

Преобразования наблюдения и проецирования

Быковских Дмитрий Александрович

11.10.2025

Преобразования наблюдения и проецирования

Функции из библиотеки glm

- `glm::lookAt(glm::vec3 eye, glm::vec3 center, glm::vec3 up);`
- `glm::ortho(float left, float right, float bottom, float top, float near, float far);`
- `glm::perspective(float fovy, float aspect, float near, float far);`
- `glm::frustum(float left, float right, float bottom, float top, float near, float far);`

2025-10-11

Наблюдение и проецирование

└ Введение

└ Преобразования наблюдения и проецирования

Функции из библиотеки glm

- `glm::lookAt(glm::vec3 eye, glm::vec3 center, glm::vec3 up);`
- `glm::ortho(float left, float right, float bottom, float top, float near, float far);`
- `glm::perspective(float fovy, float aspect, float near, float far);`
- `glm::frustum(float left, float right, float bottom, float top, float near, float far);`

Преобразование наблюдения

```
glm::lookAt(glm::vec3 eye, glm::vec3 center, glm::vec3 up);
```

Аргументы функции

- 1 eye — точка наблюдения;
- 2 center — базовая точка нашей сцены;
- 3 up — вектор верха.

Примечание.

По умолчанию GLM использует правостороннюю систему координат, что соответствует традиционным соглашениям OpenGL, а в DirectX, Vulkan, Metal используется левосторонняя система координат.

2025-10-11

Наблюдение и проецирование

Преобразование наблюдения

Преобразование наблюдения

Преобразование наблюдения

```
glm::lookAt(glm::vec3 eye, glm::vec3 center, glm::vec3 up);
```

Аргументы функции

- eye — точка наблюдения;
- center — базовая точка нашей сцены;
- up — вектор верха.

Примечание.
По умолчанию GLM использует правостороннюю систему координат, что соответствует традиционным соглашениям OpenGL, а в DirectX, Vulkan, Metal используется левосторонняя система координат.

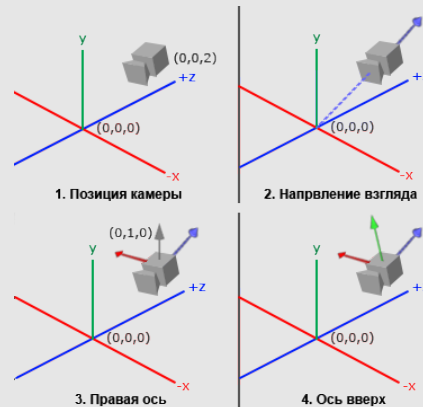


Рис. 1: Этапы преобразования наблюдения

Преобразования наблюдения

$$n = \frac{N}{|N|} = (n_x, n_y, n_z)$$

$$u = \frac{V \times n}{|V|} = (u_x, u_y, u_z)$$

$$v = n \times u = (v_x, v_y, v_z)$$

Примечание.

Векторы являются базисными и поэтому должны быть нормированные.

// 1. Вычисление направление наблюдения;

glm::vec3 zaxis = glm::normalize(center - eye);

// 2. Вычисления направления вправо;
glm::vec3 xaxis = glm::normalize(glm::cross(zaxis, up));

// 3. Определение вектора верха.
glm::vec3 yaxis = glm::cross(xaxis, zaxis);

2025-10-11

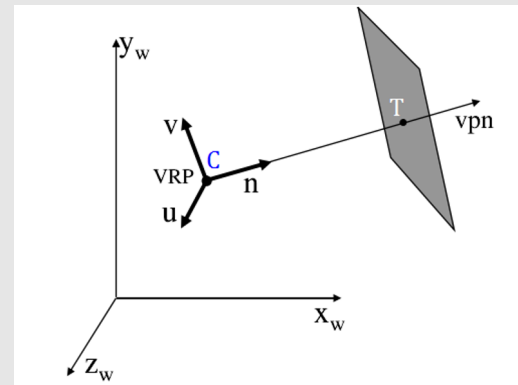
Наблюдение и проецирование
└ Преобразование наблюдения

