

Преобразования наблюдения и проектирования

Быковских Дмитрий Александрович

28.10.2023

2023-10-28

Наблюдение и проектирование

Преобразования наблюдения и проектирования

Быковских Дмитрий Александрович

28.10.2023

Конвейер трехмерного преобразования

Преобразование моделирования	Модельные координаты (МК)
Преобразование наблюдения	Внешние координаты (ВК)
Преобразование проектирования	Координаты наблюдения (КН)
Преобразование нормировки и отсечение	Координаты проекции (КП)
Преобразование поля просмотра	Нормированные координаты (НК)
	Координаты устройства (КУ)

2023-10-28

Наблюдение и проектирование

└ Введение

└ Конвейер трехмерного преобразования

	Модельные координаты (МК)
Преобразование моделирования	Внешние координаты (ВК)
Преобразование наблюдения	Координаты наблюдения (КН)
Преобразование проектирования	Координаты проекции (КП)
Преобразование нормировки и отсечение	Нормированные координаты (НК)
Преобразование поля просмотра	Координаты устройства (КУ)

Преобразования наблюдения и проектирования

Функции из библиотеки glm

- `glm::lookAt(glm::vec3 position, glm::vec3 target, 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);`

2023-10-28

Наблюдение и проектирование

└ Введение

└ Преобразования наблюдения и проектирования

Функции из библиотеки glm

- `glm::lookAt(glm::vec3 position, glm::vec3 target, 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 position, glm::vec3 target, glm::vec3 up);
```

Аргументы функции:

- 1 Точка наблюдения position;
- 2 Базовая точка нашей сцены target;
- 3 Вектор верха up.

2023-10-28

Наблюдение и проектирование
└─ Преобразование наблюдения

└─ Преобразование наблюдения

```
glm::lookAt(glm::vec3 position, glm::vec3 target, glm::vec3 up);
```

Аргументы функции:

- 1 Точка наблюдения position;
- 2 Базовая точка нашей сцены target;
- 3 Вектор верха up.

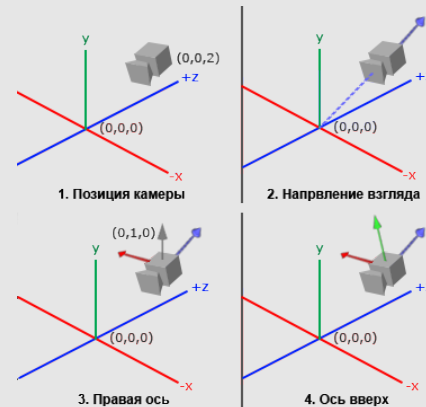


Рис. 1: Этапы преобразования наблюдения

Преобразования наблюдения

```
// 1. Вычисление направление
наблюдения;
glm::vec3 zaxis = glm::normalize(position -
target);
// 2. Вычисления направления вправо;
glm::vec3 xaxis =
glm::normalize(glm::cross(glm::normalize(up),
zaxis));
// 3. Определение вектора верха.
glm::vec3 yaxis = glm::cross(zaxis, xaxis);
```

Примечание.

Векторы являются базисными и поэтому должны быть нормированные.

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

$$u = \frac{V \times n}{|V|} = (v_x, v_y, v_z)$$

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

2023-10-28

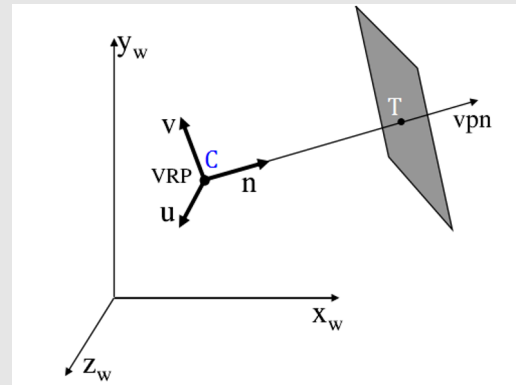
Наблюдение и проектирование

Преобразование наблюдения

Преобразования наблюдения

Преобразования наблюдения

