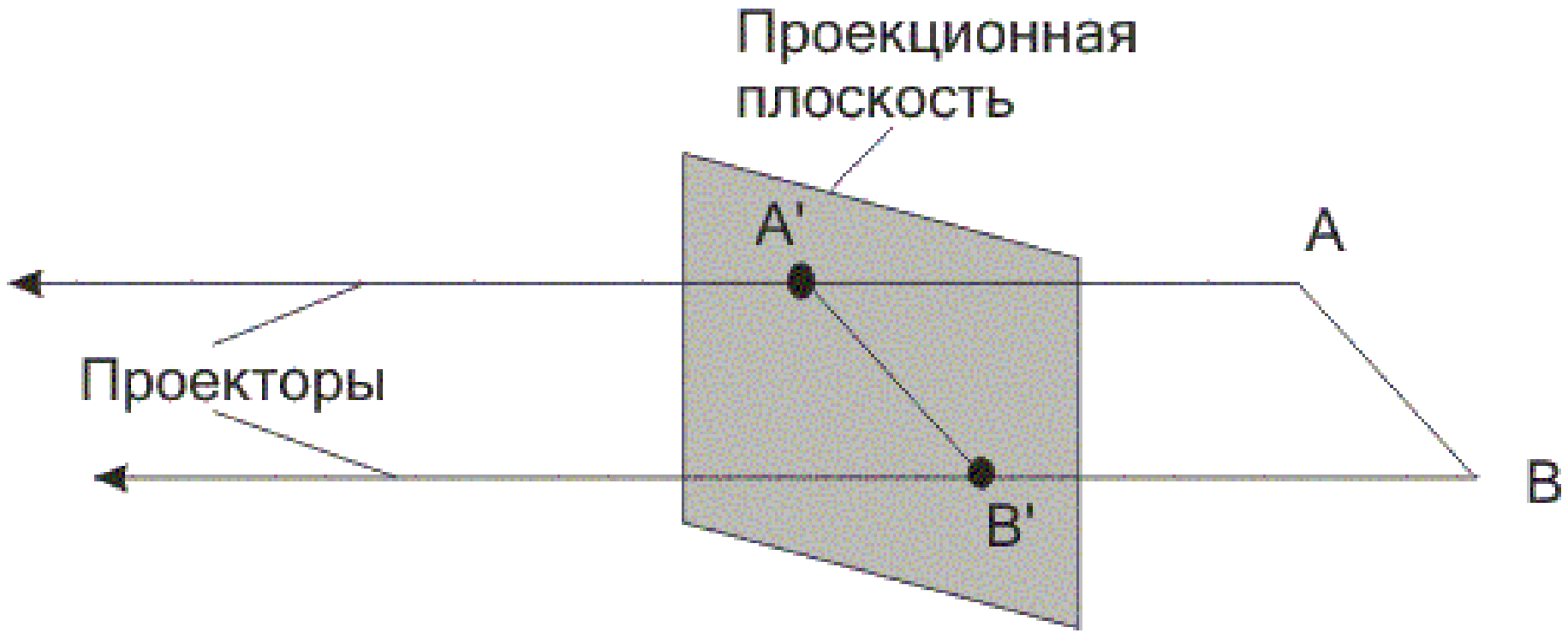
Проецирование



Основные типы проекций

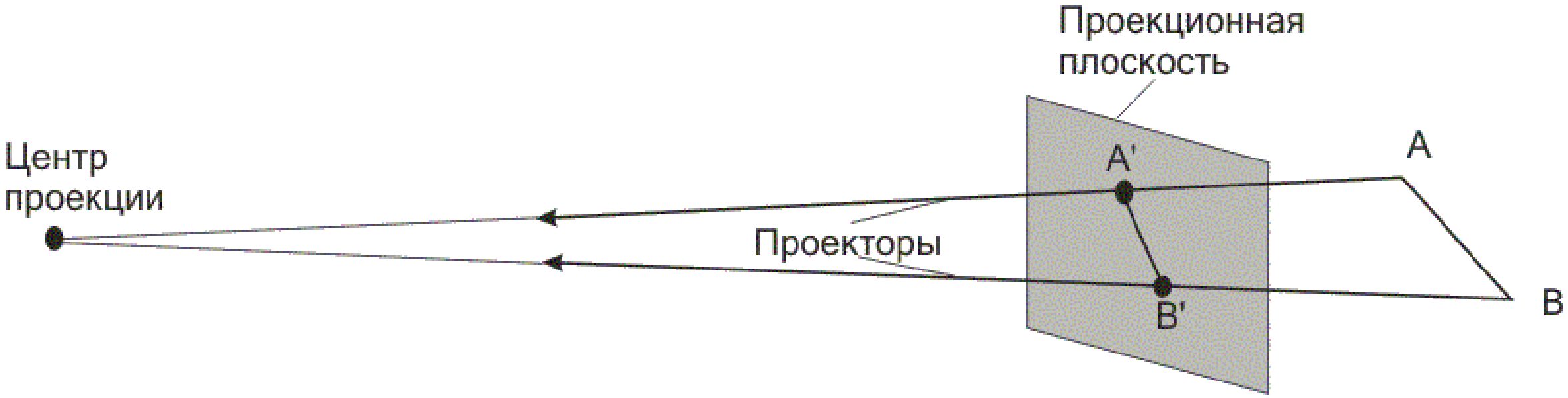
Параллельная

Центральная (перспективная)



Параллельная проекция

- (x,y,z) → (x,y)
- используется в САПР (CAD), архитектуре и т.п.
- выглядит нереалистично



Центральная (перспективная) проекция

- (x,y,z) → (x/z,y/z)
- уменьшение с удалением
- выглядит реалистично
- так работают камеры

Перспективное преобразование

ПП – это преобразование одного трёхмерного пространства в другое, при котором:

