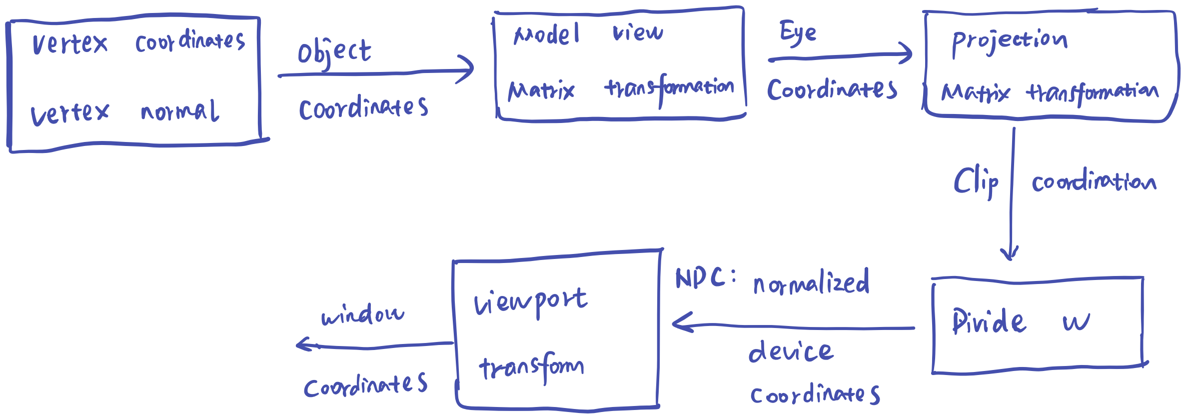
1. overview



2. object coordinates

→ 物体最原本的局部坐标下的顶点信息

3. Eye Coordinates

$$\begin{pmatrix} x_{eye} \\ y_{eye} \\ z_{eye} \\ w_{eye} \end{pmatrix} = M_{modelview} \begin{pmatrix} x_{obj} \\ y_{obj} \\ z_{obj} \\ w_{obj} \end{pmatrix} = M_{view} M_{model} \cdot \begin{pmatrix} x_{obj} \\ y_{obj} \\ z_{obj} \\ w_{obj} \end{pmatrix}$$

→ M_{model} : object world to Global world

→ M_{view} : Global word space to eye space world

→ 于 openGL 中 $M_{modelview}$ 用一个矩阵表示即 GL_MODELVIEW

→ 于 openGL 中，相机位于 (0.0.0)，看向 (-z)

→ 对于顶点法向量信息，同样将 $M_{modelview}$ 作用于 object world 下的法向量，

变换为

$$\begin{pmatrix} nx_{eye} \\ ny_{eye} \\ nz_{eye} \\ nw_{eye} \end{pmatrix} = \left(M_{modelview}^{-1}\right)^T \begin{pmatrix} nx_{obj} \\ ny_{obj} \\ nz_{obj} \\ nw_{obj} \end{pmatrix}$$

why?

4. Clip Coordinates (裁减坐标)

$$\begin{pmatrix} x_{clip} \\ y_{clip} \\ z_{clip} \\ w_{clip} \end{pmatrix} = M_{projection} \begin{pmatrix} x_{eye} \\ y_{eye} \\ z_{eye} \\ w_{eye} \end{pmatrix}$$

→ $M_{projection}$ (GL_PROJECTION) 定义了视锥 (view frustum)

→ $M_{projection}$ 可以是两种不同投影
- 正交投影: orthognal projection
- 透视投影: perspective projection

5. Normalize Device Coordinates (NDC)

$$\begin{pmatrix} x_{NDC} \\ y_{NDC} \\ z_{NDC} \end{pmatrix} = \begin{pmatrix} x_{clip} / w_{clip} \\ y_{clip} / w_{clip} \\ z_{clip} / w_{clip} \end{pmatrix}$$

→ x·y·z 坐标都被正则化到了 (-1,1)

→ x·y·z 坐标更像 window (screen) coordinates., 只是还未被压缩到二维

[6. Window Coordinates (Screen Coordinates)]

① 将 NDC 坐标进行视口变换 (view port) 生成

② NDC坐标被 (scale) & (transport) 以适应渲染屏幕

③ window Coordinates 最终被送入 rasterization pipeline 以形成片段

④ GL_viewport(x, y, w, h) 用于定义最终图像映射渲染的矩形

⑤ GL_depthRange(n, f) 用于确定窗口的 z 值.

