几何管线相关矩阵

几何管线将数据从模型坐标系转换窗口坐标系，经过一系列的转换，数据点经过以下几个步骤：X_{model} -> X_{world} -> X_{view} -> X_{clip} -> X_{ndc} -> X_{window}，模型数据转换成二维屏幕上可显示的像素数据，上述转换需要用到几个转换矩阵

D3 使用行向量，而 OpenGL 则使用列向量，此处我们采用列向量，令右手笛卡尔坐标是坐标系统的基础坐标系

世界矩阵

- 将模型坐标系中的数据转换到世界坐标系中
- 处理世界空间以及模型空间时的主要问题包括定位，定向以及模型的缩放操作，进而可在空间内正确设置模型对象。

令模型空间中的点 $P_0=(0,0,0)$，$P_1=(1,0,0)$，$P_2=(0,1,0)$，$P_3=(0,0,1)$，对应世界坐标系中的点为$Q_i = O + d_iD + u_iU + r_iR(0 <= i <= 3)$，这一过程通过构造仿射转换可以实现，即将 P_0 映射到Q_0，并将向量$P_i - P_0$映射到$Q_i - Q_0$，转换过程是$Q_i = Q_0 + M(P_i - P_0)$，我们需要构造如下等式

$$M[P_1 - P_0, P_2 - P_0, P_3 - P_0] = [Q_1 - Q_0, Q_2 - Q_0, Q_3 - Q_0]$$

这里两个分块矩阵包含基于标志向量的列矩阵，则

$$M = [Q_1 - Q_0, Q_2 - Q_0, Q_3 - Q_0][P_1 - P_0, P_2 - P_0, P_3 - P_0]^{-1}$$

M 为 3X3 的矩阵，包括缩放，旋转，反射，剪切等其它线性转换

设平移向量为 3X1 的向量 B，3X1 的模型空间点X_{model}，3X1 的世界空间点X_{world}

可得到$\begin{bmatrix} X_{world} \\ 1 \end{bmatrix} = \begin{bmatrix} M & B \\ 0^T & 1 \end{bmatrix}\begin{bmatrix} X_{model} \\ 1 \end{bmatrix} = H_{world}\begin{bmatrix} X_{model} \\ 1 \end{bmatrix}$

$$H_{world} = \begin{bmatrix} M & B \\ 0^T & 1 \end{bmatrix}$$

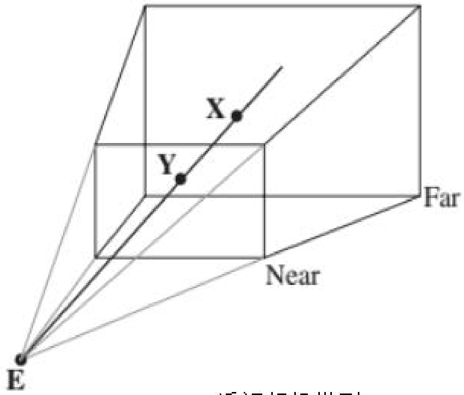
世界矩阵的逆矩阵也可知

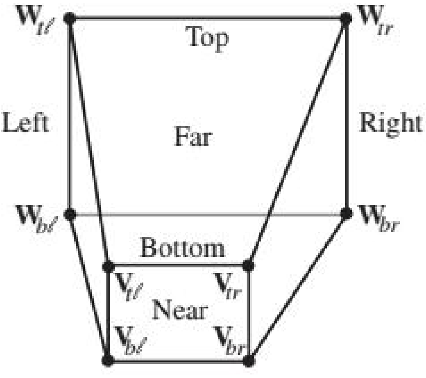
$$H^{-1}{}_{world} = \begin{bmatrix} M^{-1} & -M^{-1}B \\ 0^T & 1 \end{bmatrix}$$

透视相机模型

相机模型可以推导出相机矩阵和投影矩阵，它是由相机位置，赋予相机位置的一组坐标，视

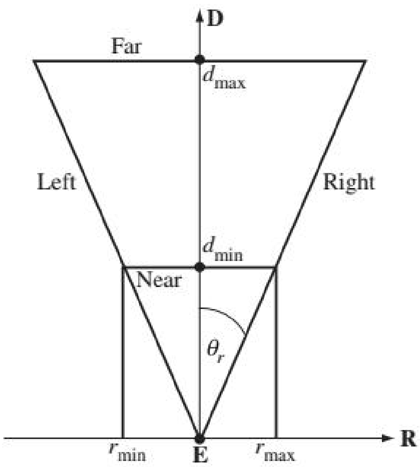
平面，视口，以及视椎体构成。



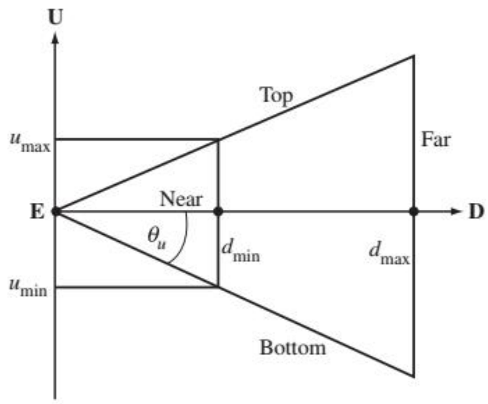


(a)