└ Преобразования наблюдения

Преобразование наблюдения

```
// 1. Вычисление направление
наблюдения;
n = N / |N| = (n_x, n_y, n_z);
glm::vec3 zaxis = glm::normalize(center - eye);
// 2. Вычисления направления вправо;
u = V x n / |V| = (u_x, u_y, u_z);
glm::vec3 xaxis = glm::normalize(glm::cross(zaxis, up));
// 3. Определение вектора верха.
v = n x u = (v_x, v_y, v_z);
glm::vec3 yaxis = glm::cross(xaxis, zaxis);
```

Примечание.
Векторы являются базисными и поэтому должны быть нормированные.



$$\vec{n} = \text{norm}(\vec{T} - \vec{C})$$

$$\vec{v} = \text{norm}(\vec{n} \times (\vec{up} \times \vec{n}))$$

$$\vec{u} = \vec{n} \times \vec{v}$$

Рис. 2: uvp-система

Преобразования наблюдения

$$M_{view} = M_t \times M_{uvn}$$

$$M_{view} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -eye_x & -eye_y & -eye_z & 1 \end{bmatrix} \begin{bmatrix} u_x & v_x & -n_x & 0 \\ u_y & v_y & -n_y & 0 \\ u_z & v_z & -n_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$= \begin{bmatrix} u_x & v_x & -n_x & 0 \\ u_y & v_y & -n_y & 0 \\ u_z & v_z & -n_z & 0 \\ -eye \cdot u & -eye \cdot v & eye \cdot n & 1 \end{bmatrix}$$

Примечание.

Чтобы после применения матрицы взгляда точка перед камерой получила отрицательное значение Z , компоненты вектора n в матрице записывают со знаком минус.

2025-10-11

Наблюдение и проецирование

└ Преобразование наблюдения

└ Преобразования наблюдения

Преобразования наблюдения

$$M_{view} = M_t \times M_{uvn}$$

$$M_{view} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -eye_x & -eye_y & -eye_z & 1 \end{bmatrix} \begin{bmatrix} u_x & v_x & -n_x & 0 \\ u_y & v_y & -n_y & 0 \\ u_z & v_z & -n_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$= \begin{bmatrix} u_x & v_x & -n_x & 0 \\ u_y & v_y & -n_y & 0 \\ u_z & v_z & -n_z & 0 \\ -eye \cdot u & -eye \cdot v & eye \cdot n & 1 \end{bmatrix}$$

Примечание.
Чтобы после применения матрицы взгляда точка перед камерой получила отрицательное значение Z , компоненты вектора n в матрице записывают со знаком минус.

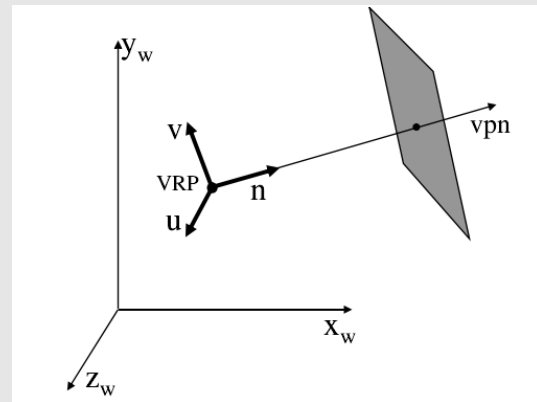


Рис. 3: uvn-система

Ортографическое проецирование

`glm::ortho(float left, float right, float bottom, float top, float near, float far);`

Аргументы функции:

- left, right расположены на оси x
- bottom, top расположены на оси y
- near, far расположены на оси z

Примечание.

Параллельные линии остаются параллельными, но при этом пропадает ощущение глубины, т.е. близко и далеко расположенные объекты относительно камеры имеют одинаковые размеры.

Все, что находится за пределами границы ортографической проекции, отсекается.