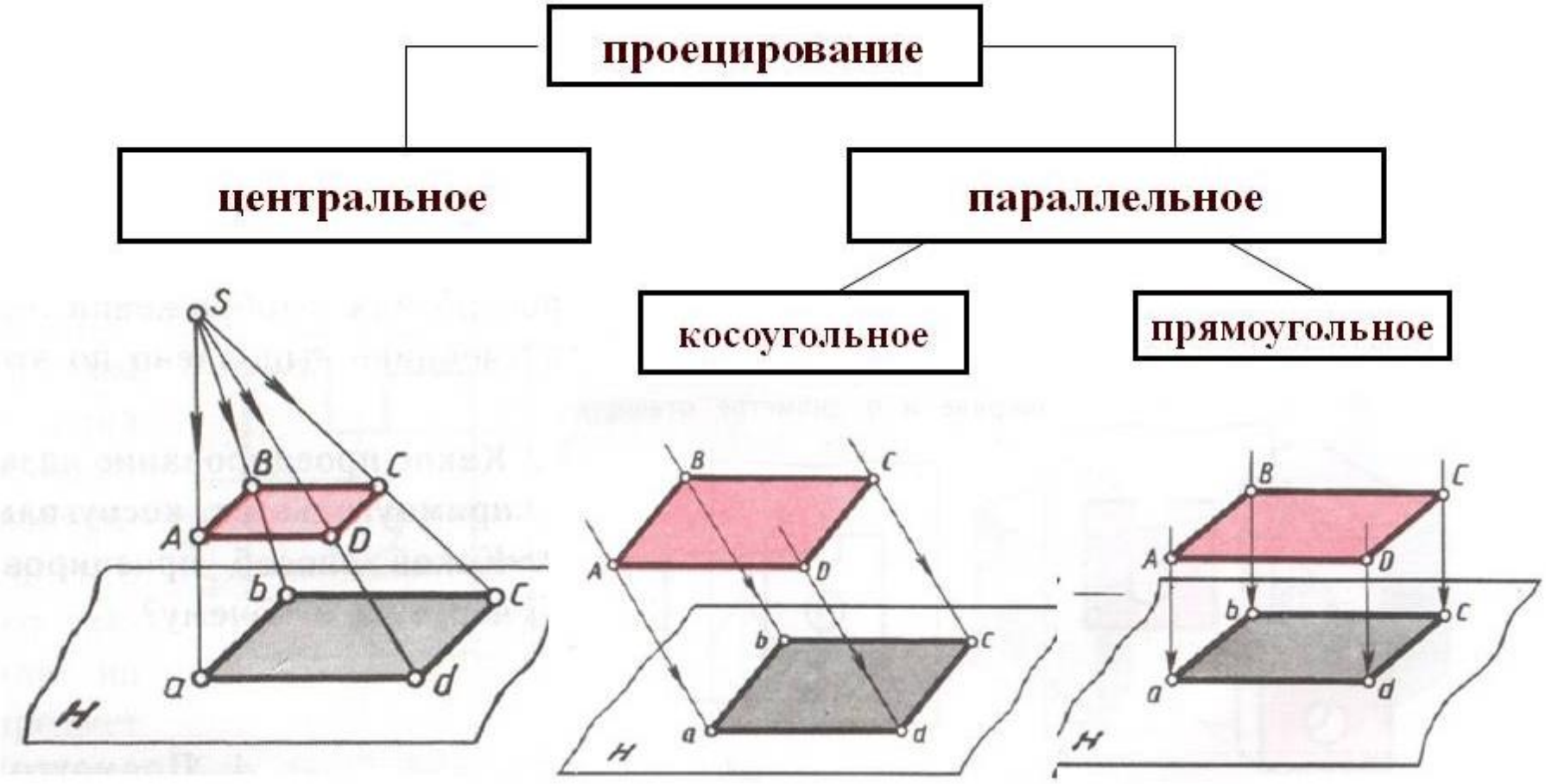
- параллельные прямые сходятся,
- размер объекта уменьшается с увеличением расстояния до центра наблюдения,
- и происходит неоднородное искажение линий объекта, зависящее от ориентации и расстояния от объекта до центра проекции