透视相机模型



(b)



(c)

视见矩阵定义在世界坐标系中，设相机位置为 E，视见方向为垂直于视平面的单位向量 D，up 方向为 U，右方向为 $R = DxU$，模型坐标系可用右手坐标系{E; D, U, R}

给定 $d_{min}, d_{max}, u_{min}, u_{max}, r_{min}, r_{max}$ 有

$$V_{bl} = E + d_{\min} D + u_{\min} U + r_{\min} R$$
$$V_{zl} = E + d_{\min} D + u_{\max} U + r_{\min} R$$
$$V_{br} = E + d_{\min} D + u_{\min} U + r_{\max} R$$
$$V_{tr} = E + d_{\min} D + u_{\max} U + r_{\max} R$$
$$W_{bl} = E + \frac{d_{\max}}{d_{\min}}(d_{\min} D + u_{\min} U + r_{\min} R)$$
$$W_{tl} = E + \frac{d_{\max}}{d_{\min}}(d_{\min} D + u_{\max} U + r_{\min} R)$$
$$W_{br} = E + \frac{d_{\max}}{d_{\min}}(d_{\min} D + u_{\min} U + r_{\max} R)$$
$$W_{tr} = E + \frac{d_{\max}}{d_{\min}}(d_{\min} D + u_{\max} U + r_{\max} R)$$

- 投影面上一点可表示为 $\mathrm{P} = \mathrm{E} + d_{\min} D$，其法线-D，令 X 为其上一点，则可表示为 $-\mathrm{D} \bullet (X - P) = 0$，可求得 $-\mathrm{D} \bullet \mathrm{X} = -(D \bullet E + d_{\min})$
- 近裁剪面上一点为 $\mathrm{P} = \mathrm{E} + d_{\min} D$，其法线 D，令 X 为其上一点，则可表示为 $\mathrm{D} \bullet (X - P) = 0$，可求得 $\mathrm{D} \bullet \mathrm{X} = D \bullet E + d_{\min}$
- 远裁剪面上一点可表示为 $\mathrm{P} = \mathrm{E} + d_{\max} D$，其法线-D，令 X 为其上一点，则可表示为 $-\mathrm{D} \bullet (X - P) = 0$，可求得 $-\mathrm{D} \bullet \mathrm{X} = -(D \bullet E + d_{\max})$
- 左侧平面包含 3 个点，即 $\mathrm{E}, V_{tl}, V_{bl}$，指向视椎体内侧的法向量为

$$\begin{aligned}
&(V_{bl} - E)\mathrm{x}(V_{tl} - E) = (d_{\min} D + u_{\min} U + r_{\min} R)\mathrm{x}(d_{\min} D + u_{\max} U + r_{\min} R) \\
&= (d_{\min} D + r_{\min} R)\mathrm{x}(u_{\max} U) + (u_{\min} U)\mathrm{x}(d_{\min} D + r_{\min} R) \\
&= (d_{\min} D + r_{\min} R)\mathrm{x}((u_{\max} - u_{\min})U) \\
&= (u_{\max} - u_{\min})(d_{\min} D\mathrm{x}U + r_{\min} R\mathrm{x}U) \\
&= (u_{\max} - u_{\min})(d_{\min} R - r_{\min} D)
\end{aligned}$$

则左侧平面法向量以及左侧平面为 $N_l = \frac{d_{\min}\mathrm{R} - \mathrm{r}_{\min}\mathrm{D}}{\sqrt{d^2_{\min} + r^2_{\min}}}, N_l \bullet (X - E) = 0$

- 指向内侧的右平面法线表示为 $(V_{tr} - E)\mathrm{x}(V_{br} - E)$,右平面法向量以及右侧平面是：

$$N_r = \frac{-d_{\min}\mathrm{R} + \mathrm{r}_{\max}\mathrm{D}}{\sqrt{d^2_{\min} + r^2_{\max}}}, N_r \bullet (X - E) = 0$$

- 指向内侧的单位长度法线以及底面可表示为

$$N_b = \frac{d_{\min}\mathrm{U} - \mathrm{u}_{\min}\mathrm{D}}{\sqrt{d^2_{\min} + u^2_{\max}}}, N_b \bullet (X - E) = 0$$

- 指向内侧的单位长度法线以及上表面可表示为

$$N_t = \frac{-d_{\min}U + u_{\max}D}{\sqrt{d^2_{\min} + u^2_{\max}}}, N_t \bullet (X - E) = 0$$