```
// 1. Вычисление направление
наблюдения;
glm::vec3 zaxis = glm::normalize(position -
target);
// 2. Вычисления направления вправо;
glm::vec3 xaxis =
glm::normalize(glm::cross(glm::normalize(up),
zaxis));
// 3. Определение вектора верха.
glm::vec3 yaxis = glm::cross(zaxis, xaxis);
Примечание.
Векторы являются базисными и поэтому должны быть
нормированные.
```



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

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

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

Рис. 2: uvpn-система

Преобразования наблюдения

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

$$M_{view} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -p_x & -p_y & -p_z & 1 \end{bmatrix} \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ n_x & n_y & n_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$= \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ n_x & n_y & n_z & 0 \\ -p_x & -p_y & -p_z & 1 \end{bmatrix}$$

2023-10-28

Наблюдение и проектирование
└ Преобразование наблюдения

└ Преобразования наблюдения

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

$$M_{uvn} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -p_x & -p_y & -p_z & 1 \end{bmatrix} \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ n_x & n_y & n_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$= \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ n_x & n_y & n_z & 0 \\ -p_x & -p_y & -p_z & 1 \end{bmatrix}$$

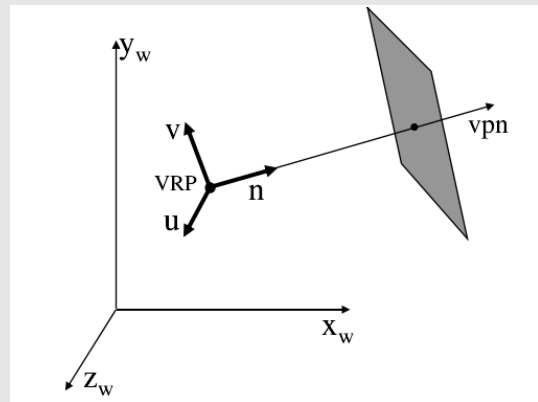


Рис. 3: uvn-система

Ортографическое проектирование

`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.

2023-10-28

Наблюдение и проектирование

└ Преобразование проектирования

└ Ортографическое проектирование

Ортографическое проектирование

`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.

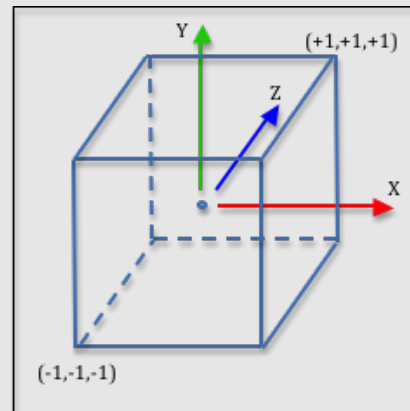


Рис. 4: Граница ортографической проекции (куб)

Ортографическое проектирование

Этапы преобразований

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

2023-10-28

Наблюдение и проектирование

└─ Преобразование проектирования

└─ Ортогографическое проектирование

Ортографическое проектирование

Этапы преобразования

- 1 Центрирование границ ортогографической проекции относительно начала системы координат в зависимости от заданных параметров.
- 2 Масштабирование границ ортогографической 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 проектирование

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

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

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

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

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

2023-10-28

Наблюдение и проектирование

└ Преобразование проектирования

└ Ортографическое проектирование

Ортографическое проектирование

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

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

Параметрическое уравнение прямой в пространстве

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

Ортографическое проектирование

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

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

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

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

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

2023-10-28

Наблюдение и проектирование

└ Преобразование проектирования

└ Ортографическое проектирование

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 проектирование

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

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

2023-10-28

Наблюдение и проектирование

└─ Преобразование проектирования

└─ Ортографическое проектирование

Левосторонняя система координат (left hand)
inversion z-axis

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

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

Ортографическое проектирование

$$M_t M_s M_{lh} =$$

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

2023-10-28

Наблюдение и проектирование

└ Преобразование проектирования

└ Ортографическое проектирование

$$M_t M_s M_{lh} =$$

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

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

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

Перспективное проектирование

```
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 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 ортогографической проекции отсекается.