Это помогает нашему восприятию глубины, но не сохраняет форму объекта

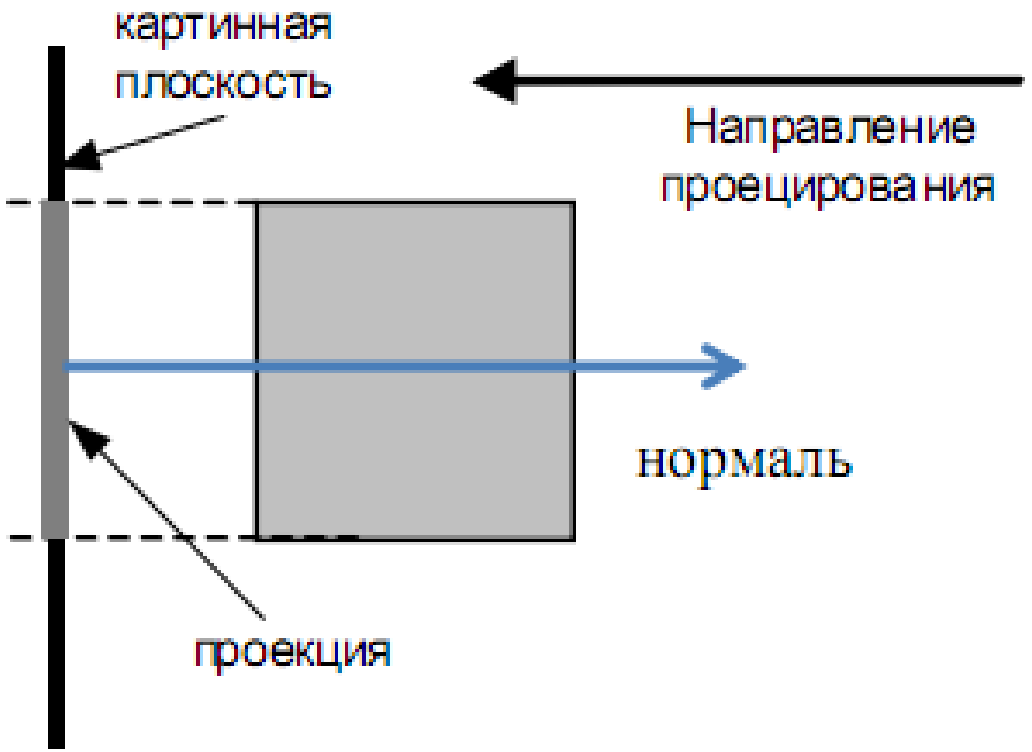
Классификация параллельных проекций



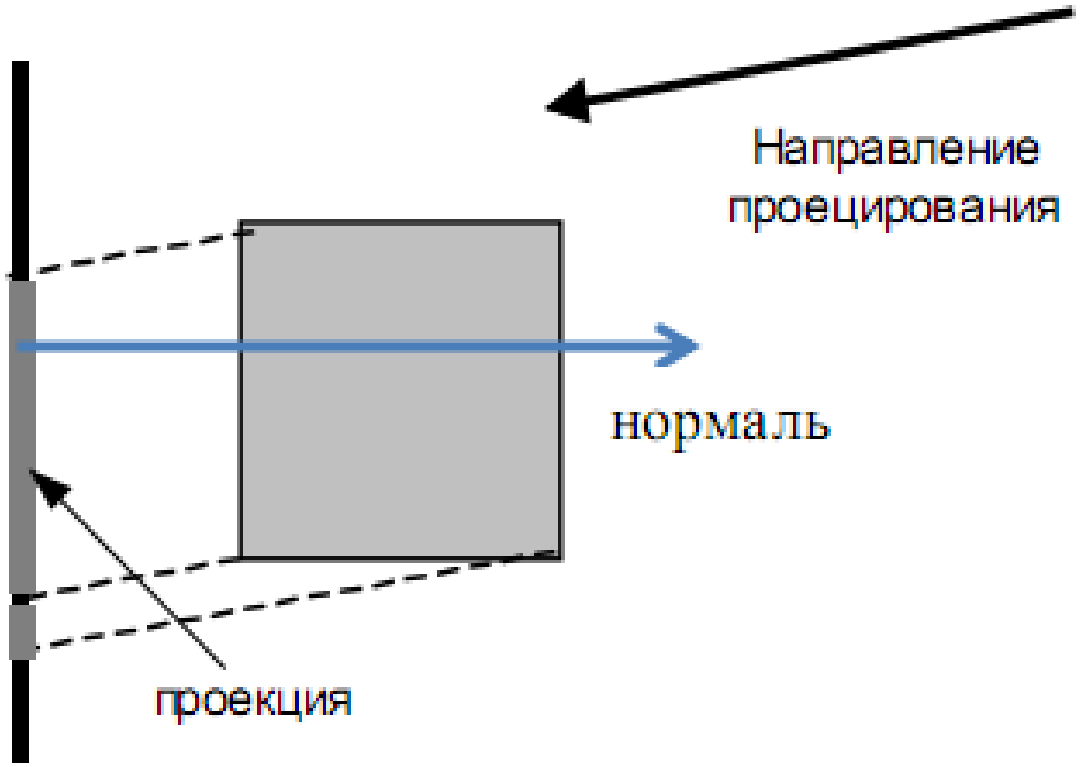
Как это выглядит



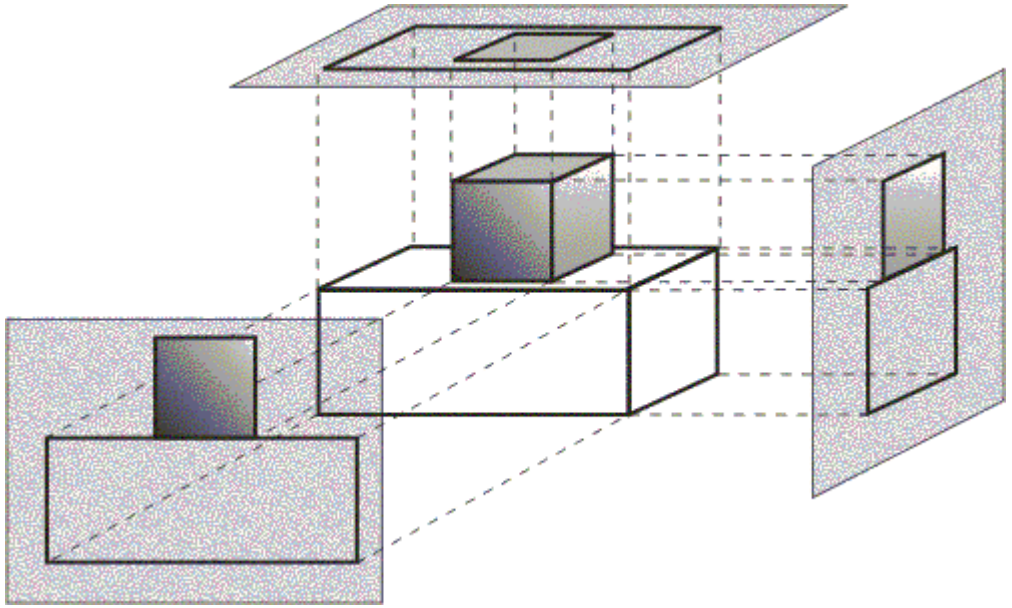
Ортографические, аксонометрические проекции



Косоугольные проекции



Ортографические проекции. Где применяются?



$$[P_x] \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ p & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ p & 0 & 0 & 1 \end{bmatrix}$$

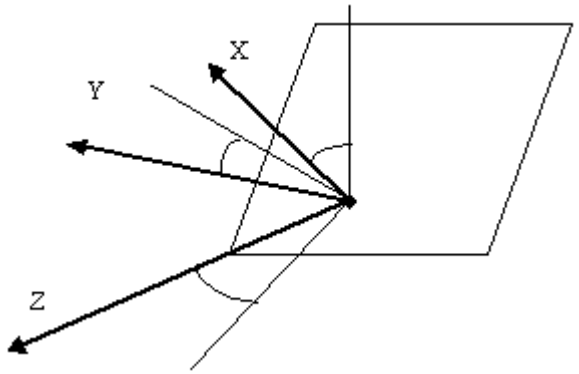
$$[P_x] = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & q & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & r & 1 \end{bmatrix}$$

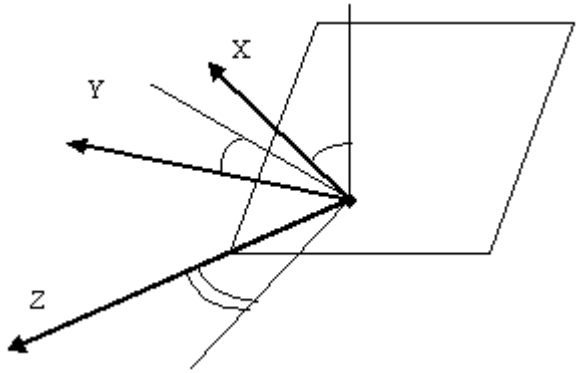
Аксонометрические проекции

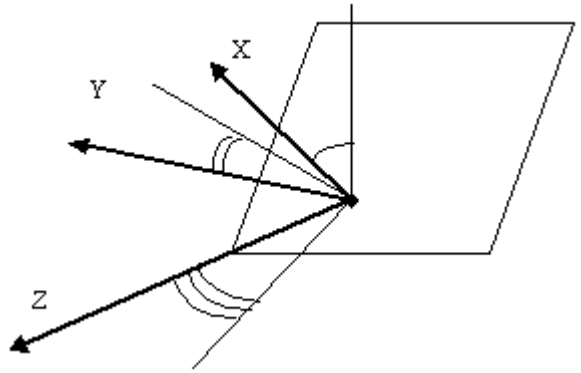
$$\begin{bmatrix} \cos\psi & 0 & -\sin\psi & 0 \\ 0 & 1 & 0 & 0 \\ \sin\psi & 0 & \cos\psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\varphi & \sin\varphi & 0 \\ 0 & -\sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

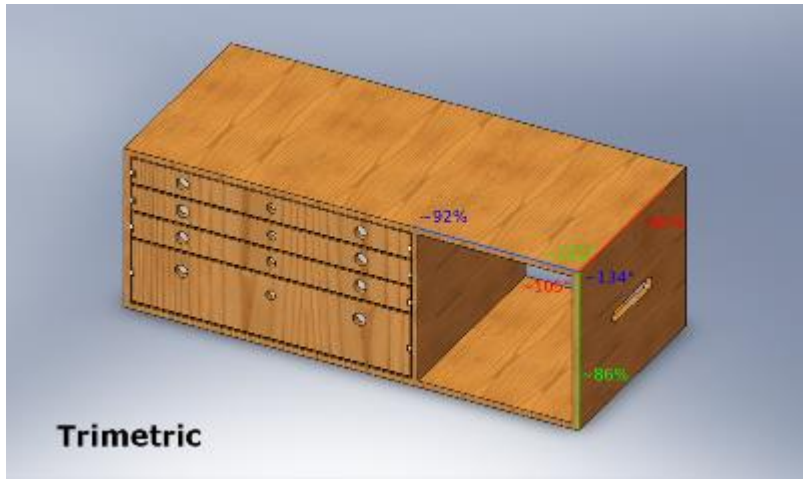
$$[M] = \begin{bmatrix} \cos\psi & \sin\phi\sin\psi & 0 & 0 \\ 0 & \cos\varphi & 0 & 0 \\ \sin\psi & -\sin\varphi\cos\psi & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

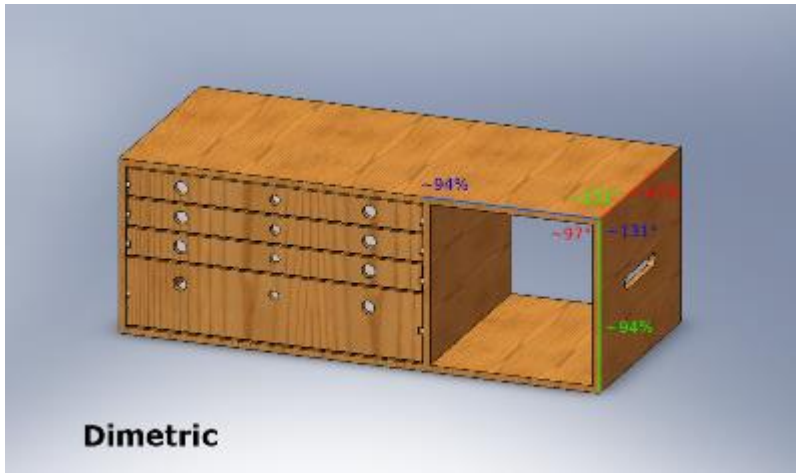
Аксонометрические проекции. Где применяются?