视见矩阵（相机矩阵）

相机矩阵可以和和 D3D 一样使用左手坐标系，也可以同 OpenGL 一样使用右手坐标系，此处与 D3D 保持一致使用左手坐标系。

则 $X_{world} = E + dD + uU + rR$

$d = D \bullet (X_{world} - E)$　$u = U \bullet (X_{world} - E)$　$r = R \bullet (X_{world} - E)$

$$X_{view} = \begin{bmatrix} r \\ u \\ d \end{bmatrix} = \begin{bmatrix} R \bullet (X_{world} - E) \\ U \bullet (X_{world} - E) \\ D \bullet (X_{world} - E) \end{bmatrix} = \begin{bmatrix} R_T \\ U_T \\ D_T \end{bmatrix} (X_{world} - E) = \begin{bmatrix} R & U & D \end{bmatrix}^T (X_{world} - E)$$

令 $Q = \begin{bmatrix} R & U & D \end{bmatrix}$，则可知 $\begin{bmatrix} \frac{X_{view}}{1} \end{bmatrix} = \begin{bmatrix} \frac{Q^T}{0^T} & \frac{-Q^T E}{1} \end{bmatrix} \begin{bmatrix} \frac{X_{world}}{1} \end{bmatrix} = H_{view} \begin{bmatrix} \frac{X_{world}}{1} \end{bmatrix}$

$$H_{view} = \begin{bmatrix} \frac{Q^T}{0^T} & \frac{-Q^T E}{1} \end{bmatrix} = \begin{bmatrix} r0 & r1 & r2 & -dot(R, E) \\ u0 & u1 & u2 & -dot(U, E) \\ d0 & d1 & d2 & -dot(D, E) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H^{-1}_{view} = \begin{bmatrix} \frac{Q}{0^T} & \frac{E}{1} \end{bmatrix} = \begin{bmatrix} r0 & u0 & d0 & e0 \\ r1 & u1 & d1 & e1 \\ r2 & u2 & d2 & e2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

示例：DirectX 帮助文档中有

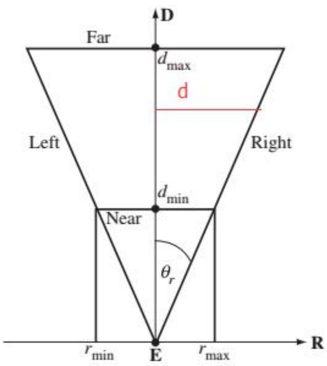
```
D3DXMATRIX * D3DXMatrixLookAtLH(
    D3DXMATRIX * pOut,
    CONST D3DXVECTOR3 * pEye,
    CONST D3DXVECTOR3 * pAt,
    CONST D3DXVECTOR3 * pUp
);

zaxis = normal(pAt - pEye)
xaxis = normal(cross(pUp, zaxis))
yaxis = cross(zaxis, xaxis)
```

$$\begin{bmatrix} \text{xaxis.x} & \text{yaxis.x} & \text{zaxis.x} & 0 \\ \text{xaxis.y} & \text{yaxis.y} & \text{zaxis.y} & 0 \\ \text{xaxis.z} & \text{yaxis.z} & \text{zaxis.z} & 0 \\ -\text{dot(xaxis, eye)} & -\text{dot(yaxis, eye)} & -\text{dot(zaxis, eye)} & 1 \end{bmatrix}$$

透视投影矩阵

相机模型中的视椎体轴可看作是包含原点和视口中心点的射线，此射线可以通过视见坐标 d 实现参数化，$\left(\frac{(r_{min}+r_{max})d}{2d_{min}} \quad \frac{(u_{min}+u_{max})d}{2d_{min}} \quad d\right)$

证明如下：见下图，点形式（r,u,d）表示射线，可知 $\frac{d_{min}}{d}=\frac{\frac{r_{min}+r_{max}}{2}}{r}$，所以 $r=\frac{r_{min}+r_{max}}{2}\cdot\frac{d}{d_{min}}$，同理可知 $u=\frac{u_{min}+u_{max}}{2}\cdot\frac{d}{d_{min}}$



此时我们可以将斜视锥体转换为基于 $[-1,1]^2$ 视口的正交视椎体，我们可以通过移除斜交系统并对结果进行缩放完成

$$r'=\frac{2}{r_{max}-r_{min}}\left(d_{min}r-\frac{r_{min}+r_{max}}{2}d\right)$$

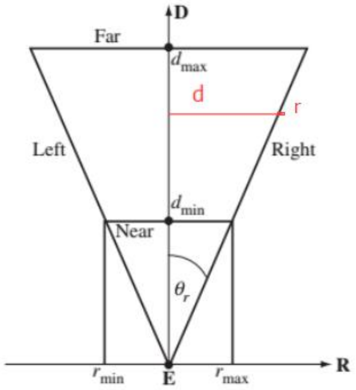
$$u'=\frac{2}{u_{max}-u_{min}}\left(d_{min}u-\frac{u_{min}+u_{max}}{2}d\right)$$