2025-10-11

Наблюдение и проецирование

└ Преобразование проецирования

└ Ортографическое проецирование

Ортографическое проецирование

`glm::ortho(float left, float right, float bottom, float top, float near, float far);`

Аргументы функции:

- left, right расположены на оси x
- bottom, top расположены на оси y
- near, far расположены на оси z

Примечание.

Параллельные линии остаются параллельными, но при этом пропадает ощущение глубины, т.е. близко и далеко расположенные объекты относительно камеры имеют одинаковые размеры. Все, что находится за пределами границы ортографической проекции, отсекается.

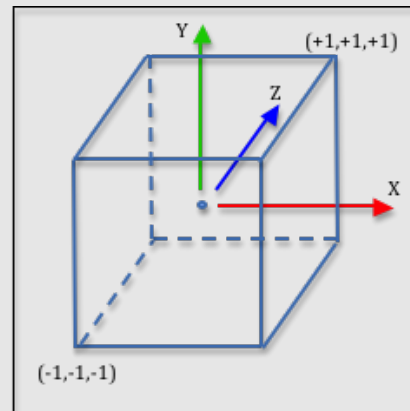


Рис. 4: Граница ортографической проекции (куб)

Ортографическое проецирование

Этапы преобразований

- 1 Центрирование границ ортографической проекции относительно начала системы координат в зависимости от заданных параметров.
- 2 Масштабирование границ ортографической проекции так, чтобы получился куб с размерами сторон, равными 2.
- 3 Изменение направления оси z на противоположное, чтобы она соответствовала системе координат отсекающего пространства.

2025-10-11

Наблюдение и проецирование

└─ Преобразование проецирования

└─ Ортографическое проецирование

Этапы преобразования

- 1 Центрирование границ ортографической проекции относительно начала системы координат в зависимости от заданных параметров.
- 2 Масштабирование границ ортографической проекции так, чтобы получился куб с размерами сторон, равными 2.
- 3 Изменение направления оси z на противоположное, чтобы она соответствовала системе координат отсекающего пространства.

Ортографическое проецирование

1. Центрирование границ ортографической проекции относительно начала системы координат в зависимости от заданных параметров.

$$c_x = \frac{left + right}{2}$$

$$c_y = \frac{bottom + top}{2}$$

$$c_z = \frac{near + far}{2}$$

$$M_t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -c_x & -c_y & -c_z & 1 \end{bmatrix}$$

2025-10-11

Наблюдение и проецирование

└ Преобразование проецирования

└ Ортографическое проецирование

Ортографическое проецирование

1. Центрирование границ ортографической проекции относительно начала системы координат в зависимости от заданных параметров.

$$c_x = \frac{left + right}{2}$$

$$c_y = \frac{bottom + top}{2}$$

$$c_z = \frac{near + far}{2}$$

$$M_t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -c_x & -c_y & -c_z & 1 \end{bmatrix}$$

Параметрическое уравнение прямой в пространстве

$$\begin{cases} x = x_l + (x_r - x_l)t \\ y = y_b + (y_t - y_b)t \\ z = z_n + (z_f - z_n)t \end{cases}$$

или

$$\begin{cases} x = (1 - t)x_l + x_r t \\ y = (1 - t)y_b + y_t t \\ z = (1 - t)z_n + z_f t \end{cases}$$

Ортографическое проецирование

2. Масштабирование границ ортографической проекции так, чтобы получился куб с размерами сторон, равными 2.

$$s_x = \frac{2}{right - left}$$

$$s_y = \frac{2}{top - bottom}$$

$$s_z = \frac{2}{far - near}$$

$$M_s = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2025-10-11

Наблюдение и проецирование

└ Преобразование проецирования

└ Ортографическое проецирование

Чтобы значения были в диапазоне от -1 до 1

$$s_x = \frac{2}{right - left}$$

$$s_y = \frac{2}{top - bottom}$$

$$s_z = \frac{2}{far - near}$$

$$M_s = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ортографическое проецирование

3. Изменение направления оси z на противоположное, чтобы она соответствовала системе координат отсекающего пространства.