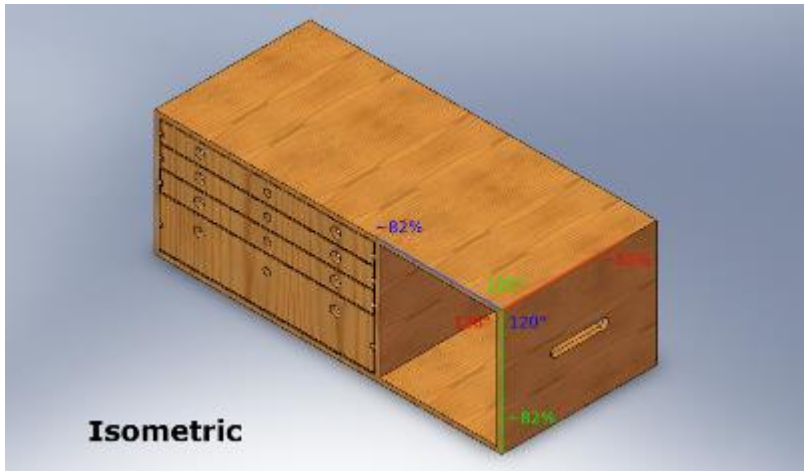




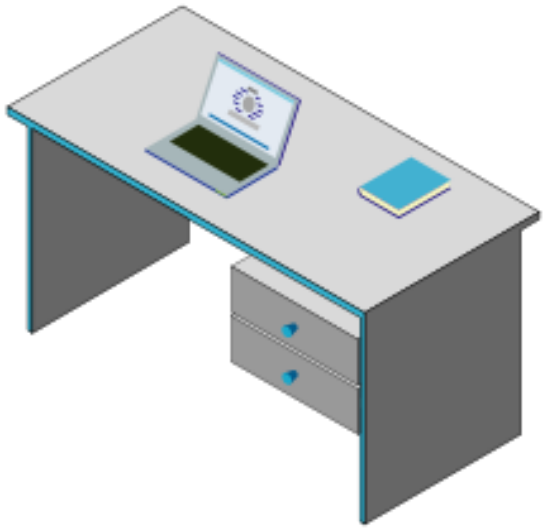


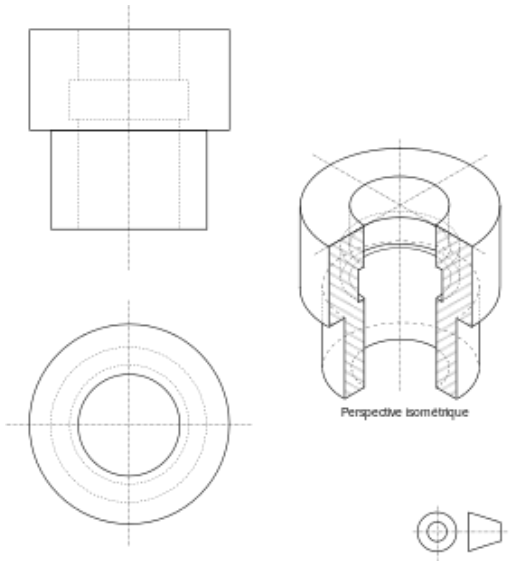


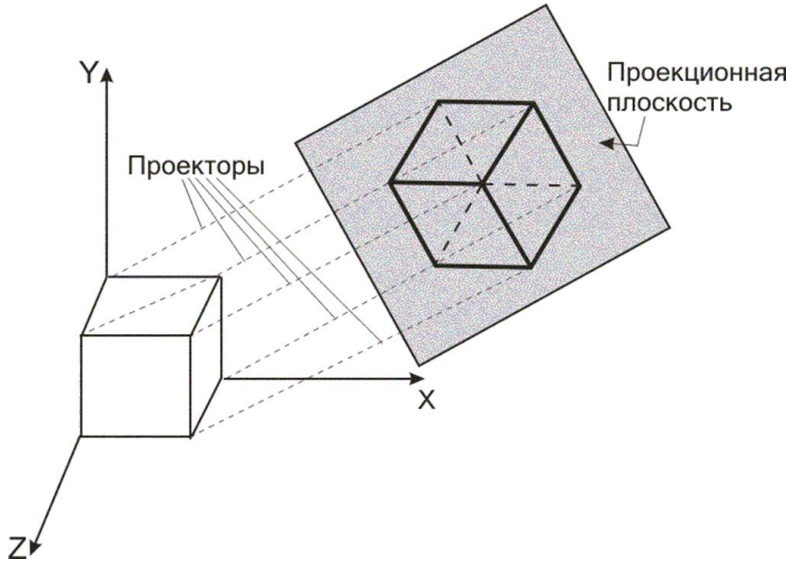




Изометрическая проекция. Применение

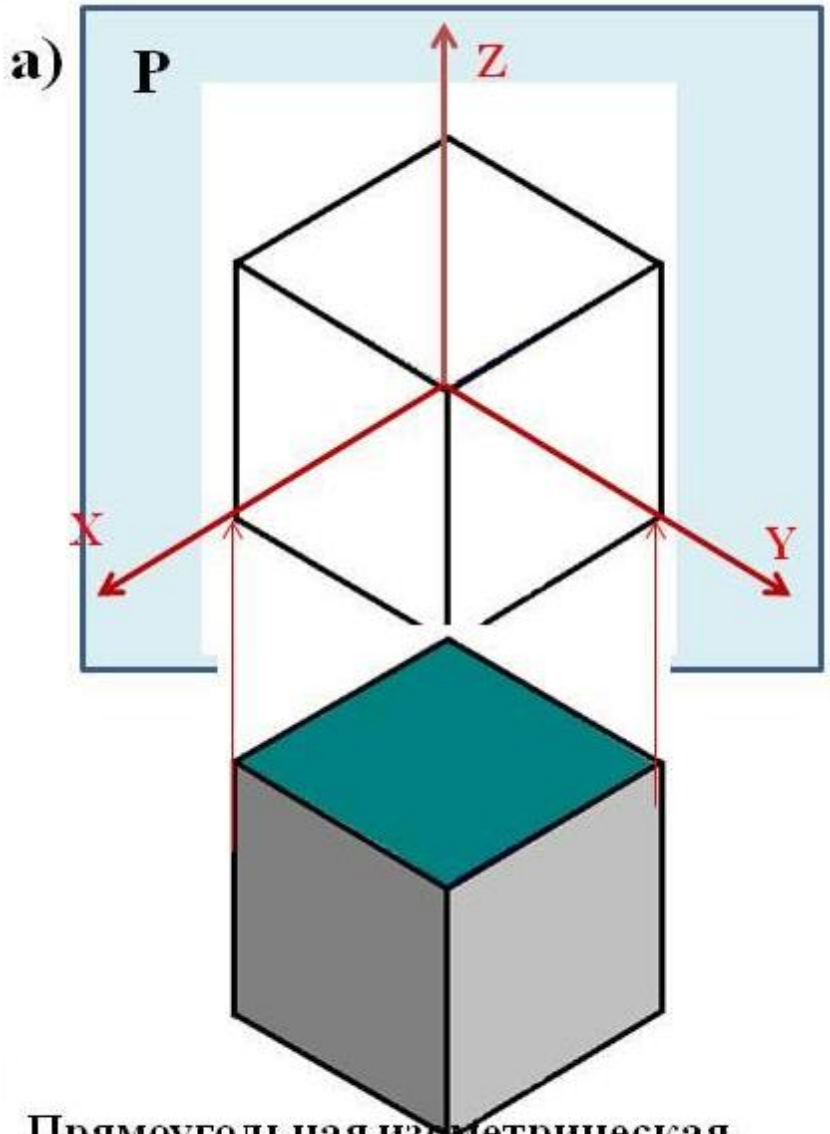






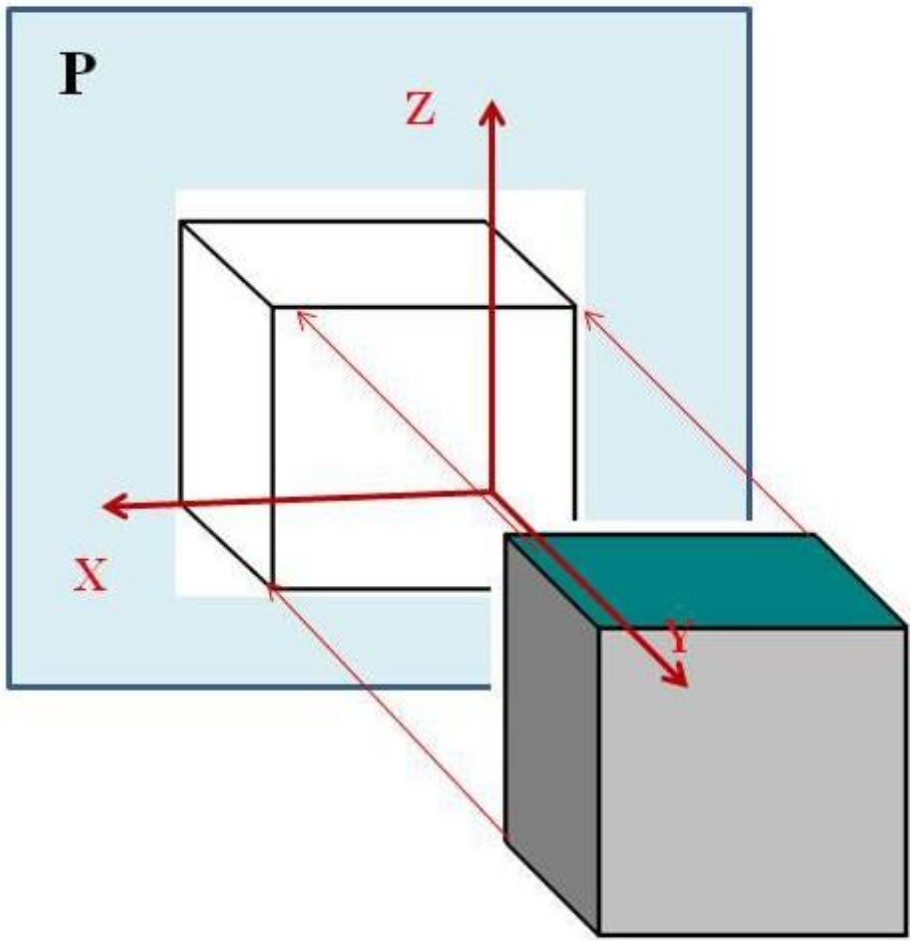
Используется