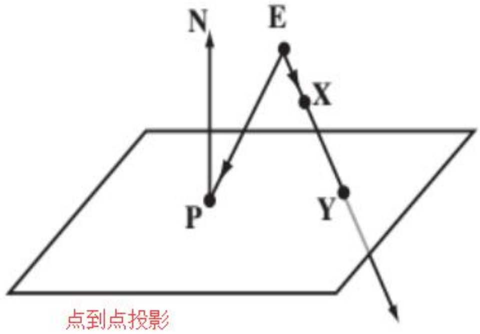
证明如下：见下图，斜视锥体内所有点都缩放到以 r_{max} 为基准，可先求缩放系数，知 $\frac{r}{r_{max}}=\frac{d}{d_{min}}$， 故 $r_{max}=\frac{d_{min}}{d}\cdot r$，缩放系数为 $\frac{d_{min}}{d}$

则：$\frac{\left(r'-\frac{(r_{min}+r_{max})d}{2d_{min}}\right)\cdot\frac{d_{min}}{d}}{\frac{r_{max}-r_{min}}{2}}\in[-1 \quad 1]$ ==> $r'd\in[-1 \quad 1]$ ==> $r'\in[-d \quad d]$

同上可推导出 u′，且 $u'\in[-d \quad d]$

令 $w'=d$， $d_{min} <= w' <= d_{max}$





看下直线到直线的投影模型

直线到直线投影形式可知投影后 s 变为 X 为视见坐标系中一点，Y 是 X 的投影点

$\frac{Y-E}{X-E}=\frac{d_{min}}{d}$ ==> $Y=E+\frac{d_{min}}{d}(X-E)$，故视见坐标系下点（$r,u,d$）的投影点为$(\frac{d_{min}}{d}r,\frac{d_{min}}{d}u,d_{min})$

端点为$Q_i=(d_i,u_i,r_i)(i=0,1)$的直线，投影点$P_i=(r_i/w_i,u_i/w_i,d_{min})$，$w_i=d_i/d_{min}(i=0,1)$

3D的直线段表示为$Q(s)=Q_0+s(Q_1-Q_0)=(r_0+s(r_1-r_0),u_0+s(u_1-u_0),d_0+s(d_1-d_0))$

$$P(s)=(\frac{r_0+s(r_1-r_0)}{w_0+s(w_1-w_0)},\frac{u_0+s(u_1-u_0)}{w_0+s(w_1-w_0)},d_{min})=P_0+\frac{w_1 s}{w_0+s(w_1-w_0)}(P_1-P_0)$$

故$\bar{s}=\frac{w_1 s}{w0+(w_1-w_0)s}$

我们可以将 d 映射到$[0\quad 1]$

取$s\in[0\quad 1]$可知 $d=(1-s)d_{min}+sd_{max}$，故 $s=\frac{d-d_{min}}{d_{max}-d_{min}}$

$$\bar{s}=\frac{w_1 s}{w0+(w_1-w_0)s}=\frac{d_{max}}{d_{max}-d_{min}}\cdot\frac{d-d_{min}}{d}\in(0,1)$$，

令$d'd=\bar{s}$ ==> $d'=\frac{d_{max}}{d_{max}-d_{min}}\cdot(d-d_{min})$

可知 $X_{clip}=\begin{bmatrix}r'\\u'\\d'\\w'\end{bmatrix}=\begin{bmatrix}\frac{2d_{min}}{r_{max}-r_{min}} & 0 & \frac{-(r_{max}+r_{min})}{r_{max}-r_{min}} & 0\\ 0 & \frac{2d_{min}}{u_{max}-u_{min}} & \frac{-(u_{max}+u_{min})}{u_{max}-u_{min}} & 0\\ 0 & 0 & \frac{d_{max}}{d_{max}-d_{min}} & \frac{-d_{max}d_{min}}{d_{max}-d_{min}}\\ 0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}r\\u\\d\\1\end{bmatrix}=H_{proj}\begin{bmatrix}X_{view}\\1\end{bmatrix}$

所以投影矩阵是

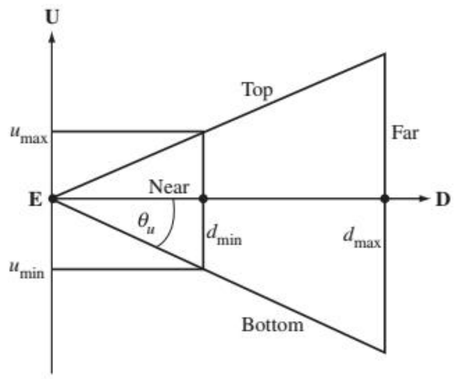
$$H_{proj}=\begin{bmatrix} \frac{2d_{min}}{r_{max}-r_{min}} & 0 & \frac{-(r_{max}+r_{min})}{r_{max}-r_{min}} & 0 \\ 0 & \frac{2d_{min}}{u_{max}-u_{min}} & \frac{-(u_{max}+u_{min})}{u_{max}-u_{min}} & 0 \\ 0 & 0 & \frac{d_{max}}{d_{max}-d_{min}} & \frac{-d_{max}d_{min}}{d_{max}-d_{min}} \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

容易得到

$$H^{-1}_{proj}=\begin{bmatrix} \frac{r_{max}-r_{min}}{2d_{min}} & 0 & 0 & \frac{r_{max}+r_{min}}{2d_{min}} \\ 0 & \frac{u_{max}-u_{min}}{2d_{min}} & 0 & \frac{u_{max}+u_{min}}{2d_{min}} \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{d_{min}-d_{max}}{d_{max}d_{min}} & \frac{1}{d_{min}} \end{bmatrix}$$