$$M_{lh} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2025-10-11

Наблюдение и проецирование

└ Преобразование проецирования

└ Ортографическое проецирование

Левосторонняя система координат (left hand)
inversion z-axis

3. Изменение направления оси z на противоположное, чтобы она соответствовала системе координат отсекающего пространства.

$$M_{lh} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ортографическое проецирование

$$M_t M_s M_{lh} =$$

$$= \begin{bmatrix} \frac{2}{right-left} & 0 & 0 & 0 \\ 0 & \frac{2}{top-bottom} & 0 & 0 \\ 0 & 0 & -\frac{2}{far-near} & 0 \\ -\frac{left+right}{right-left} & -\frac{bottom+top}{top-bottom} & -\frac{near+far}{far-near} & 1 \end{bmatrix}$$

2025-10-11

Наблюдение и проецирование

└ Преобразование проецирования

└ Ортографическое проецирование

$$M_t M_s M_{lh} =$$

$$= \begin{bmatrix} \frac{2}{right-left} & 0 & 0 & 0 \\ 0 & \frac{2}{top-bottom} & 0 & 0 \\ 0 & 0 & -\frac{2}{far-near} & 0 \\ -\frac{left+right}{right-left} & -\frac{bottom+top}{top-bottom} & -\frac{near+far}{far-near} & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\frac{left+right}{2} & -\frac{bottom+top}{2} & -\frac{near+far}{2} & 1 \end{bmatrix} \cdot$$

$$\begin{bmatrix} \frac{2}{right-left} & 0 & 0 & 0 \\ 0 & \frac{2}{top-bottom} & 0 & 0 \\ 0 & 0 & \frac{2}{far-near} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

Перспективное проецирование

```
glm::perspective(float fovy, float aspect, float near, float far);
glm::frustum(float left, float right, float bottom, float top, float near, float far);
```

Примечание.

Параллельные линии сходятся в точки схода (фокуса), создавая ощущение глубины:

- Ближе к камере объекты кажутся больше, а дальше – меньше, пропорционально обратному расстоянию до камеры.
- Всё, что находится за пределами симметричного усечённого пирамидального объёма (фрустума), отсекается.
- Ближайшая и дальняя плоскости (near и far) задают диапазон глубины.

2025-10-11

Наблюдение и проецирование

- Преобразование проецирования

- Перспективное проецирование

Перспективное проецирование

```
glm::perspective(float fovy, float aspect, float near, float far);
glm::frustum(float left, float right, float bottom, float top, float near, float far);
```

Примечание.
Параллельные линии сходятся в точки схода (фокуса), создавая ощущение глубины:

- Ближе к камере объекты кажутся больше, а дальше – меньше, пропорционально обратному расстоянию до камеры.
- Всё, что находится за пределами симметричного усечённого пирамидального объёма (фрустума), отсекается.
- Ближайшая и дальняя плоскости (near и far) задают диапазон глубины.

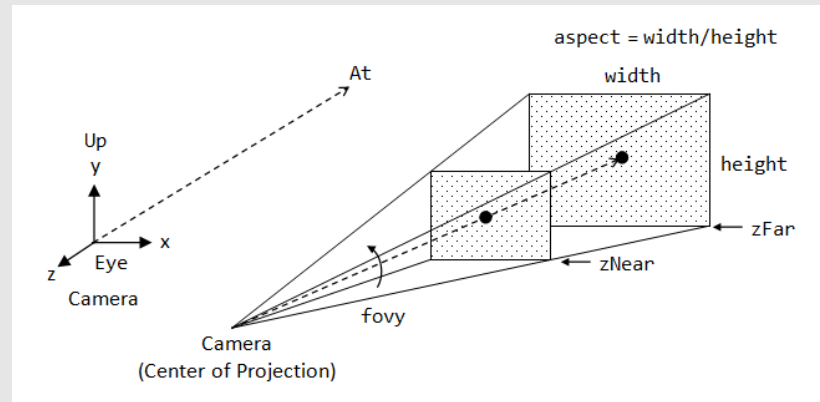


Рис. 5: Перспективная проекция

Перспективное проецирование

связь функций `perspective` и `frustum`

```
glm::perspective(float fovy, float aspect, float near, float far);
glm::frustum(float left, float right, float bottom, float top, float near, float far);
```

Эти функции делают одно и тоже, но различаются аргументами.