- в машиностроительном черчении и САПР для построения наглядного изображения детали на чертеже
- а также в компьютерных играх для трёхмерных объектов и панорам

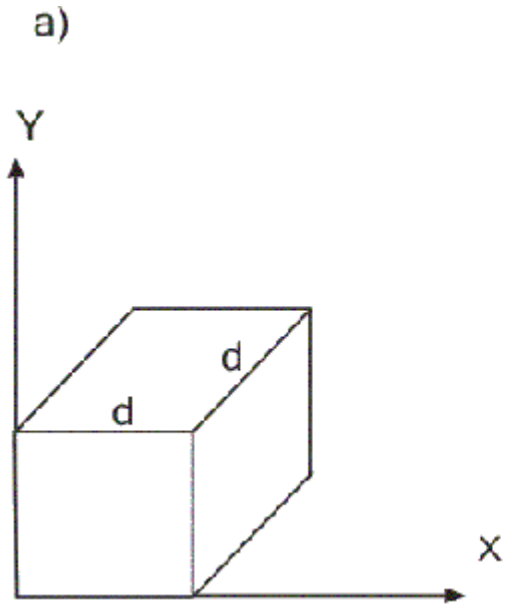


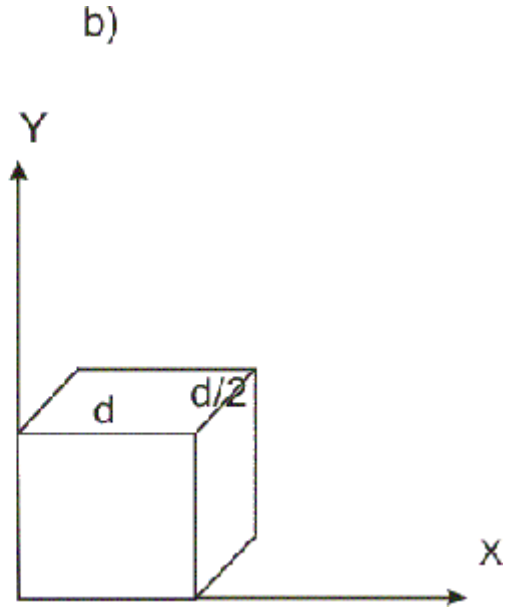
Прямоугольная изометрическая проекция

б) Косоугольная фронтальная диметрическая проекция



Проекции: кавалье (cavalier) и кабине (cabinet)