一般在构造投影矩阵时，会给视场角 θ_u，宽高比σ，近裁剪面的距离 n，远裁剪面的距离 f，同时 $r_{max}=-r_{min}$ $u_{max}=-u_{min}$， 我们可以得到$\tan(\theta_u)=\frac{u_{max}}{n}$，故 $u_{max}=\tan(\theta_u)\cdot n$

$\sigma=\frac{r_{max}}{u_{max}}$， $r_{max}=\sigma\cdot u_{max}=\sigma\cdot\tan(\theta u)\cdot n$



所以投影矩阵也能简化为

$$H_{proj}=\begin{bmatrix} \frac{1}{\sigma\cdot\tan(\theta_u)} & 0 & 0 & 0 \\ 0 & \frac{1}{\tan(\theta_u)} & 0 & 0 \\ 0 & 0 & \frac{f}{f-n} & \frac{-fn}{f-n} \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

示例：DirectX 帮助文档里有，这里不同的地方是因为 D3D 使用行向量

```
D3DXMATRIX * D3DXMatrixPerspectiveFovLH(
    D3DXMATRIX * pOut,
    FLOAT fovy,
    FLOAT Aspect,
```

```
    FLOAT   zn,
    FLOAT   zf
);
```

$$\begin{bmatrix} \frac{2}{\text{Aspect} * \tan(\text{fovY}/2)} & 0 & 0 & 0 \\ 0 & \frac{2}{\tan(fovY/2)} & 0 & 0 \\ 0 & 0 & \frac{1}{\text{zf} - zn} & 0 \\ 0 & 0 & \frac{zn}{zn - zf} & 1 \end{bmatrix}$$

正交投影矩阵

正交投影矩阵模型和透视投影矩阵的模型差不多，不同的是这里使用的是正交视椎体，推导过程基本和推导窗口矩阵一致，此处就不推导了

$$\mathrm{H}_{ortho} = \begin{bmatrix} \frac{2}{r\max - r\min} & 0 & 0 & -\frac{r_{\max} + r_{\min}}{r_{\max} - r_{\min}} \\ 0 & \frac{2}{u_{\max} - u_{\min}} & 0 & -\frac{u_{\max} + u_{\min}}{u_{\max} - u_{\min}} \\ 0 & 0 & \frac{1}{zf - zn} & -\frac{zn}{zf - zn} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$\mathrm{r}_{\max} - r_{\min} = W$

若 $u_{\max} - u_{\min} = H$

$u_{\max} = -u_{\min}, r_{\max} = -r_{\min}$

则

$$\mathrm{H}_{ortho} = \begin{bmatrix} \frac{2}{\mathrm{W}} & 0 & 0 & 0 \\ 0 & \frac{2}{\mathrm{H}} & 0 & 0 \\ 0 & 0 & \frac{1}{zf - zn} & -\frac{zn}{zf - zn} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

示例：

```
D3DXMATRIX *  D3DXMatrixOrthoLH(
    D3DXMATRIX *  pOut,
    FLOAT   w,
    FLOAT   h,
```

```
    FLOAT   zn,
    FLOAT   zf
);
```

$$\begin{bmatrix} \frac{2}{w} & 0 & 0 & 0 \\ 0 & \frac{2}{h} & 0 & 0 \\ 0 & 0 & \frac{1}{zf - zn} & 0 \\ 0 & 0 & \frac{zn}{zn - zf} & 1 \end{bmatrix}$$

窗口矩阵

窗口矩阵需要做的就是将裁剪空间的数据转换成屏幕上显示的像素数据
裁剪空间点

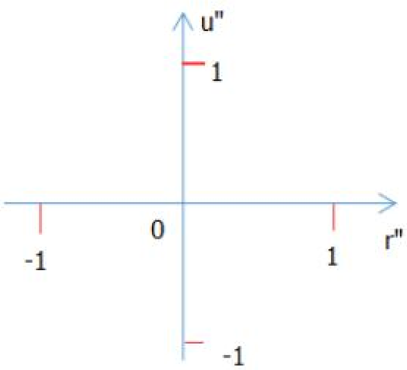
$(r' \quad u' \quad d' \quad w')$，其中$|r'| <= w', |u'| <= w', 0 <= d' <= d_{max}, d_{min} <= w' <= d_{max}$

执行透视除法，可以得到

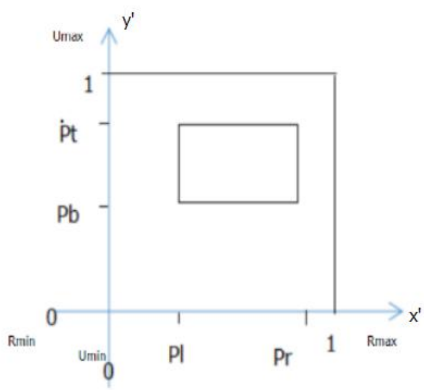
$$X_{ndc} = \begin{bmatrix} r'' \\ u'' \\ d'' \\ 1 \end{bmatrix} = \begin{bmatrix} r'/w' \\ u'/w' \\ d'/w' \\ w'/w' \end{bmatrix}, \text{其中} r'' \in [-1 \quad 1], u'' \in [-1 \quad 1], d'' \in [0 \quad 1]$$