⑥ 窗口坐标计算 $\begin{cases} gl\_viewport\ (x, y, w, h) \\ gl\_depthrange\ (n, f) \end{cases}$

$$\begin{pmatrix} x_w \\ y_w \\ z_w \end{pmatrix} = \begin{pmatrix} \frac{w}{2} x_{NDC} + \left(x + \frac{w}{2}\right) \\ \frac{h}{2} x_{NDC} + \left(y + \frac{h}{2}\right) \\ \frac{f-n}{2} x_{NDC} + \frac{f+n}{2} \end{pmatrix}$$

最终变换后 (x_w, y_w, z_w) 范围为

$$\begin{cases} -1 \longrightarrow x \\ 1 \longrightarrow x+w \end{cases} \qquad \begin{cases} -1 \longrightarrow y \\ 1 \longrightarrow y+h \end{cases} \qquad \begin{cases} -1 \longrightarrow n \\ 1 \longrightarrow f \end{cases}$$

[7. openGL 中的 Transformation Matrix]

① 变换矩阵的形式

$$\begin{pmatrix} m_0 & m_4 & m_8 & m_{12} \\ m_1 & m_5 & m_9 & m_{13} \\ m_2 & m_6 & m_{10} & m_{14} \\ m_3 & m_7 & m_{11} & m_{15} \end{pmatrix}$$

以列主元行式排布

② glMatrixMode (type) 有四种 type

- GL_MODElVIEW
- GL_PROJECTION
- GL_TEXTURE
- GL_COLOR

[8. 关于如何在 openGL 中构造 GL_PROJCETION (投影矩阵)]

① GL_PROJCTION 做了两件事

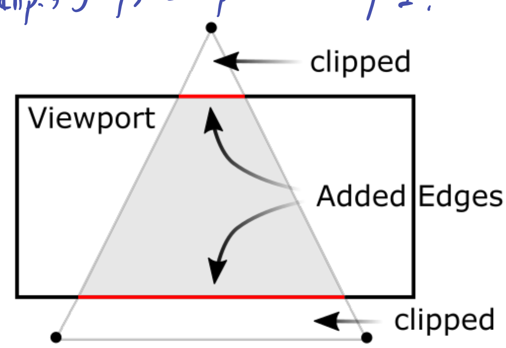
1) 将 eye_coordinates 转为 Clip Coordinates.

2) 将 Clip Coordinates 同除 w分量转为 NDC

② GL_PROJCTION 构建需6个参数 $(\overset{l}{\text{left}}, \overset{r}{\text{right}}, \overset{b}{\text{bottom}}, \overset{t}{\text{top}}, \overset{n}{\text{near}}, \overset{f}{\text{far}})$

③ 注意由于除以 W_{clip} 原因. 故 $[-W_{clip} < X_{clip}, y_{clip}, Z_{clip} < W_{clip}]$.

所有没有在范围内坐标都因此会被裁剪



④ 经过一番推导，由 Eey-world to NDC 的坐标变换公式为

(透视投影)

perspective projection

$$x_n = \left(\frac{2n}{r-l}\cdot x_e + \frac{r+l}{r-l}\cdot z_e\right) \Big/ -z_e$$

Note: 这里的 $\left(\frac{2n}{r-l}\cdot x_e + \frac{r+l}{r-l}\cdot z_e\right)$ 即由 Eey-world 变到了 clip world.

Note: 除以 $(-z_e)$ 由原本在 Eey-world 下的右手系

变为了于 NDC坐标下的左手系

$$y_{ndc} = \underbrace{\left(\frac{2n}{t-b}\cdot y_{eye} + \frac{t+b}{t-b}\cdot z_{eye}\right)}_{y_{clip}} \Big/ -z_{eye}$$

⑤ 因此单纯的投影矩阵为 (! 仅是投影矩阵. 没有包括除以 W_c 获得 NDC那一步)

$$\text{Projection Matrix} \quad \begin{pmatrix} 2n/r-l & 0 & \frac{r+l}{r-l} & 0 \\ 0 & 2n/t-b & \frac{t+b}{t-b} & 0 \\ 0 & 0 & \frac{-(f+n)}{f-n} & \frac{-2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

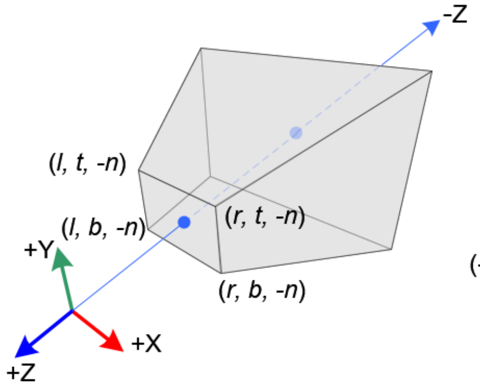
由这个矩阵可以实现. 由 eye-world to clip world. 即

$$\begin{pmatrix} x_{clip} \\ y_{clip} \\ z_{clip} \\ w_{clip} \end{pmatrix} = \begin{pmatrix} 2n/r-l & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & \frac{-(f+n)}{f-n} & \frac{-2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} x_{eye} \\ y_{eye} \\ z_{eye} \\ w_{eye} \end{pmatrix}$$