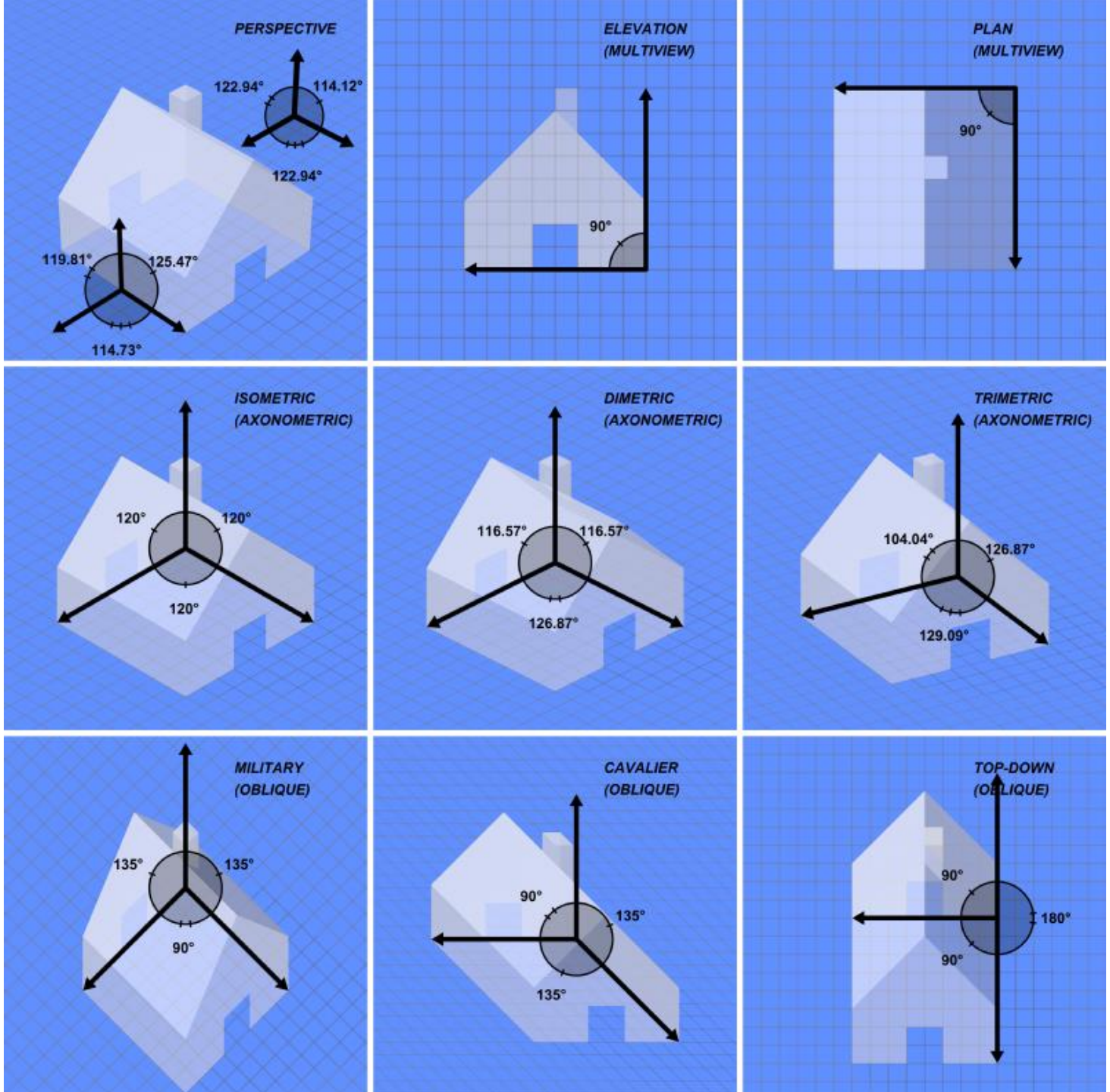


$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \alpha & \beta & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\alpha = \beta = \cos\frac{\pi}{4}$$