同时我们需要考虑子视口，如下图所示,

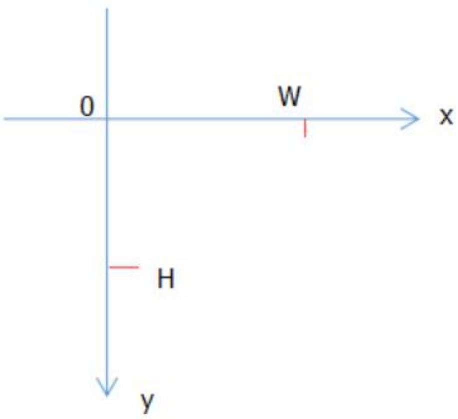
$$0 <= p_l < p_r <= 1, 0 <= p_b < p_t <= 1, 0 <= p_n < p_f <= 1$$



标准化设备坐标系



视口



屏幕坐标系

需要经过两次转换，第一次将数据从标准化设备坐标系转换到子视口空间中，第二次将子视口空间的数据转换到屏幕坐标系中

第一步：

对于 r’’ 和 x'

r’’ : -1 --> 1

x’: pl --> pr

取 $s \in [0 \quad 1]$

$$r'' = -1 + (1-(-1))s = -1 + 2s，可知 s = \frac{1+r''}{2}$$

$$x' = p_l + (p_r - p_l)s = (1-s)p_l + sp_r = \frac{1-r''}{2}p_l + \frac{1+r''}{2}p_r$$

对于 u’’ 和 y'

u’’ : -1 --> 1

y’: pl --> pr

取 $s \in [0 \quad 1]$

$$u'' = -1 + (1-(-1))s = -1 + 2s，可知 s = \frac{1+u''}{2}$$

$$y' = p_b + (p_t - p_b)s = (1-s)p_b + sp_t = \frac{1-u''}{2}p_b + \frac{1+u''}{2}p_t$$

对于 d’’ 和 d'

d’’ :0 --> 1

d’: pn --> pf

取 $s \in [0 \quad 1]$

$$d'' = 0 + (1-0)s = s，s = d''$$

$$d' = pn + (pf - pn)s = pn + (pf - pn)d''$$

第二步：

对于 x’ 和 x

x’: 0 --> 1

x: 0 --> w

$$x' = 0 + (1-0)s = s$$

$$\mathrm{x} = 0 + (w-0)s = s \cdot w = x' \cdot w = \left(\frac{1-r''}{2} p_l + \frac{1+r''}{2} p_r\right) w = \frac{\mathrm{w}}{2}[(p_r + p_l) + (p_r - p_l) r'']$$

对于 y’ 和 y

y’: 1 --> 0

y: 0 --> H

$$\mathrm{y}' = 1 + (0-1)s = 1 - s\text{，则 } \mathrm{s} = 1 - y' = 1 - \left(\frac{1-u''}{2} p_b + \frac{1+u''}{2} p_t\right)$$

$$\mathrm{y} = 0 + (\mathrm{H} - 0)s = s \cdot \mathrm{H} = \left(1 - \left(\frac{1-u''}{2} p_b + \frac{1+u''}{2} p_t\right)\right) H = \frac{\mathrm{H}}{2}[(2 - p_t - p_b) + (p_b - p_t) u'']$$

对于 d’ 和 d

d’: 0 --> 1

d: 0 --> 1

$$\mathrm{d}' = 0 + (1-0)s = s\text{，则 } \mathrm{s} = \mathrm{d}'$$

$$\mathrm{d} = 0 + (1-0)s = s = \mathrm{d}' = pn + (pf - pn) d''$$

故可以得到 X_{ndc} 到 $\mathrm{X}_{\mathrm{window}}$ 的转换矩阵

$$\begin{bmatrix} \mathrm{X}_{\mathrm{window}} \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{w(p_r - p_l)}{2} & 0 & 0 & \frac{w(p_r + p_l)}{2} \\ 0 & \frac{H(p_b - p_t)}{2} & 0 & \frac{H(2 - p_t - p_b)}{2} \\ 0 & 0 & pf - pn & pn \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{X}_{\mathrm{ndc}} \\ 1 \end{bmatrix} = \mathrm{H}_{window} \begin{bmatrix} X_{ndc} \\ 1 \end{bmatrix}$$