Переход от `glm::frustum` к `glm::perspective` имеет вид:

Первый аргумент `fovy` (field of view in y) обозначает «поле обзора по вертикали» (или угол обзора) и вычисляется как:

$$fovy = 2 \cdot \arctan(top/near);$$

Второй аргумент `aspect` (или `aspectratio`) обозначает «соотношение сторон» (отношение высоты к ширине, например, 16:9 или 4:3) и вычисляется как:

$$aspect = right/top$$

2025-10-11

Наблюдение и проецирование

Преобразование проецирования

Перспективное проецирование

В дальнейшем же будет применяться обратные преобразования, а именно:

$$top = near \cdot \tan\left(\frac{fovy}{2}\right)$$

$$bottom = -top$$

$$right = top \cdot aspect$$

$$left = -right$$

Перспективное проецирование
связь функций `perspective` и `frustum`

```
glm::perspective(float fovy, float aspect, float near, float far);
glm::frustum(float left, float right, float bottom, float top, float near, float far);
```

Эти функции делают одно и тоже, но различаются аргументами.

Переход от `glm::frustum` к `glm::perspective` имеет вид:

Первый аргумент `fovy` (field of view in y) обозначает «поле обзора по вертикали» (или угол обзора) и вычисляется как:

$$fovy = 2 \cdot \arctan(top/near);$$

Второй аргумент `aspect` (или `aspectratio`) обозначает «соотношение сторон» (отношение высоты к ширине, например, 16:9 или 4:3) и вычисляется как:

$$aspect = right/top$$

Перспективное проецирование

Этапы преобразований

- 1 Центрирование границ усеченной пирамиды относительно начала двумерной системы координат XoY .
- 2 Масштабирование значения глубины z в нормализованный диапазон $(-1, +1)$.
- 3 Расчет перспективы.
- 4 Масштабирование двумерных величин (x', y') к квадратной области размером $[-1, 1]^2$

2025-10-11

Наблюдение и проецирование

└ Преобразование проецирования

└ Перспективное проецирование

Этапы преобразований

- 1 Центрирование границ усеченной пирамиды относительно начала двумерной системы координат XoY .
- 2 Масштабирование значения глубины z в нормализованный диапазон $(-1, +1)$.
- 3 Расчет перспективы.
- 4 Масштабирование двумерных величин (x', y') к квадратной области размером $[-1, 1]^2$

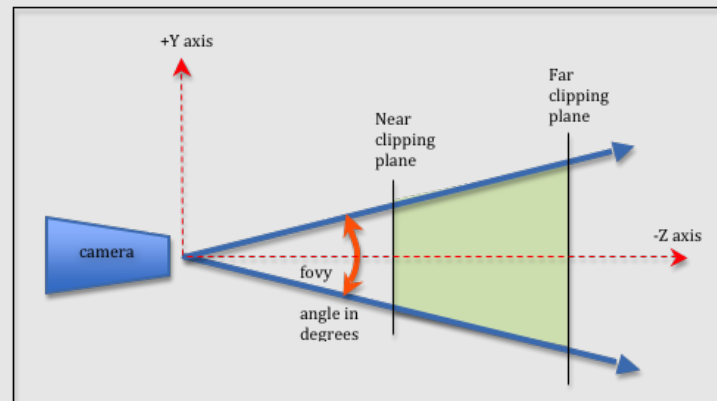


Рис. 6: Перспективная проекция

Перспективное проецирование

1. Центрирование границ усеченной пирамиды относительно начала двумерной системы координат XoY .

$$sh_{xz} = \frac{left + right}{2near}$$

$$sh_{yz} = \frac{bottom + top}{2near}$$

$$M_t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ sh_{xz} & sh_{yz} & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2025-10-11