$$\alpha = \beta = \frac{1}{2}\cos\frac{\pi}{4}$$

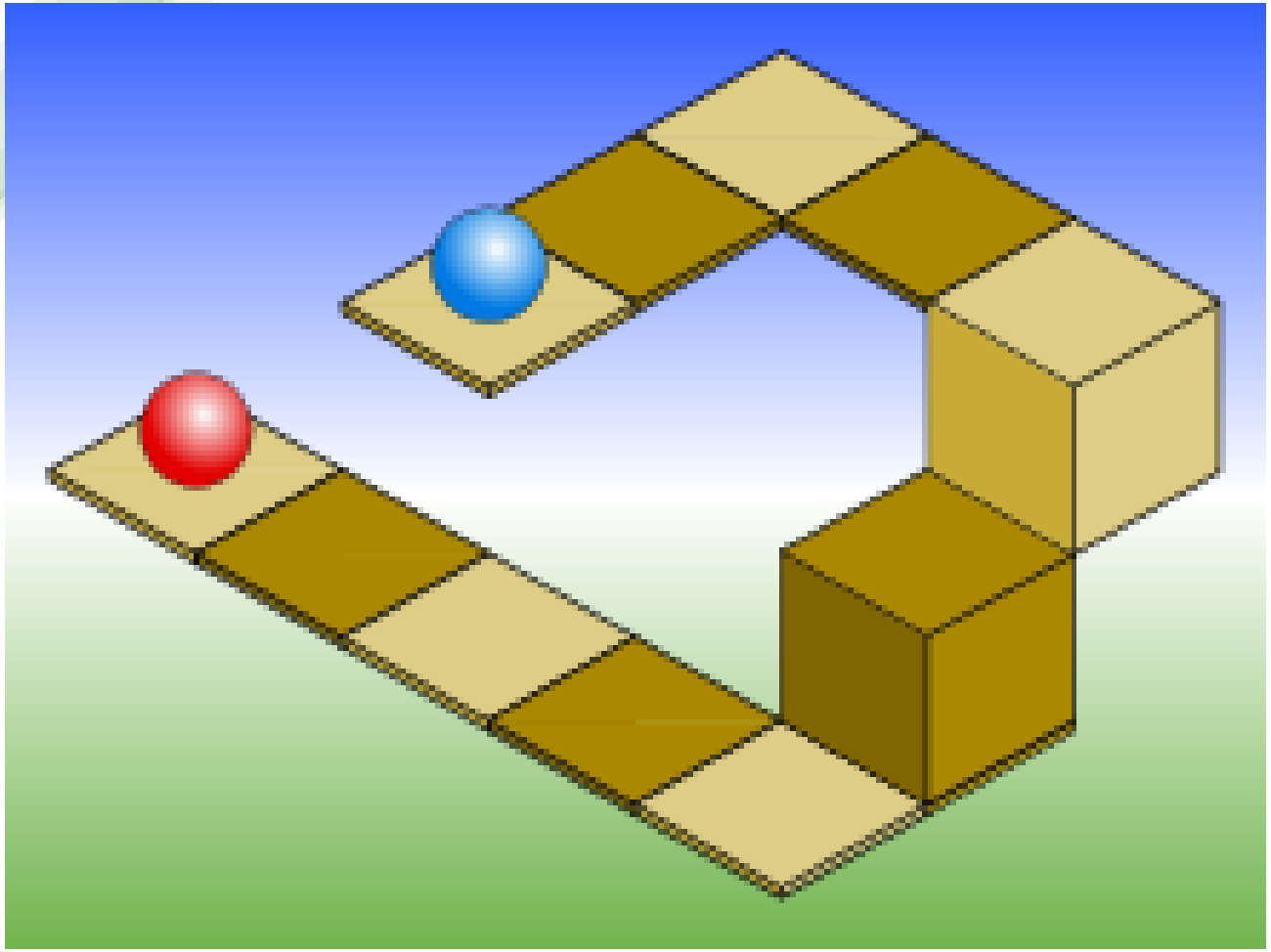
Сравни проекции



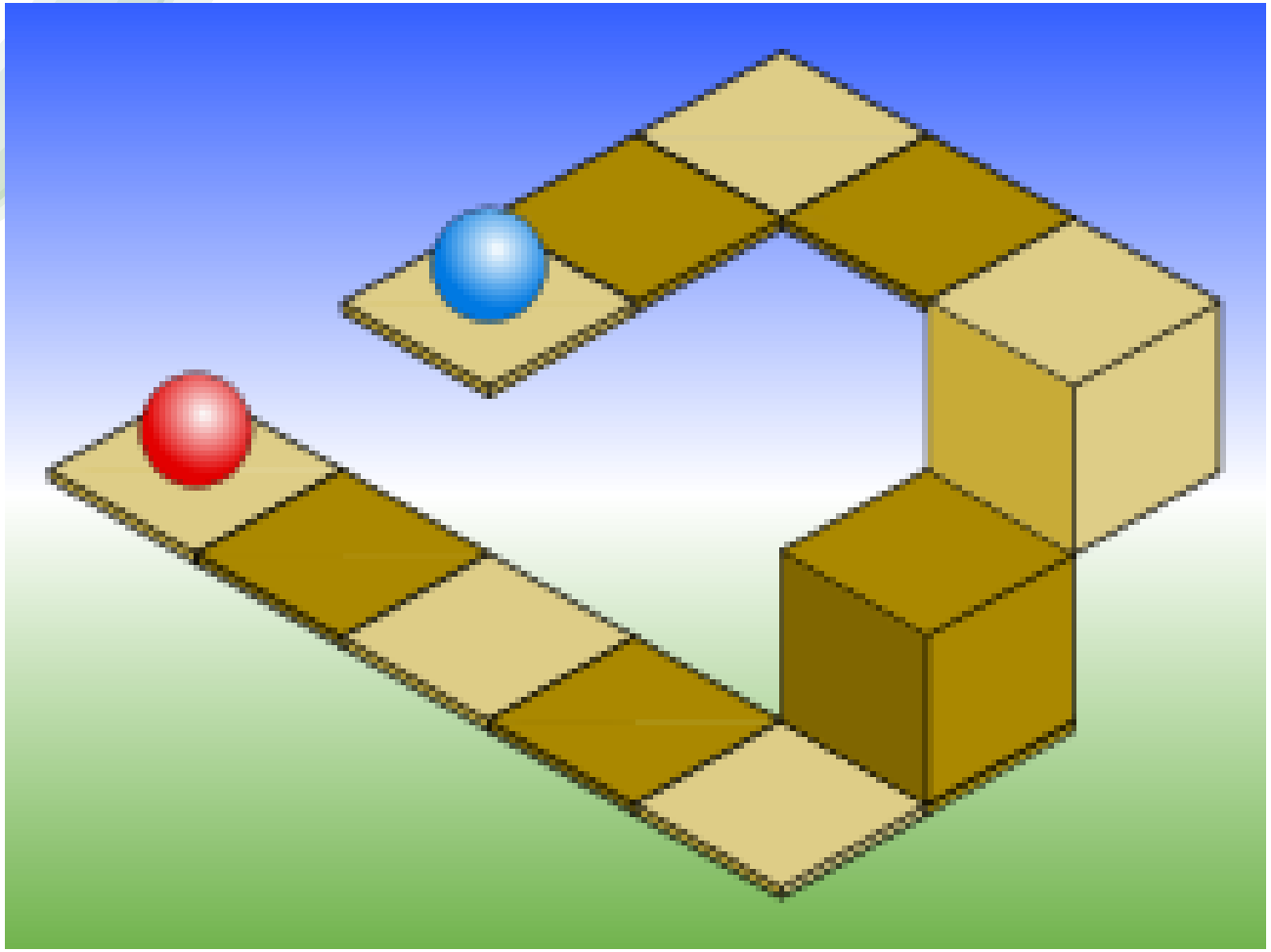
Ограничения аксонометрической проекции

Как и в других видах параллельных проекций, объекты в аксонометрической проекции не выглядят больше или меньше при приближении или удалении от наблюдателя

Это удобно в спрайто-ориентированных компьютерных играх, но, в отличие от перспективной проекции, приводит к ощущению искривления, поскольку человеческий глаз работает иначе



Ограничения аксонометрической проекции: что вы видите?



Ограничения аксонометрической проекции: что на самом деле

Голубой шар на два уровня выше красного



«Водопад» — литография голландского художника Эшера (октябрь 1961)

Что видите?



«Водопад» — парадокс

Изображён парадокс — падающая вода водопада управляет колесом, которое направляет воду на вершину водопада

Водопад имеет структуру «невозможного» треугольника Пенроуза: Водопад на литографии работает как вечный двигатель

Треугольник Пенроуза



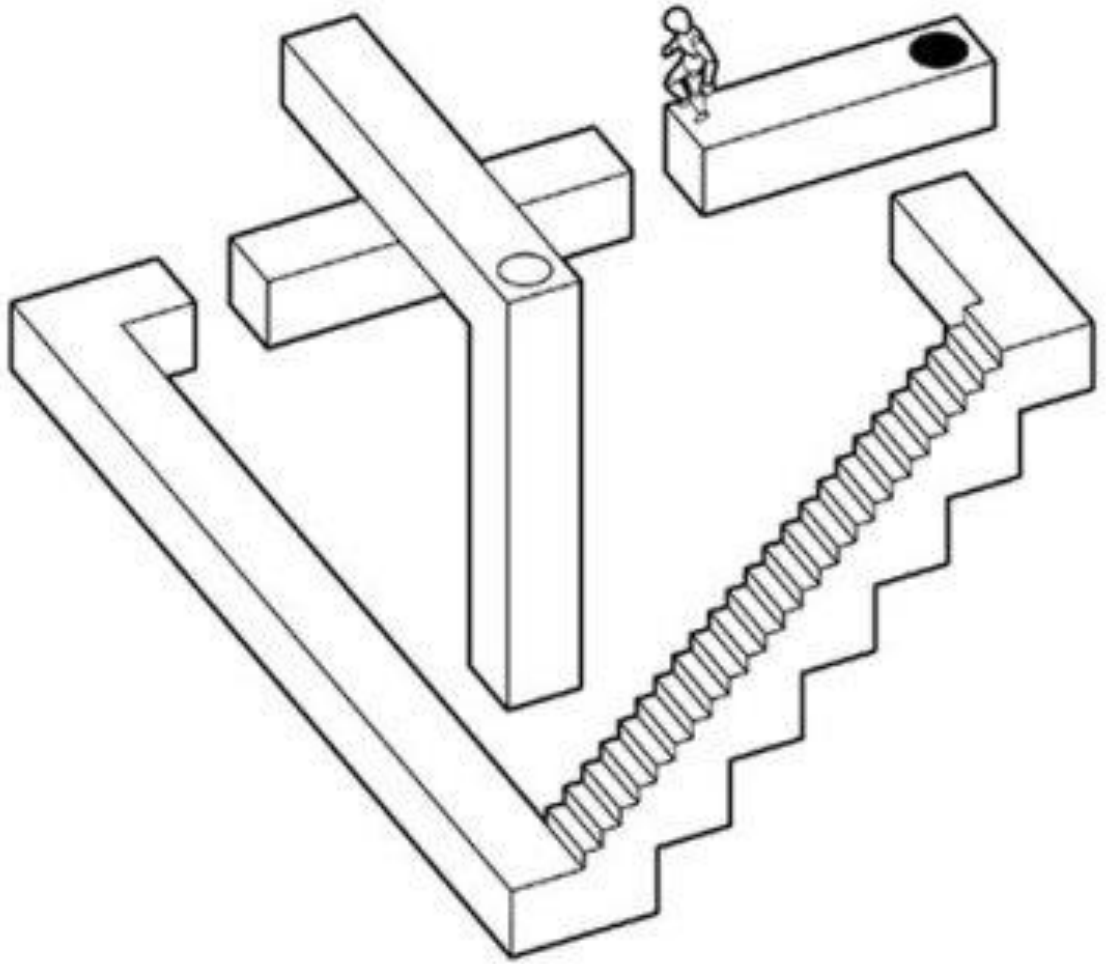
Скульптура, кажущийся треугольник, Немецкий технический музей



Та же скульптура при изменении точки просмотра

Слоган игры — «В этом мире то, что ты видишь, становится реальностью»

Кадр из игры «echochrome»