2023-10-28

Наблюдение и проектирование

└─ Преобразование проектирования

└─ Перспективное проектирование

Перспективное проектирование

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

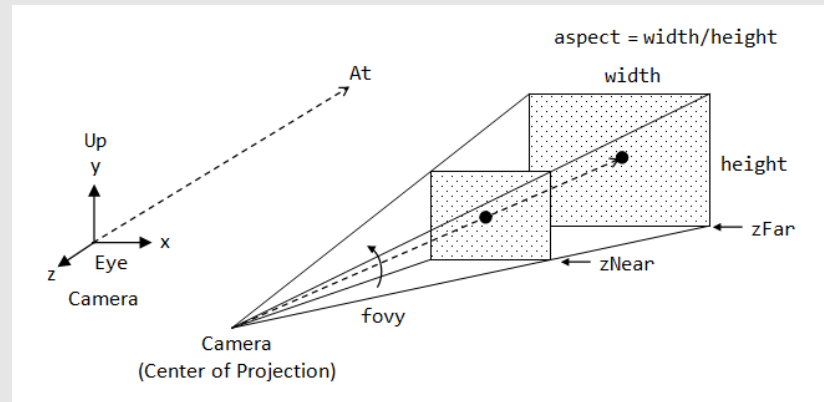


Рис. 5: Перспективная проекция

Перспективное проектирование

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" 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:

$$aspect = right/top$$

2023-10-28

Наблюдение и проектирование

└─ Преобразование проектирования

└─ Перспективное проектирование

Перспективное проектирование
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" 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:

$$aspect = right/top$$

В дальнейшем же будет применяться обратные преобразования:

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

$$bottom = -top$$

$$right = top \cdot aspect$$

$$left = -right$$

Перспективное проектирование

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 квадрату с диапазоном от $(-1, -1)$ до $(+1, +1)$.

2023-10-28

Наблюдение и проектирование

└ Преобразование проектирования

└ Перспективное проектирование

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 квадрату с диапазоном от $(-1, -1)$ до $(+1, +1)$.

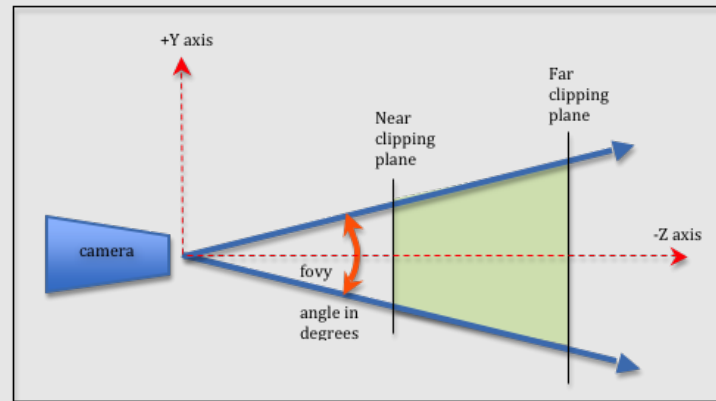


Рис. 6: Перспективная проекция

Перспективное проектирование

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

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

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

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

2023-10-28

Наблюдение и проектирование

└ Преобразование проектирования

└ Перспективное проектирование

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

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

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

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

Перспективное проектирование

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

В основе лежит нелинейное уравнение и имеет вид:

$$z' = c_1/(-z) + c_2,$$

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

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

2023-10-28

Наблюдение и проектирование

Преобразование проектирования

Перспективное проектирование

Перспективное проектирование

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

В основе лежит нелинейное уравнение и имеет вид:

$$z' = c_1/(-z) + c_2,$$

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

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

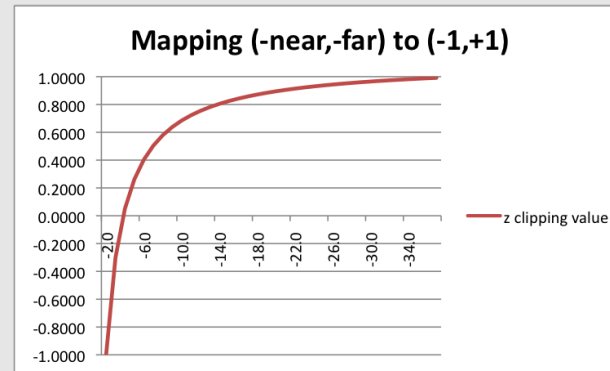


Рис. 7: Кривая нелинейного уравнения

Перспективное проектирование

В результате получаем следующую систему:

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

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?

2023-10-28

Наблюдение и проектирование

Преобразование проектирования

Перспективное проектирование

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:

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

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 проектирование

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

$$(-c_2 * z + c_1)/(-z)$$

Используем матричные преобразования:

перемещение, т.е. сложение,
масштабирование, т.е. умножение,
проецирование, т.е. деление.

Тогда получим следующую матрицу.

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

2023-10-28

Наблюдение и проектирование

└ Преобразование проектирования

└ Перспективное проектирование

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

$$(-c_2 * z + c_1)/(-z)$$

Используем матричные преобразования:
перемещение, т.е. сложение,
масштабирование, т.е. умножение,
проецирование, т.е. деление.
Тогда получим следующую матрицу:

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

Перспективное проектирование

3. Расчет перспективы

Пусть (x, y, z) — координаты вершины.

Задача: отобразить на 2D окне просмотра.

Проецируем вершину на ближнюю плоскость окна просмотра, используя следующие формулы:

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

Интерпретация.

Здесь near - это значение, представляющее ближнюю плоскость отсечения. Значения 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.

2023-10-28

Наблюдение и проектирование

└ Преобразование проектирования

└ Перспективное проектирование

Перспективное проектирование

3. Расчет перспективы

Пусть (x, y, z) — координаты вершины.

Задача: отобразить на 2D окне просмотра.

Проецируем вершину на ближнюю плоскость окна просмотра, используя следующие формулы:

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

Интерпретация.

Здесь near - это значение, представляющее ближнюю плоскость отсечения. Значения 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.

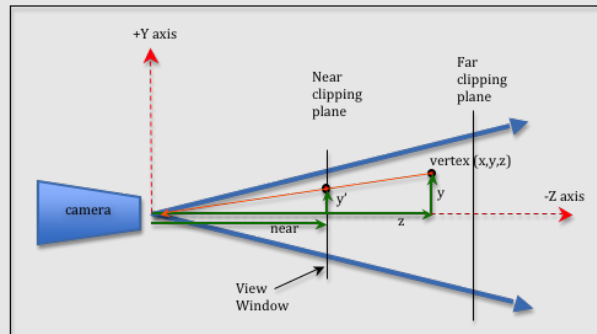


Рис. 8: Схема смещения значений

Перспективное проектирование

Тогда матрица будет иметь вид:

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

2023-10-28

Наблюдение и проектирование

└─ Преобразование проектирования

└─ Перспективное проектирование

Тогда матрица будет иметь вид:

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

Перспективное проектирование

4. Масштабирование двумерных величин (x', y') к квадрату с диапазоном от $(-1, -1)$ до $(+1, +1)$

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

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

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

2023-10-28

Наблюдение и проектирование

└ Преобразование проектирования

└ Перспективное проектирование

4. Масштабирование двумерных величин (x', y') к квадрату с диапазоном от $(-1, -1)$ до $(+1, +1)$

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

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

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

Перспективное проектирование

2023-10-28

Наблюдение и проектирование

└ Преобразование проектирования

└ Перспективное проектирование

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

$$M_t M_{sz} M_p M_{s2d} =$$

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

Промежуточные расчеты

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

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