Наблюдение и проецирование

Преобразование проецирования

Перспективное проецирование

1. Смещение координат вершины относительно начала двумерной системы координат XoY исходя из размеров усеченной пирамиды.

$$x' = x - \frac{left + right}{2}$$

$$y' = y - \frac{bottom + top}{2}$$

2. Смещение границ координат вершины относительно начала двумерной системы координат XoY исходя из размеров усеченной пирамиды и значения глубины z .

$$x' = x - \frac{left + right}{2} \frac{-z}{near}$$

$$y' = y - \frac{bottom + top}{2} \frac{-z}{near}$$

1. Центрирование границ усеченной пирамиды относительно начала двумерной системы координат XoY .

$$sh_{xz} = \frac{left + right}{2near}$$

$$sh_{yz} = \frac{bottom + top}{2near}$$

$$M_t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ sh_{xz} & sh_{yz} & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Перспективное проецирование

2. Масштабирование значения глубины z в нормализованный диапазон $(-1, +1)$

В основе лежит нелинейное уравнение (модифицированное уравнение гиперболы) и имеет вид:

$$z' = \frac{c_1}{-z} + c_2,$$

где c_1 и c_2 — константы, которые вычисляются на основе диапазона $(-near, -far)$.

Примечание.

Когда $z = -near$, уравнение должно давать -1 .

Когда $z = -far$, уравнение должно давать $+1$.

Что в итоге дает нам два уравнения для решения c_1 и c_2 .

2025-10-11

Наблюдение и проецирование

- Преобразование проецирования

- Перспективное проецирование

Перспективное проецирование

2. Масштабирование значения глубины z в нормализованный диапазон $(-1, +1)$

В основе лежит нелинейное уравнение (модифицированное уравнение гиперболы) и имеет вид:

$$z' = \frac{c_1}{-z} + c_2,$$

где c_1 и c_2 — константы, которые вычисляются на основе диапазона $(-near, -far)$.

Примечание.

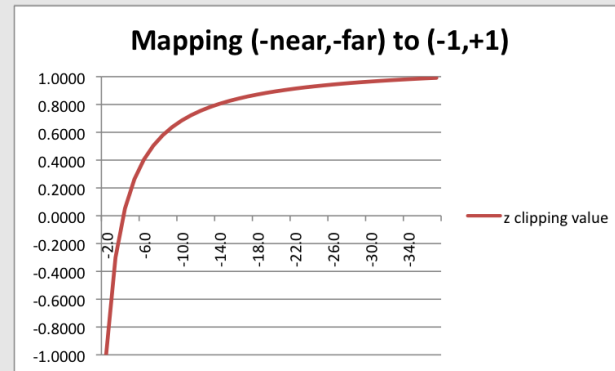
Когда $z = -near$, уравнение должно давать -1 .Когда $z = -far$, уравнение должно давать $+1$.Что в итоге дает нам два уравнения для решения c_1 и c_2 .

Рис. 7: Кривая нелинейного уравнения

Перспективное проецирование

В результате получаем следующую систему:

$$\begin{cases} -1 = c_1 / -(-near) + c_2 \\ +1 = c_1 / -(-far) + c_2 \end{cases}$$

Т.е. получили СЛАУ, где c_1 и c_2 теперь неизвестные.

Зная, чему равняются $near$ и far в каждом частном случае, найдем константы c_1 и c_2 .

$$\begin{cases} c_1 = 2far \cdot near / (near - far) \\ c_2 = (far + near) / (far - near) \end{cases}$$

Но куда же их подставлять в нашу матрицу?

2025-10-11

Наблюдение и проецирование

Преобразование проецирования

Перспективное проецирование

Нельзя просто отбросить значение z , так как оно указывает на расстояние между вершиной и камерой, что позволяет нам определить, какие объекты находятся перед другими. Мы могли бы сделать линейное отображение между диапазоном $(-near, -far)$ и $(-1, +1)$. Однако числа с плавающей запятой подвержены погрешностям округления при выполнении математических операций. В графических приложениях иногда разница между 0.1234568 и 0.1234567 может оказать визуальное воздействие на рендеринг. Мы хотели бы использовать большую точность для значений, близких к камере, и меньшую точность для вершин, находящихся дальше от камеры. Это означает, что нам нужно нелинейное отображение между $(-near, -far)$ и $(-1, +1)$.