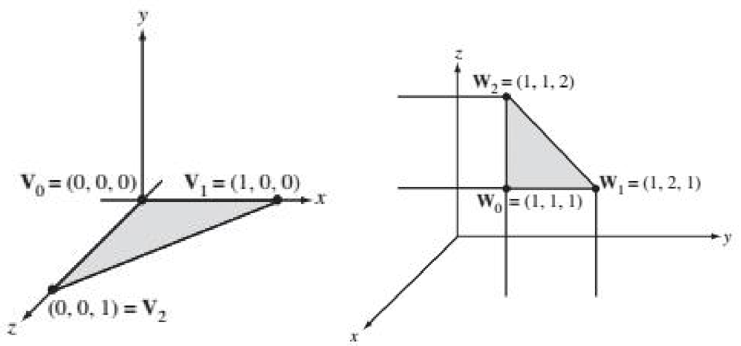
则知

$$\mathrm{H}_{window} = \begin{bmatrix} \frac{w(p_r - p_l)}{2} & 0 & 0 & \frac{w(p_r + p_l)}{2} \\ 0 & \frac{H(p_b - p_t)}{2} & 0 & \frac{H(2 - p_t - p_b)}{2} \\ 0 & 0 & pf - pn & pn \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H^{-1}{}_{window}=\begin{bmatrix}\frac{2}{w(p_r-p_l)} & 0 & 0 & \frac{-(p_r+p_l)}{p_r-p_l}\\ 0 & \frac{2}{H(p_b-p_t)} & 0 & \frac{2-p_t-p_b}{p_b-p_t}\\ 0 & 0 & \frac{1}{pf-pn} & \frac{-pn}{pf-pn}\\ 0 & 0 & 0 & 1\end{bmatrix}$$

例子：

如下图在模型空间中有个三角形，顶点为$V_0=(0,0,0), V_1=(1,0,0), V_2=(0,0,1)$，世界空间包含原点(0,0,0)以及正 z 轴方向上的 Up 向量。三角模型执行旋转和平移操作，且有$W_0=(1,1,1), W_1=(1,2,1), W_2=(1,1,2)$



可以得到世界矩阵$H_{world}=\begin{bmatrix}0 & -1 & 0 & 1\\ 1 & 0 & 0 & 1\\ 0 & 0 & 1 & 1\\ 0 & 0 & 0 & 1\end{bmatrix}$，$H_{world}$可以将所有的$(V_i,1)$转换到$(W_i,1)$

这里我们将屏幕的宽度定义为 640 像素，高度定义为 480 像素，相机位置为 E=(5/2,3,7/2);视见向量定义为 D=(-1,-1,-1)，Up 向量$U=(-1,-1,2)/\sqrt{6}$，右向向量$R=DxU=(-1,1,0)/\sqrt{2}$

$$H_{view}=\begin{bmatrix}\frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & \frac{-1}{2\sqrt{2}}\\ \frac{-1}{\sqrt{6}} & \frac{-1}{\sqrt{6}} & \frac{2}{\sqrt{6}} & \frac{-3}{2\sqrt{6}}\\ \frac{-1}{\sqrt{3}} & \frac{-1}{\sqrt{3}} & \frac{-1}{\sqrt{3}} & \frac{9}{\sqrt{3}}\\ 0 & 0 & 0 & 1\end{bmatrix}$$

正交视椎体的近值和原值为$d_{min}=1, d_{max}=10$，垂直视域为$2\theta_u=\pi/3$，宽高比为

$\rho = 4/3 = 640/480$，可以计算得到 $u_{max} = d_{min}\tan(\theta_u) = 1/\sqrt{3}$，以及 $r_{max} = \rho u_{max} = 4/3\sqrt{3}$，考虑到对称性有 $u_{min} = -u_{max}$， $r_{min} = -r_{max}$，投影矩阵如下所示：