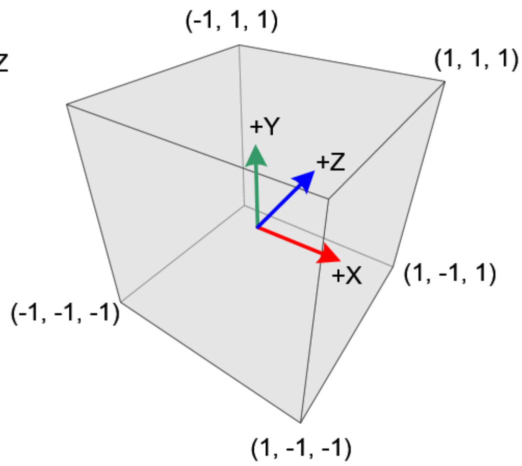
且这里有 $-z_{eye} = w_{clip}$. 之后只要对 x_{clip}, y_{clip}, z_{clip} 除以 w_{clip}

就相当于除以 $(-z_{eye})$ 了. 所以以由 eye-world 直接到 NDC world.

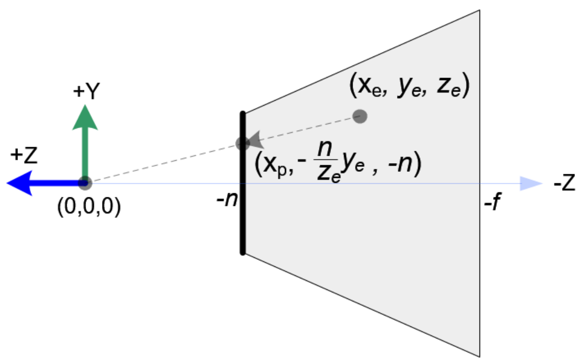
图示



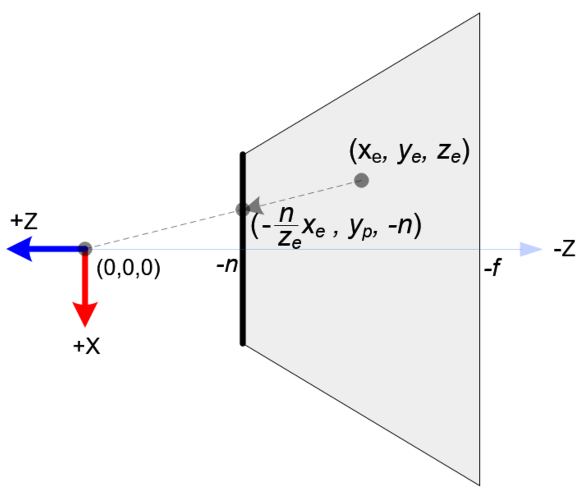
-Z
(l, t, -n)
(l, b, -n)
(r, t, -n)
(r, b, -n)
+Y
+X
+Z



(-1, 1, 1)
(1, 1, 1)
+Y
+Z
+X
(1, -1, 1)
(-1, -1, -1)
(1, -1, -1)



(x_e, y_e, z_e)
$(x_p, -\frac{n}{z_e}y_e, -n)$
+Y
+Z
(0,0,0)
-n
-f
-Z



(x_e, y_e, z_e)
$(-\frac{n}{z_e}x_e, y_p, -n)$
+Z
(0,0,0)
-n
-f
-Z
+X

⑥ 正交投影的投影矩阵 . M_{ortho}

只需将原本矩形变为长方形的 Cube，再移动至原点即可

$$M_{ortho} = \begin{pmatrix} \frac{2}{r-l} & 0 & 0 & -\frac{r+l}{r+l} \\ 0 & \frac{2}{t-b} & 0 & -\frac{t+b}{t-b} \\ 0 & 0 & \frac{-2}{f-n} & -\frac{f+n}{f-n} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

即

$$\begin{pmatrix} x_{NDC} \\ y_{NDC} \\ z_{NDC} \\ w_{NDC} \end{pmatrix} = \begin{pmatrix} 2/r-l & 0 & 0 & -\frac{r+l}{r-l} \\ 0 & 2/t-b & 0 & -\frac{t+b}{t-b} \\ 0 & 0 & -2/f-n & -\frac{f+n}{f-n} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{eye} \\ y_{eye} \\ z_{eye} \\ w_{eye} \end{pmatrix}$$