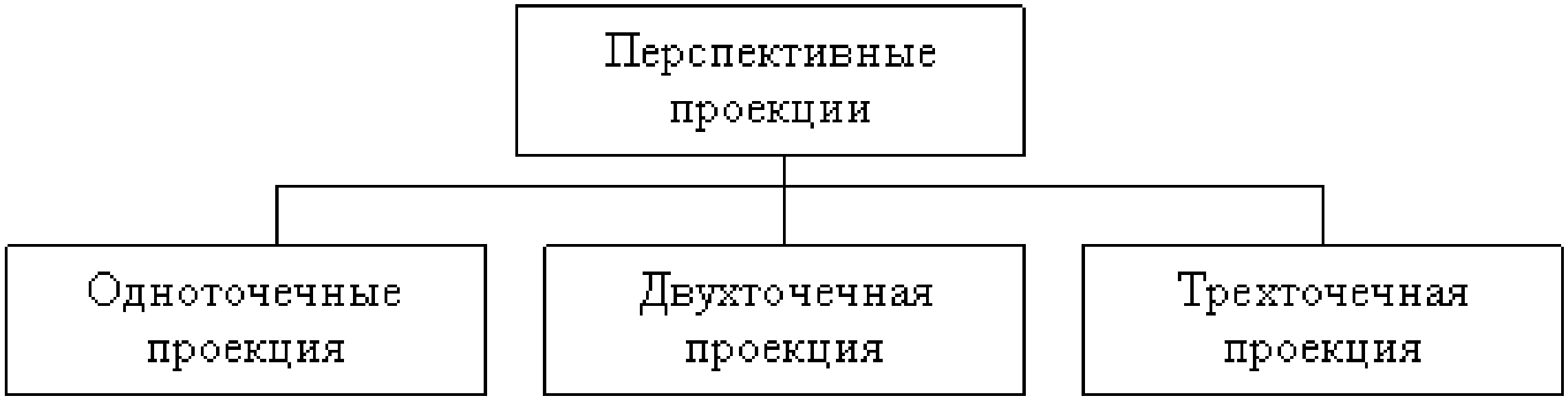


Аркадные игры начала 1980-х

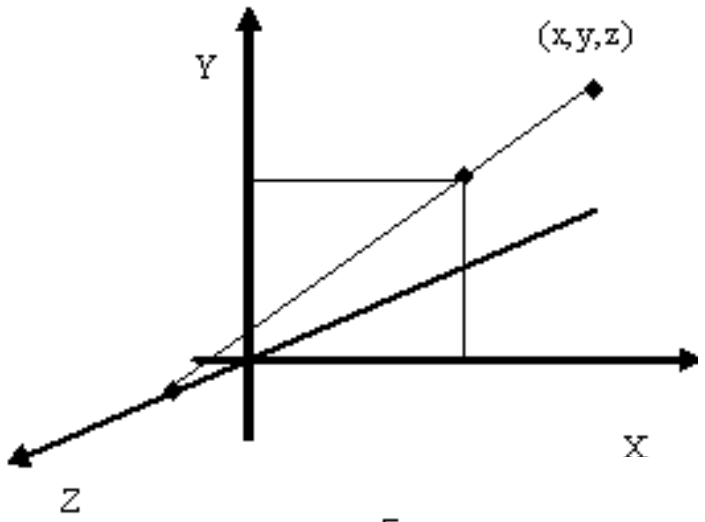


Q*bert (1982) — одна из первых игр с изометрической графикой

Классификация центральных проекций



Центральные проекции



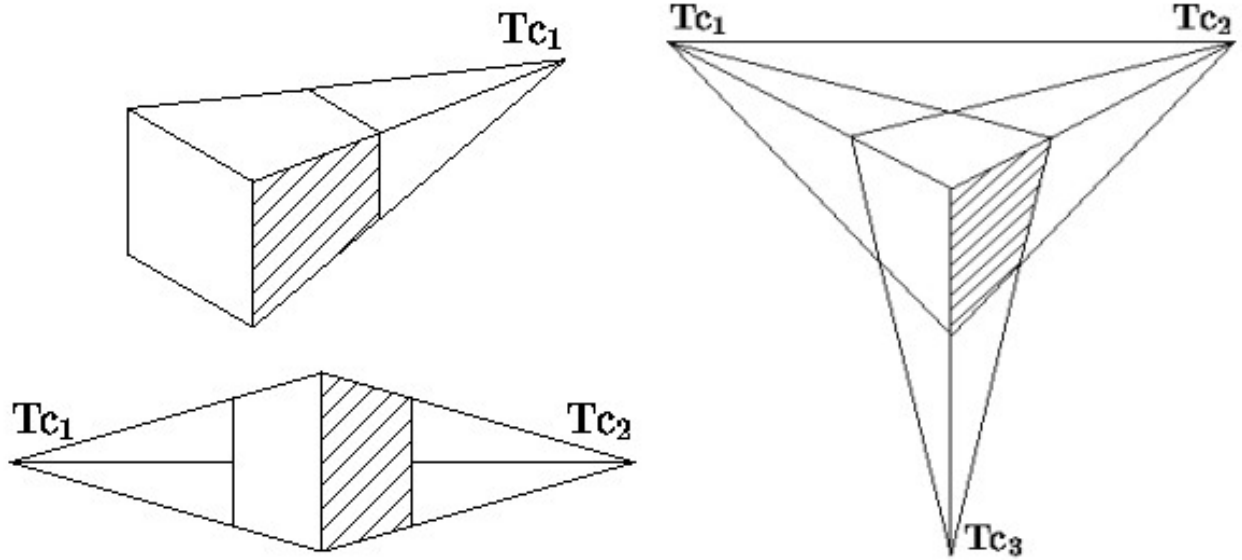
$$x^*=xt,\quad Y^*=yt,\quad Z^*=c+(z-c)\,t$$

$$X^*=\frac{1}{1-\frac{z}{c}}x,\quad Y^*=\frac{1}{1-\frac{z}{c}}y\quad t^*=\frac{1}{1-\frac{z}{c}}$$

$$(x\ y\ z\ 1)\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&-1/c\\0&0&0&1\end{bmatrix}=(x\ y\ 0\ 1-z/c)\qquad \left(\frac{x}{1-\frac{z}{c}}\ \ \frac{y}{1-\frac{z}{c}}\ \ 0\ \ 1\right)$$

Одноточечная, двухточечная и трёхточечная проекции

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1/c \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 0 & -1/a \\ 0 & 1 & 0 & -1/b \\ 0 & 0 & 1 & -1/c \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Итог

- Аффинные преобразования — это язык перемещения, вращения и масштабирования объектов в виртуальном мире
- Однородные координаты — это унифицированный и эффективный способ представить все эти преобразования
- Перспективное проецирование — это «магия», превращающая 3D в 2D, и она целиком основана на «хитрой» матрице и операции деления
- Вместе они образуют математический фундамент всего, что движется и отображается в компьютерной графике

Зачем это в GPU?

Что мы изучали	Где это в GPU
Model Matrix	Вершинный шейдер
View Matrix	Вершинный шейдер
Projection Matrix	Вершинный шейдер
Перспективное деление	Аппаратное деление на w
NDC → экран	Viewport Transform

Что запомнить

- Model → World: Model Matrix
- World → Camera: View Matrix
- Camera → NDC: Projection Matrix
- NDC → Screen: Viewport Transform
- Все матрицы — 4×4, однородные координаты

Проверочные вопросы

- Зачем нужны однородные координаты в 3D?
- Как повернуть объект вокруг произвольной прямой?
- В чём разница между параллельной и центральной проекцией?
- Почему в перспективной проекции объекты уменьшаются с расстоянием?
- Что такое NDC?
- Где в GPU используются матрицы Model/View/Projection?