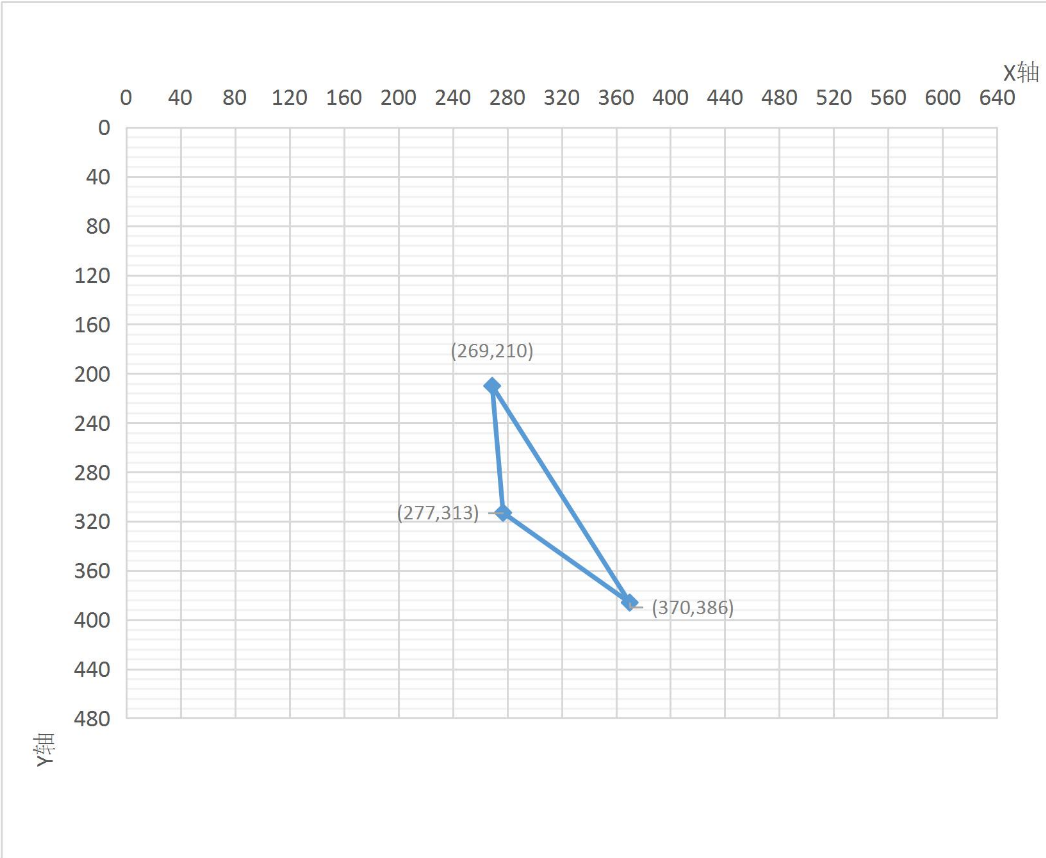
$$
H_{proj} = \begin{bmatrix} \frac{3\sqrt{3}}{4} & 0 & 0 & 0 \\ 0 & \sqrt{3} & 0 & 0 \\ 0 & 0 & \frac{10}{9} & \frac{-10}{9} \\ 0 & 0 & 1 & 0 \end{bmatrix}
$$

此处采用全视口显示，则 $p_l = p_b = 0, p_r = p_t = 1$，采用全值深度，即 $p_n = 0, p_f = 1$

$$
H_{window} = \begin{bmatrix} \frac{639}{2} & 0 & 0 & \frac{639}{2} \\ 0 & -\frac{479}{2} & 0 & \frac{479}{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

空间	定点 0	定点 1	定点 2
模型空间	(0,0,0;1)	(1,0,0;1)	(0,0,1;1)
世界空间	(1,1,1;1)	(1,2,1;1)	(1,1,2;1)
视见空间	$(\frac{-1}{2\sqrt{2}},\frac{-3}{2\sqrt{6}},\frac{6}{\sqrt{3}};1)$	$(\frac{1}{2\sqrt{2}},\frac{-5}{2\sqrt{6}},\frac{5}{\sqrt{3}};1)$	$(\frac{-1}{2\sqrt{2}},\frac{1}{2\sqrt{6}},\frac{5}{\sqrt{3}};1)$
剪裁空间	$(\frac{-3\sqrt{3}}{8\sqrt{2}},\frac{-3}{2\sqrt{2}},\frac{20}{3\sqrt{3}}-\frac{10}{9};\frac{6}{\sqrt{3}})$	$(\frac{3\sqrt{3}}{8\sqrt{2}},\frac{-5}{2\sqrt{2}},\frac{50}{9\sqrt{3}}-\frac{10}{9};\frac{5}{\sqrt{3}})$	$(\frac{-3\sqrt{3}}{8\sqrt{2}},\frac{1}{2\sqrt{2}},\frac{50}{9\sqrt{3}}-\frac{10}{9};\frac{5}{\sqrt{3}})$
NDC 空间	$(\frac{-3}{16\sqrt{2}},\frac{-\sqrt{3}}{4\sqrt{2}},\frac{10}{9}-\frac{5\sqrt{3}}{27};1)$	$(\frac{9}{40\sqrt{2}},\frac{-\sqrt{3}}{2\sqrt{2}},\frac{10}{9}-\frac{2\sqrt{3}}{9};1)$	$(\frac{-9}{40\sqrt{2}},\frac{\sqrt{3}}{10\sqrt{2}},\frac{10}{9}-\frac{2\sqrt{3}}{9};1)$
屏幕空间	$(\frac{639}{2}-\frac{1917}{32\sqrt{2}},\frac{479}{2}+\frac{479\sqrt{3}}{8\sqrt{2}},\frac{10}{9}-\frac{5\sqrt{3}}{27};1)$	$(\frac{639}{2}-\frac{5751}{80\sqrt{2}},\frac{479}{2}+\frac{479\sqrt{3}}{4\sqrt{2}},\frac{10}{9}-\frac{2\sqrt{3}}{9};1)$	$(\frac{639}{2}-\frac{5751}{80\sqrt{2}},\frac{479}{2}-\frac{479\sqrt{3}}{20\sqrt{2}},\frac{10}{9}-\frac{2\sqrt{3}}{9};1)$

最终投影定点的像素位置是(277,313)，(370,386)，(269,210)，


X轴
0 40 80 120 160 200 240 280 320 360 400 440 480 520 560 600 640
0 40 80 120 160 200 240 280 320 360 400 440 480
Y轴
(269,210)
(277,313)
(370,386)