В результате получаем следующую систему:

$$\begin{cases} -1 = c_1 / -(-near) + c_2 \\ +1 = c_1 / -(-far) + c_2 \end{cases}$$

Т.е. получили СЛАУ, где c_1 и c_2 теперь неизвестны. Зная, чему равняются $near$ и far в каждом частном случае, найдем константы c_1 и c_2 .

$$\begin{cases} c_1 = 2far \cdot near / (near - far) \\ c_2 = (far + near) / (far - near) \end{cases}$$

Но куда же их подставлять в нашу матрицу?

Перспективное проецирование

Рассмотрим правую часть нелинейного уравнения и вынесем знаменатель $-z$ за скобки:

$$z' = \frac{1}{-z}(-c_2 \cdot z + c_1)$$

Используем матричные преобразования, чтобы перейти от конкретного значения z к модифицированному значению z' :

перемещение, т.е. прибавление c_1 ,
масштабирование, т.е. умножение на $-c_2$,
проецирование, т.е. деление $-z$.

Тогда получим следующую матрицу.

$$M_{sz} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -c_2 & -1 \\ 0 & 0 & c_1 & 0 \end{bmatrix}$$

2025-10-11

Наблюдение и проецирование

Преобразование проецирования

Перспективное проецирование

Перспективное проецирование

Рассмотрим правую часть нелинейного уравнения и вынесем знаменатель $-z$ за скобки:

$$z' = \frac{1}{-z}(-c_2 \cdot z + c_1)$$

Используем матричные преобразования, чтобы перейти от конкретного значения z к модифицированному значению z' :
перемещение, т.е. прибавление c_1 ,
масштабирование, т.е. умножение на $-c_2$,
проецирование, т.е. деление $-z$.
Тогда получим следующую матрицу:

$$M_{sz} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -c_2 & -1 \\ 0 & 0 & c_1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix}^T \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -c_2 & -1 \\ 0 & 0 & c_1 & 0 \end{bmatrix} = \begin{bmatrix} p_x \\ p_y \\ -c_2 p_z + c_1 \\ -z \end{bmatrix}^T$$

$$\begin{bmatrix} p_x^* \\ p_y^* \\ p_z^* \\ 1 \end{bmatrix}^T = \begin{bmatrix} p_x / (-z) \\ p_y / (-z) \\ (-c_2 p_z + c_1) / (-z) \\ 1 \end{bmatrix}^T$$

Перспективное проецирование

3. Расчет перспективы

Пусть (x, y, z) — координаты вершины.

Задача: отобразить на 2D окне просмотра.

Спроецируем вершину на ближнюю плоскость окна просмотра, т.е. нужно перейти от (x, y, z) к $(x', y', near)$.

Интерпретация (см. схему).

Здесь $near$ — значение, представляющее ближнюю плоскость отсечения. Значения x , y и z различаются для каждой вершины в сцене и представляют ее трехмерные координаты. Результатом этих вычислений будут координаты $(x', y', near)$, которые представляют положение вершины на ближней плоскости окна просмотра.

2025-10-11

Наблюдение и проецирование

- Преобразование проецирования

- Перспективное проецирование

Перспективное проецирование

3. Расчет перспективы

Пусть (x, y, z) — координаты вершины.
Задача: отобразить на 2D окне просмотра.

Спроецируем вершину на ближнюю плоскость окна просмотра, т.е. нужно перейти от (x, y, z) к $(x', y', near)$.

Интерпретация (см. схему).

Здесь $near$ — значение, представляющее ближнюю плоскость отсечения. Значения x , y и z различаются для каждой вершины в сцене и представляют ее трехмерные координаты. Результатом этих вычислений будут координаты $(x', y', near)$, которые представляют положение вершины на ближней плоскости окна просмотра.

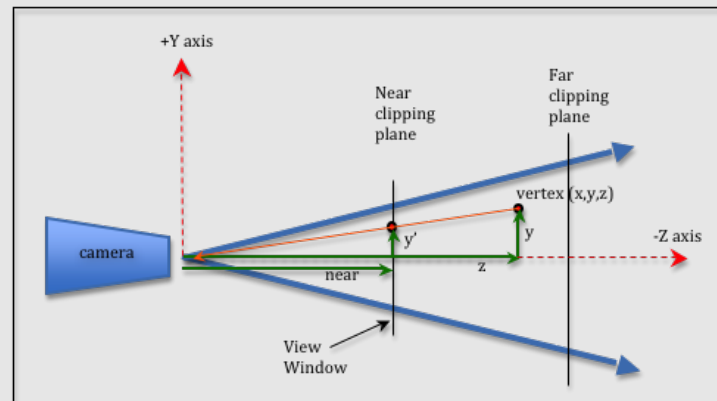


Рис. 8: Схема смещения значений

Перспективное проецирование

Тогда матрица будет иметь вид:

$$M_p = \begin{bmatrix} near & 0 & 0 & 0 \\ 0 & near & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2025-10-11

Наблюдение и проецирование

└ Преобразование проецирования

└ Перспективное проецирование

Тогда матрица будет иметь вид:

$$M_p = \begin{bmatrix} near & 0 & 0 & 0 \\ 0 & near & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Тогда можно составить следующие пропорции:

$$\frac{x'}{x} = \frac{near}{-z} \quad \frac{y'}{y} = \frac{near}{-z}$$

В результате получаем следующие формулы:

$$x' = \frac{near}{-z} x \quad y' = \frac{near}{-z} y$$

Учитывая предыдущий шаг, получается, что не хватает лишь множителя *near*.

Перспективное проецирование

4. Масштабирование двумерных величин (x', y') к квадратной области размером $[-1, 1]^2$

$$s_x = \frac{2}{right - left}$$

$$s_y = \frac{2}{top - bottom}$$

$$M_{s2d} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2025-10-11

Наблюдение и проецирование

└ Преобразование проецирования

└ Перспективное проецирование

4. Масштабирование двумерных величин (x', y') к квадратной области размером $[-1, 1]^2$

$$s_x = \frac{2}{right - left}$$

$$s_y = \frac{2}{top - bottom}$$

$$M_{s2d} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Перспективное проектирование

2025-10-11

Наблюдение и проецирование

└ Преобразование проецирования

└ Перспективное проектирование

$$M_t M_{sz} M_p M_{s2d} = \begin{bmatrix} \frac{2near}{right-left} & 0 & 0 & 0 \\ 0 & \frac{2near}{top-bottom} & 0 & 0 \\ \frac{right+left}{right-left} & \frac{top+bottom}{top-bottom} & -\frac{far+near}{far-near} & -1 \\ 0 & 0 & \frac{2far \cdot near}{near-far} & 0 \end{bmatrix}$$

$$M_t M_{sz} M_p M_{s2d} = \begin{bmatrix} \frac{2near}{right-left} & 0 & 0 & 0 \\ 0 & \frac{2near}{top-bottom} & 0 & 0 \\ \frac{right+left}{right-left} & \frac{top+bottom}{top-bottom} & -\frac{far+near}{far-near} & -1 \\ 0 & 0 & \frac{2near \cdot far}{near-far} & 0 \end{bmatrix}$$

Промежуточные расчеты

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \frac{left+right}{2near} & \frac{bottom+top}{2near} & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{far+near}{far-near} & -1 \\ 0 & 0 & \frac{2far \cdot near}{near-far} & 0 \end{bmatrix} \cdot \begin{bmatrix} near & 0 & 0 & 0 \\ 0 & near & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{2}{right-left} & 0 & 0 & 0 \\ 0 & \frac{2}{top-bottom} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$